Redundancy and Security Planning in Nagios XI 2024

Purpose

This document describes Redundancy and Security Planning in NagiosXI.

Documentation Purpose

To explain the purpose of this documentation a scenario is used as follows:

- Nagios XI server crashes unexpectedly
- A new Nagios XI server is restored from backup
- None of the agents accept monitoring requests from the new Nagios XI server
    - They do not have the new Nagios XI server address on the approved list
- Time is wasted updating the agent configurations

In this scenario it's obvious that an address should have been reserved and added to the agents configuration before the crash occurred. This documentation aims to highlight all the different settings for the agents so you can plan for such disasters.

The documentation also provides information on what needs to be changed on the Nagios XI server and the agents when a security breach occurred in your organization.

This documentation is repetitive, this is intended as the goal is to provide clear technical information.

User Macros

In relation to usernames, passwords and tokens, on the Nagios XI server it is more sensible to define these in user macros where possible. User macros provide the following functionality:

- Store sensitive information in a central location, masking the actual values being used
- When macros are used in the host and service objects, you are able to change the value of the macro and immediately that value is used by all objects that use the macro

Detailed information about user macros can be found in the Understanding The User Macros Component documentation.

NRDP

Nagios Remote Data Processor

Location:

Resides on the Nagios XI server

Documentation:

NRDP Overview

Purpose:

NRDP has the capability of allowing remote agents, applications, and Nagios instances to submit commands and host and service check results to a Nagios server

Configuration File:

```
/usr/local/nrdp/server/config.inc.php
```

Can be configured via **Admin > Check Transfers > Inbound Transfers > NRDP.**

Directives:

```
$cfg['authorized_tokens'] = array("xxx","yyy",);
```

This is an array of tokens (strings) that grant permission to incoming connections

Example:

```
$cfg['authorized_tokens'] = array("dhr8hbki9jse","NRDP_T0k3n",);
```

Removing a token will deny in incoming connections for those using the old token

If you are planning to phase out a token you can leave the old token in the array until all agents have been updated **Dependencies:**

- NCPA sending passive checks
- NSClient++ sending passive checks
- NRDS sending passive checks
- Nagios Log Server alerting
- Nagios Network Analyzer alerting

NSCA

Nagios Service Check Acceptor

Location:

Resides on the Nagios XI server

Documentation:

How To Use The NSCA Addon

Purpose:

NSCA has the capability of allowing remote agents, applications, and Nagios instances to submit commands and host and service check results to a Nagios server

Configuration File:

Nagios XI Server:

```
/etc/xinetd.d/nsca
/usr/local/nagios/etc/nsca.cfg
```

Can be configured via **Admin > Check Transfers > Inbound Transfers > NSCA**.

Remote Machine: send_nsca:

```
/usr/local/nagios/etc/send_nsca.cfg
```

NSClient++:

`C:\Program Files\NSClient++\nsclient.ini`

Directives:

password

The password that the agent provides to connect

Defined in both `nsca.cfg` and `send_nsca.cfg` or nsclient.ini

Example:

```
password=Str0ngP@ssw0rd
```

Notes:

Changing the password will deny in incoming connections for those using the old password

only_from

Space separated list of addresses that are allowed to connect to the Nagios XI server

Defined in `/etc/xinetd.d/nsca`

Example:

```
only_from  = 127.0.0.1 10.25.0.0/16
```

Notes:

Removing an address from the list will deny in incoming connections for those addresses.

If you are planning to phase out an address you can leave the old address in the list until all agents have been updated.

Dependencies:

- NSClient++ sending passive checks

NRPE

Nagios Remote Plugin Executor

Location:

Remote machine

Documentation:

Monitoring Hosts Using NRPE

Purpose:

NRPE is an agent and protocol that is often used to monitor Linux and Unix machines

It executes plugins on the remote machine as requested by the Nagios XI server

Configuration File:

```
/etc/xinetd.d/nrpe
/usr/local/nagios/etc/nrpe.cfg
```

Directives:

only_from

Space separated list of addresses that are allowed to connect to NRPE

Defined in *`/etc/xinetd.d/nsca`*

Example:

```
only_from = 127.0.0.1 10.25.0.0/16
```

Notes:

Removing an address from the list will deny in incoming connections for those addresses

If you are planning to phase out an address you can leave the old address in the list until all agents have been updated

allowed_hosts

Comma separated list of addresses that are allowed to connect to NRPE

Defined in `/usr/local/nagios/etc/nrpe.cfg`

This directive is ignored if `only_from` is used and NRPE is run via XINETD

Example:

```
allowed_hosts =127.0.0.1,::1,10.25.0.0/16
```

Notes:

Removing an address from the list will deny in incoming connections for those addresses

If you are planning to phase out an address you can leave the old address in the list until all Nagios servers have the correct addresses

```
ssl_client_certs
ssl_cert_file
ssl_privatekey_file
ssl_cacert_file
```

SSL/TLS certificates can be used with NRPE, this is explained in NRPE v3 Enhanced Security

Notes:

A common certificate authority should be used to issue certificates

Dependencies:

Nagios XI server monitoring hosts using check_nrpe

NSClient++

Location: Remote machine

Documentation:

Monitoring Windows Using NSClient++

Enabling The NRPE Listener In NSClient++ For Nagios XI

Using NSClient++ For Passive Checks

Purpose:

NSClient++ is an agent that is used to monitor Windows machines (Linux version also available)

It can provide active and passive monitoring

Active monitoring is provided by `check_nt` (simple) or `check_nrpe` (advanced) Passive monitoring allows the agent to return check results via **NRDP** or **NSCA**

Configuration File:

```
C:\Program Files\NSClient++\nsclient.ini
```

Directives (active):

allowed hosts

Comma separated list of addresses that are allowed to connect via `check_nt` or `check_nrpe`

Example:

```
allowed hosts = 127.0.0.1,10.25.5.11,10.25.5.12
```

Notes:

Removing an address from the list will deny in incoming connections for those addresses

If you are planning to phase out an address you can leave the old address in the list until all Nagios servers have the correct addresses

password

The password to allow incoming connections via `check_nt`

Example:

```
password = Str0ngP@ssw0rd
```

Notes:

Changing the password will deny in incoming connections for those using the old password

Core Config Manager Setting:

```
$ARG1$
```

Directives (passive):

address

The destination address of the Nagios server listening with **NRDP** or **NSCA**

Example:

NRDP

```
address = https://10.25.5.12/nrdp
```

NSCA

```
address = 10.25.5.12
```

Notes:

Requires that NSCA accepts incoming connections from this machine token

The token used to authenticate with the Nagios server listening with **NRDP**

Example:

```
token = Str0ngT0k3n
```

Notes:

Requires that NRDP has this token in it's list of allowed tokens password

The password required to connect to the Nagios server using **NSCA**

Example:

```
password = Str0ngP@ssw0rd
```

Notes:

Requires that NSCA has this password defined in it's configuration

Dependencies:

Nagios XI server monitoring hosts using `check_nt` or `check_nrpe`

Nagios XI server accepting passive check results with **NRDP** or **NSCA**

NCPA

Nagios Cross-Platform Agent

Location:

Remote machine

Documentation:

Installing NCPA

Monitoring Devices Using The NCPA Agent And Nagios XI

Purpose:

NCPA is an agent that is often used to monitor operating systems like Windows, Linux, AIX

It can provide active and passive monitoring

Active monitoring is provided by `check_ncpa.py`

Passive monitoring allows the agent to return check results via **NRDP**

Configuration File:

```
/usr/local/ncpa/etc/ncpa.cfg
C:\Program Files (x86)\Nagios\NCPA\etc\ncpa.cfg
```

Directives (active):

```
community_string
```

The token that will be used to authenticate requests through `check_ncpa.py`

Example:

```
community_string = Str0ngT0k3n
```

Notes:

Changing the token will deny in incoming connections for those using the old token

Core Config Manager Setting:

`$ARG1$` following the `-t` argument

Directives (passive):

parent

The destination address of the Nagios server listening with **NRDP**

Example:

```
address = https://10.25.5.12/nrdp/ token
```

The token used to authenticate with the Nagios server listening with **NRDP**

Example:

```
token = Str0ngT0k3n
```

Notes:

Requires that NRDP has this token in it's list of allowed tokens

Dependencies:

Nagios XI server monitoring hosts using `check_ncpa.py`

Nagios XI server accepting passive check results with **NRDP**

SSH

Secure Shell

Location:

Remote machine

Documentation:

Monitoring Hosts Using SSH

Purpose:

Using the `check_by_ssh` plugin you can monitor machines using the SSH protocol

Agent-less monitoring, however plugins are required to be installed on the remote machine

Configuration File:

```
/home/nagios/.ssh/id_rsa
```

The security key that Nagios XI server sends to the destination to authenticate

Notes:

This file resides on the Nagios XI server

If you update the security key file, the known_hosts file on the remote machine needs to include the updated id_rsa.pub file

```
/home/nagios/.ssh/known_hosts
```

This contains the contents of `/home/nagios/.ssh/id_rsa.pub` from the Nagios server This is how the remote machine validates the incoming connection request

Notes:

This file resides on the remote machine

Dependencies:

Nagios XI server monitoring hosts using `check_by_ssh`

SNMP

Simple Network Management Protocol

Location:

Remote machine/device

Documentation:

Monitoring Linux Using SNMP

Using The SNMP Walk Wizard

Purpose:

Using the `check_snmp` plugin (and others) you can monitor machines using SNMP Agent-less monitoring, does require configuration of the remote machine / device

Configuration File:

Linux / Unix:

```
/etc/snmp/snmpd.conf
```

Windows:

Configured through the services management console Network Devices:

Each network device that supports SNMP will have a configuration section Directives v1 & v2:

rocommunity

v1 & v2 of SNMP have a community string that is used to authenticate incoming connections

Example:

```
rocommunity Str0ngC0mmunity 10.25.5.12
```

Notes:

The string can be restricted to a specific IP address or a subnet of addresses

Core Config Manager Setting:

`$ARG1$` following the `-C` argument Directives v3:

```
net-snmp-create-v3-user
```

Use the `net-snmp-create-v3`-user command to define the configuration

Example:

```
net-snmp-create-v3-user -ro -a SHA -A Str0ng@uth3ntic@ti0n -x AES -X
Str0ngPriv@cy nagios
```

Notes:

SNMP v3 is a stronger authentication method, more detailed information on this can be found in the Monitoring Linux Using SNMP documentation

Core Config Manager Setting:

`$ARG1$` contains all of the v3 arguments

Notes:

Removing a v1/v2 community or v3 authentication method will deny in incoming connections for those methods

If you are planning to phase out a v1/v2 community or v3 authentication method you can leave the old method in the configuration until all Nagios servers have been updated

Dependencies:

Nagios XI server monitoring hosts/devices using the SNMP configuration wizards and plugins

SNMP Traps

Simple Network Management Protocol Trap Sending

Location:

Nagios XI server and remote machine/device

Documentation:

Integrating SNMP Traps With Nagios X

SNMP Trap Tutorial

Purpose:

SNMP Traps is when remote hosts/devices send trap messages to the Nagios server

Configuration File:

Linux / Unix:

```
/etc/snmp/snmptrapd.conf
/etc/snmp/snmptt.conf
```

Windows:

Configured through the services management console Network Devices:

Each network device that supports SNMP will have a configuration section

Directives:

The main directive you need to be aware of is the trap destination, the Nagios XI server

Notes:

Each host/device will have it's own method/reason why it is sending a trap

If your Nagios XI server address changes then you will need to update your devices to send to the new address

Dependencies:

Nagios XI must be configured to accept incoming SNMP Traps

NRDS

Nagios Remote Data Sender

Location:

Nagios XI server and remote machines

Documentation:

Passive Monitoring With NRDS

Purpose:

NRDS is an agent that is often used to monitor operating systems like Windows, Linux, AIX

It provides passive monitoring where the agent returns check results via **NRDP**

NRDS provides capability of auto-updating the agent configurations, allowing you to centrally manage the agents from the Nagios XI server

Configuration File:

```
/usr/local/nrdp/clients/nrds/nrds.cfg
```

Directives:

URL

The destination address of the Nagios server listening with **NRDP**

Example:

```
URL="https://10.25.5.12/nrdp/"
TOKEN
```

The token used to authenticate with the Nagios server listening with **NRDP**

Example:

```
TOKEN="Str0ngT0k3n"
```

Notes:

Requires that NRDP has this token in it's list of allowed tokens

Notes:

In Nagios XI the configs can be managed via **Admin > Monitoring Config > NRDS Config Manager**

When planning on changing a token make sure both the new and old token exist in NRDP to ensure a seamless transition

Dependencies:

Nagios XI server accepting passive check results with **NRDP**

MRTG

Multi Router Traffic Grapher

Location:

Nagios XI server

Documentation:

Nagios XI - Switch And Router Wizard Architecture

Purpose:

MRTG is a program that uses SNMP to poll network devices to monitor network bandwidth

It runs every five minutes and calculates the traffic difference from the last data poll and the current one

The Network Switch / Router configuration wizard creates the configuration file for you, however to change the configuration requires manual intervention

Configuration File:

```
/etc/mrtg/conf.d/*.cfg
```

Directives SNMP v1 & v2:

```
Target[10.25.13.15_2]: 2: snmp_community_string@10.25.13.15:161::::2
```

Lines that end with : 1 or :2 identify the SNMP version being used

The string before the @ is the SNMP community string used to connect to the device The number: before the community string is the port being targeted

Notes:

Every port being monitored has a Target line that requires updating if you change the community string on the remote device

Directives SNMP v3:

```
Target[centos10_2]: 2:public@centos10:::::3
SnmpOptions[centos10_2]: privpassword-
=>'Str0ngPriv@cy',authpassword=>'Str0ng@uth3ntic@ti0n',authprotocol=>'sha',privprotocol=>'aes',u\
sername=>'nagios'
```

Lines that end with :3 identify the SNMP version being used, while it defines a public community string described above it is not used

All off the SNMPv3 directives are on the SnmpOptions line (the example above is one long line, it simply wraps over three lines in this documentation)

Notes:

Every port being monitored has a SnmpOptions line that requires updating if you change the authentication settings on the remote device

SNMP v3 is a stronger authentication method, more detailed information on this can be found in the Monitoring Linux Using SNMP documentation **Core Config Manager Settings:**

When the Network Switch / Router configuration wizard creates services the following applies:

- There are no SNMP directives defined on the service objects for Bandwidth services
- Port Status services have the SNMP directives defined in `$ARG2$`

Dependencies:

Devices that have been configured for monitoring using the Network Switch / Router wizard

Finishing Up

This completes the documentation on Nagios XI redundancy and security planning. If you have additional questions or other support-related questions, please visit us at our Nagios Support Forum, Nagios Knowledge Base, or Nagios Library:

Visit Nagios Support Forum Visit Nagios Knowledge Base Visit Nagios Library