

NoMachine Remote Access Evaluation Report

2015

Version 1.0



Table of Contents

- [A. Document History](#)
- [B. Document Scope](#)
- [C. Executive Summary](#)
- [D. Introduction](#)
- [E. Prerequisites](#)
- [F. Evaluation Details](#)
- [G. Suggestions for Evaluation Procedure Revisions](#)
- [H. Evaluation Result](#)

A. Document History

Document History				
Entire Document	0.1	01/12/2015	Created Document	
Introduction	0.2	02/11/2015	First draft	S. Litzinger
Evaluation Details	0.3	03/25/2015	Vana	
Usage Information	0.3	03/25/2015		Vana
Installation Information	0.3	03/25/2015	Vana	
Evaluation Details	0.3	03/25/2015	Susan	
Installation information	0.3	03/31/2015	Susan	
Group Edit	0.4	04/10/2015	Group	
Group Edit	0.5	04/10/2015	Group	
Group Edit	0.6	04/15/2015	Group	
Group Edit	1.0	04/22/2015	Group	

B. Document Scope

This document is the evaluation report for the functional evaluation of NoMachine. The results of the evaluation and the collected information here are intended for the benefit of the XSEDE project and the general eScience community.

C. Executive Summary and Recommendation

NoMachine was selected for evaluation because campus bridging users showed a desire to have a utility to open and maintain an interactive graphical session (e.g., an NX remote desktop or X-Windows session) on a remote resource for a period of at least several days.

The evaluation was performed on the open source version of NoMachine, which is a remote desktop program which allows a user to interact with a machine as if they were sitting in front of it. NoMachine version 4.4.6 was evaluated.

NoMachine worked well for the most part. Most of the issues seen while testing were either related to using it over a limited network connection or when multiple users attempted to connect to the same machine. The open source version does not allow multiple users to connect to the same machine at the same time. There is a commercial version of NoMachine that contains that feature and more.

We recommend the open source version of NoMachine for single users connecting to a server.

D. Introduction

The Evaluation team was led by Peter Enstrom. The other members of the evaluation team were Susan Litzinger, Dan Lapine and Vanamala Venkataswamy. The evaluation was started on January 12, 2015. The evaluation finished on March 27, 2015.

NoMachine provides remote access that allows users to view and control their desktop from wherever they are. It allows them to run any graphic application through their NoMachine client using any network connection. On a reasonable network connection it performs well enough to allow them to use their desktop securely and as if they were sitting in front of it.

NoMachine is available for all variations of Linux as well as Mac, IOS and Windows. An Android version is also available but only for large-screened tablets (not for phones).

This evaluation covers the open source version of NoMachine. An enterprise version that is available for purchase was not covered during this evaluation.

E. Prerequisites

While NoMachine requires that the NX server be installed for all versions of Linux it is included as part of the installation package. It is not necessary for Windows. You will need to open ports in order for NoMachine to run, so authority to make those changes is necessary.

F. Evaluation Details

The requirements for this evaluation were:

1. Run for at least 1 week uninterrupted
2. Have a very high degree of reliability for maintaining open sessions
3. Support audio and video
4. Maintain a responsive, steady network connection
5. Update the screen reasonably quickly
6. Low latency mouse tracking and scrolling
7. Support dual monitor configurations
8. Allow file transfers between the server and client machine

In order to evaluate the software a test with four parts was created that detailed the actions for the evaluators. The results of their testing were captured in test logs that are linked on the wiki page for this evaluation. The first part tested the basic functionality of connecting to a remote desktop, viewing and controlling it. The second part of the test exercised NoMachine's more advanced features. The third part of the test introduced faults to see how NoMachine would handle them. The last part of the test investigated NoMachine's ability to maintain a long term connection to a desktop.

Testing took place on servers at NCSA, PSC and UVa. In order to test connections from a remote network, home machines were also used. A template virtual machine (VM) containing NoMachine was created on the NCSA OpenStack cluster. Virtual images on the cluster were created from the template and used in the testing as well. Multiple operating systems were used in testing: Windows, Mac OS, and various Linux flavors (CentOS, SuSe, and Ubuntu). Client connections were also tested from an iPad and a Nexus tablet. The remote desktops used for the test included basic desktops along with servers with high resolution and dual screens.

Test 1 : Basic Functionality

In this section we tested routine actions a user would expect to be able to perform on their desktop. In order to test how responsive the update of the desktop was we used a browser to display video with audio. This allowed us to easily detect any sluggishness in the output. We also tested cut and paste from the NoMachine window to a native window on the client machine. Also tested was NoMachine's feature that allows files to be passed between the server and the client. Each of these tasks were repeated using clients and servers running on different operating systems.

Steps for test 1

1. Select 4 different test machines, different types, different OSs.
2. Create accounts for 2 users, "tis" and "tis2".
3. Install NoMachine client/server on machines A and B which are on the same LAN.
4. Install client only on machine C which is on the same LAN.
5. Install client only on machine D which is outside the LAN.

6. Connect from A to B as user tis, pass a file from A to B, retrieve a file from B to A, open a browser and play video with sound, cut and paste text from the NoMachine window to a native window, disconnect.
7. Connect from C to B as user tis, pass a file from C to B, retrieve a file from B to C, open a browser and play video with sound, disconnect.
8. Connect from D to B as user tis, pass a file from D to B, retrieve a file from B to D, open a browser and play video with sound, disconnect.
9. Connect from A to B as user tis, connect from C to B as user tis, disconnect.
10. Connect from A to B as user tis, connect from C to B as user tis2, disconnect.

Results

File transfer between machines on the same LAN and between machines on different LANs all worked as expected. There were no noticeable speedups or delays when files were transferred between client/server machines on same LAN and client/server machines on different LANs. Copying and pasting the text between client/server machines worked seamlessly. Playing video on server machine and viewing it on client also worked. Sound was transferred from server to client machine. Connecting to the same server machine through two different client machines as the same user (user tis) disconnected the first user as it established the second connection without prompting the first user. This is the designed behavior of the open source version of NoMachine. Connecting to the same server machine from two different client machines as different users (users tis and tis2) failed. The connection from machine A to machine B as user tis was dropped as expected when user tis2 tried to connect to machine B, but the new connection from C to B was never established.

Test 2 : Advanced Functionality

In this section, we tested features beyond the basic set. By default a user establishes a NoMachine connection using a username and password. In this test an alternate login method was tested using private and public keys. The network was interrupted briefly to see how NoMachine handled the interruption. A longer interruption was then tested as well. We tested how NoMachine would react to a dual screen setup and a high resolution screen where more information is transferred. Also tested was how NoMachine reacted to chaining connections through more than one machine.

Steps for test 2

1. Select 4 different test machines (A, B, C and D), different types, different OSs.
2. Test machine C should have a dual screen setup.
3. Test machine D should have a high resolution screen (1900 x 1200 at least).
4. Create accounts for user "tis" on the machines.
5. Install NoMachine client/server on all 4 machines.
6. Connect from A to B as user tis using a private key, open a browser and play video with sound.
7. Briefly interrupt the network while connected, see if it recovers.
8. Interrupt the network for several minutes then restore.
9. Disconnect.

10. Connect from A to C as user tis, open a browser and play video with sound, disconnect.
11. Connect from A to D as user tis, open a browser and play video with sound, disconnect.
12. Connect from D to A as user tis, on machine A connect through no machine to machine B, open a browser and play video with sound, disconnect.

Results

NoMachine handled network interruptions well. A brief network interruption was not very noticeable to the user. With a longer network interruption the screen remained static until the network was restored at which point it became responsive once again. With 2 videos playing panning between screens slowed down but still worked OK. Chaining through 2 NoMachine connections did not cause any problems.

Test 3: Error Reporting

In this section we tested how NoMachine would handle induced error conditions and how it would report the problems to the user. We tested how NoMachine reacts when a second user attempts to connect to the same server that another user is already connected to when the new connection is denied. We tested switching between which remote user is connected to the server. We tested how NoMachine reacted to failed connection attempts due to trying to connect through a blocked port, attempting to connect to a non-existent server, and by connecting to a server that is not running the NoMachine server software. We tested having 2 users try to control the desktop at the same time, one from the server's desktop and the other remotely from a client connected to that desktop through NoMachine.

Steps for test 3

1. Set up machine A using SUSE Linux.
2. Select 2 different test machines (B and C), different types, different OSs.
3. Create accounts for user "tis" on the machines.
4. Install NoMachine client/server on all 3 machines.
5. Repeatedly connect from machine B to machine A with NoMachine, log into the KDM/GDM, log out, exit NoMachine, repeat several times. When done with the test, disconnect.
6. Log into A as yourself. Connect from B to A as user tis and accept the connection "read only" from the screen on A, open a browser and play video with sound. Connect from C to A as user tis2 and accept the connection "read only" from the screen on A. Does the server hang? When done with the test, log out of A and disconnect.
7. Block port 4000 in the firewall on machine A. Attempt to connect using the standard port 4000 from machine B to machine A. Attempt should fail. Unblock the port on machine A. Retry the connection, it should work. When done with the test, disconnect.
8. Try to connect from machine B to a non-existent machine. A meaningful error message should be displayed.
9. Terminate the NoMachine server on machine A. Attempt to connect from machine B to machine A. A meaningful error message should be displayed. Restart the server on A. Retry the connection, it should work. When done with the test, disconnect.
10. Connect from B to A as user tis, open a browser and play video with sound. Connect from C

to A as user tis, this user should take control. Alternate re-connecting from B to A then from C to A. It should work but we have encountered problems. When done with the test, disconnect.

11. Connect from B to A. Attempt to use the mouse and keyboard both remotely from B and locally from A at the same time. Should work.

Results

For the most part, NoMachine handled switching between users, whether accepting or rejecting the connections, well. However, the specific case tested in step 6 where 2 users attempt to connect in “view only” mode always caused the first connection to close as expected but the second connection was never established. Attempts to connect to a NoMachine server that failed gave adequate messages regarding the problem with the connection. Using the server’s desktop while someone else is connected through NoMachine worked fine but was disconcerting if they weren’t aware of the client control.

Test 4: Remote/Long-term Connection

In this test, we exercised long-distance, long-duration connections between clients and servers. Server machines were at NCSA and client machines were at UVa and PSC. Server machines were running YouTube videos in a loop to continually update the display to make it easy to verify that the connection was still active. Clients were connected to the servers and the connections were left alive for 7 days. We checked on the connections every day during the 7 days to make sure connections between the clients and servers were still active. We also tested playing OpenGL video with sound.

Steps for test 4

1. Select 4 different test machines (A, B, C and D), different types, different OSs.
2. Test machines A, B and C should be on the same LAN.
3. Test machine D should be off site.
4. Create accounts for user “tis” on the machines.
5. Install NoMachine client/server on all 4 machines.
6. Connect from A to B as user tis using a private key, open a browser and play video with sound. Leave the connection up for 7 days and check that the connection is still active daily during the test.
7. * Alternate to 6: Connect from A to B (should be linux server) as user tis using a private key, open a browser and play OpenGL video. Leave the connection up for 7 days and check that the connection is still active daily during the test.
8. Connect from C to D as user tis using a private key, open a browser and play video with sound. Leave the connection up for 7 days and check that the connection is still active daily during the test.

Results

Connections made on the same LAN, i.e client and server on the same LAN, had no issues and the connections stayed alive for all 7 days. There were some issues for one of the testers with long-

distance connections where they were not very stable for several attempts and were getting dropped after 12-24 hrs. However, everyone was able to keep the connections alive for 7 to 10 days in other trials.

Prior to the formal test, a NoMachine connection was established with a colleague in Tunisia and it worked well.

G. Installation Information

Visit the NoMachine home download page (<https://www.nomachine.com/download>) to retrieve the software. It's smart enough to know what type of machine you are on and offers up the correct version for you to download. Browse to the NoMachine download page [here](#), select the correct operating system.

For Windows, download the `nomachine_{version_no}.exe` file and execute it. Follow the prompts for installation. Following the installation, you will be asked to restart your computer to allow nomachine to run.

For Linux you will need to download and install the software with root privileges before initializing it and starting up the NoMachine server. For Ubuntu and OpenSuSE, download the tarball and extract to the `/usr/NX` directory. Run the `"/usr/NX/nxserver --install"` command to install NoMachine. For CentOS, download the RPM and then install the package by running: `"sudo rpm -i nomachine_<version>.rpm"`. At this point both the client, which allows connections to other machines, and the server which allows other machines to connect the server should be installed. Execute `"service nxserver start"` to bring the NoMachine server up and allow remote clients to connect to your desktop.

H. Usage Information

The open source version of NoMachine only allows one connection to a server at a time, even if multiple desktop sessions are running on the server. When a connection to NoMachine is made its behavior depends upon what login is active on the desktop and what login is being used to connect through NoMachine. If a different login is being used to connect, the NoMachine server will display a dialog on the desktop asking if the user wants to allow the remote connection. If, however, the same login is being used as is logged into the desktop, no popup is displayed and the connection is made.

If one user is connected to a NoMachine server and another user attempts to connect with the same login as the server's desktop is logged in as the original connection will close and the new connection will be established without warning to the first user. If the second connection is made with a different login, the first user will see a dialog and will be able to either accept or reject the new connection.

An apparent bug was discovered with the software when connecting to the NoMachine server in view only mode. If one user is connected in view only mode, a second connection is initiated and allowed also in view only mode, both connections are terminated and the NoMachine server becomes unresponsive for a time.

On some systems a slowdown was noticed when connected to a dual screen system that was playing video on each of its screens. NoMachine captures the screen display as a video stream, on systems with a lot of display real-estate this can slow things down, both in the encoding and transmission.

On a NoMachine server host, it is necessary open ports TCP 4000, UDP 4000,4300 for clients to communicate with the server. Note that networks can be filtering these ports out as well.

I. Evaluation Result

We recommend the open source version of NoMachine for single users connecting remotely to a server for a long duration.

NoMachine worked well for the most part. Most of the issues seen while testing it were either related to using it over a limited network connection or when multiple users attempted to connect to the same machine. The open source version does not allow multiple users to connect to the same machine at the same time. There is a commercial version of NoMachine that contains that feature.

The installation, and user interface for NoMachine were well documented and easy to use. NoMachines performed well, even over long distances.

There were times when the NoMachine server became unresponsive for a time, particularly after switching between connections from two different clients. Most of the time the server recovered after about 10 minutes but in some cases the service was not available until the NoMachine server process was restarted.

J. Suggestions for Evaluation Procedure Revisions

For this evaluation we created the Evaluation Report at the beginning of the evaluation process. We were able to share rough text within the team and edit the document in collaborative sessions which made writing the report easier. Including text snippets throughout the review process, even in rough draft format, allowed us to go back and make edits instead of starting from scratch after the review was complete.