Purpose

This document describes how to setup encryption between Nagios Log Server and NXLog on Windows using self signed certificates.

Target Audience

This document is intended for use by Nagios Log Server Administrators who would like encryption between NLS and their Windows NXLog clients. Encryption ensures that the traffic between the Windows machine and Nagios Log Server is not sent in plain text.

Overview

This documentation is broken up into the following sections:

- Create Certificates on the Nagios Log Server
  - Create a Certificate Authority (CA)
  - Create a certificate for the Nagios Log Server
  - Create a certificate for the Windows NXLog client
- Copy New Certificates
- Create Firewall Rule
- Create Input in Nagios Log Server using the certificates
- Configure NXLog to use the certificates

Prerequisites

It is assumed that you already have NXLog installed on your Windows machine, the installation steps and client are available in Nagios Log Server by clicking **+ Add Log Source** on the navigation bar. The following documentation is available as well:

Sending Windows Logs To Nagios Log Server

Terminology

For your information:

- SSL = Secure Sockets Layer
- TLS = Transport Layer Security

TLS replaces SSL, however the tools used to implement both generally use SSL in their name/directives. For simplicity reasons, the rest of this document will use the term SSL.

The steps in this documentation will create a CA and that CA will sign two certificates. This allows Nagios Log Server to use the CA to trust that the certificates used by the source and destination are valid.

Global Config vs Per Instance

This documentation walks you through creating certificate files that will be used in the Logstash Input that is created.

If you define this Input in the **Global Config**, you will be required to place the certificate files on **ALL** of your Nagios Log Server instances. If you do not, the configuration will **NOT** be applied on the instances that do not have the certificate files. This means that the input configuration will never be updated on these instances.

If you do not wish to implement the certificates on each Nagios Log Server instance, you will need to create the Input as a **Per Instance** config for the instance that has the certificate files (this will be explained later).

Installing Necessary Components

Establish a terminal session to your Nagios Log Server and as root and execute the following command:

RHEL | CentOS | Oracle Linux

```
yum install -y mod_ssl openssl
```

Debian | Ubuntu

```
apt-get install -y openssl
```

All of the remaining steps will be performed from within the root user's home directory to ensure the files you create are not accessible to anyone except the root user. Change into the home directory with this command:

```
cd ~
```

You will continue to use this terminal session throughout this documentation.

Create Certificate Authority

First step is to generate the private key file, execute the following command:

```
openssl genrsa -out ca.key 2048
```

That would have generated some random text. Next you will generate a request and sign the key:

```
openssl req -x509 -new -nodes -key ca.key -sha256 -days 1024 -out ca.pem
```

You will need to supply some values, some can be left blank, the following is an example:

```
Country Name (2 letter code) [XX]:AU
State or Province Name (full name) []:NSW
Locality Name (eg, city) [Default City]:Sydney
Organization Name (eg, company) [Default Company Ltd]:My Company Pty Ltd
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:ca
Email Address []:
```

As you can see above, I did not supply an `Organizational Unit Name` or `email address`.

Create Nagios Log Server Certificate

Now you need to create a certificate for your Nagios Log Server instance(s). Execute the following command:

```
openssl genrsa -out device-nls.key 2048
```

That would have generated some random text. Next you will generate a request:

```
openssl req -new -key device-nls.key -out device-nls.csr
```

You will need to supply some values, some can be left blank, the following is an example:

```
Country Name (2 letter code) [XX]:AU
State or Province Name (full name) []:NSW
Locality Name (eg, city) [Default City]:Sydney
Organization Name (eg, company) [Default Company Ltd]:My Company Pty Ltd
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:nls
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

As you can see above, I did not supply an `Organizational Unit Name`, `email address`, `password` or optional `company name`. Specifically, providing a `password` is not necessary.

One more command is required to sign the key, execute the following command *(the following is one long command that has wrapped over two lines)*:

```
openssl x509 -req -in device-nls.csr -CA ca.pem -CAkey ca.key -CAcreateserial
-out device-nls.crt -days 500 -sha256
```

Which should produce output saying the Signature was OK and it was Getting Private Key.

Create Windows NXLog Certificate

Now you need to create a certificate for your Windows NXLog client. Execute the following command:

```
openssl genrsa -out device-nxlog.key 2048
```

That would have generated some random text. Next you will generate a request:

```
openssl req -new -key device-nxlog.key -out device-nxlog.csr
```

You will need to supply some values, some can be left blank, the following is an example:

```
Country Name (2 letter code) [XX]:AU
State or Province Name (full name) []:NSW
Locality Name (eg, city) [Default City]:Sydney
Organization Name (eg, company) [Default Company Ltd]:My Company Pty Ltd
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:nxlog
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
```

```
A challenge password []:
An optional company name []:
```

As you can see above, I did not supply an `Organizational Unit Name`, `email address`, `password` or optional `company name`. Specifically, providing a `password` is not necessary.

One more command is required to sign the key, execute the following command *(the following is one long command that has wrapped over two lines)*:

```
openssl x509 -req -in device-nxlog.csr -CA ca.pem -CAkey ca.key
-CAcreateserial -out device-nxlog.crt -days 500 -sha256
```

Which should produce output saying the Signature was OK and it was Getting Private Key.

Copy New Certificates

Use the following commands to copy the new certificates to the correct locations:

```
cp ca.key /etc/pki/tls/private/
cp device-nls.key /etc/pki/tls/private/
cp ca.pem /etc/pki/tls/certs/
cp device-nls.crt /etc/pki/tls/certs/
```

If you plan on creating the Input as part of the Global Config, you will need to copy these certificate files to all the instances in your Nagios Log Server cluster. Please refer to the Global Config vs Per Instance section of this document for more information.

Create Firewall Rule

You need to create a firewall rule to allow the incoming TCP traffic. In your terminal session execute the following commands (depending on your operating system version):

RHEL | CentOS | CentOS Stream | Oracle Linux

```
firewall-cmd --zone=public --add-port=7777/tcp
firewall-cmd --zone=public --add-port=7777/tcp --permanent
```

Debian:

The local firewall is not enabled on Debian by default and no steps are required here. **IF** it is enabled then the commands are:

```
iptables -I INPUT -p tcp --destination-port 7777 -j ACCEPT
```

Ubuntu:

The local firewall is not enabled on Ubuntu by default and no steps are required here. **IF** it is enabled then the commands are:

```
sudo ufw allow 7777/tcp
sudo ufw reload
```

If you plan on creating the Input as part of the Global Config, you will need to create this firewall rule on all the instances in your Nagios Log Server cluster.

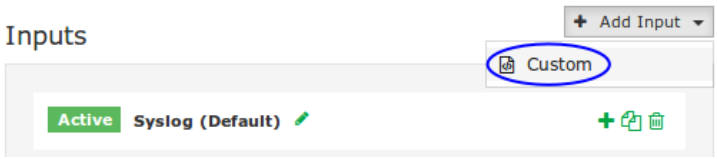
Create Input

This creates an Input that uses the certificates you have created and will be listening on TCP port 7777. Login to one of your Nagios Log Server instances as an Admin user Click **Configure** on the navigation bar.

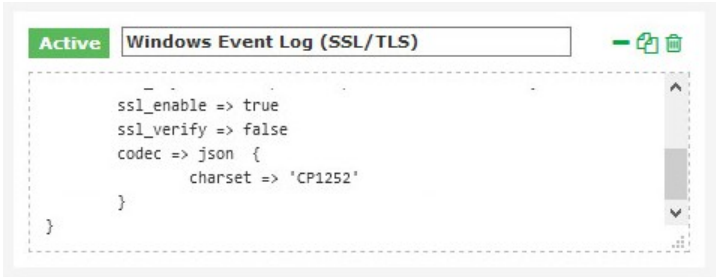
Select one of these options:

- For a Global Config
  - In the left pane under **Global (All Instances)** click **Global Config**
- For a Per Instance Config
  - In the left pane under **Per Instance (Advanced)** click the **Log Server Instance** which has the certificate files you created.

The remaining steps are common to either option.



On the right side of the screen there click the **+ Add Input** button and select **Custom**.
A new block appears at the bottom of the Inputs table.



Type a unique **name** for the input which will be **`Windows Event Log (SSL/TLS)`**.
In the text area field enter the following code (you can copy and paste):

```
tcp {
     port => 7777
     type => 'eventlog'
```

```
        ssl_extra_chain_certs => ['/etc/pki/tls/certs/ca.pem']
        ssl_cert => '/etc/pki/tls/certs/device-nls.crt'
        ssl_key => '/etc/pki/tls/private/device-nls.key'
        ssl_enable => true
        ssl_verify => false
        codec => json  {
             charset => 'CP1252'
        }
    }
```

If you have an version of Nagios Log Server before 1.5.0 then the `ssl_extra_chain_certs` line needs to be `ssl_cacert` instead, as per:

```
        ssl_cacert => '/etc/pki/tls/certs/ca.pem'
```

The `ssl_extra_chain_certs` option is an array which allows for multiple CA certs, this allows you to have a chain of CA certificates.

Click the **Save & Apply** button to create this filter and apply the configuration.

Configuring NXLog On Windows

The CA certificate and the NXLog certificate need to be copied to your Windows machine:

`/root/ca.pem`
copied to
`C:\Program Files (x86)\nxlog\cert\ca.pem`

`/root/device-nxlog.crt`
copied to
`C:\Program Files (x86)\nxlog\cert\device-nxlog.crt`

You could do this with a program like WinSCP or you could simply copy the contents of the files and paste them into Notepad on Windows and save the files with the correct filenames. You can view the contents of a certificate by using the `cat` command, for example:

```
cat /root/ca.pem
```

Once you've done this, open up the `C:\Program Files (x86)\nxlog\conf\nxlog.conf` file in Notepad. Find the section that looks like this and comment it out by adding a `#` at the beginning of each line:

```
#<Output out>
#    Module om_tcp
#    Host 10.25.5.99
#    Port 3515
#
#    Exec  $tmpmessage = $Message; delete($Message); rename_field("tmpmessage","message");
#    Exec  $raw_event = to_json();
#
#     # Uncomment for debug output
#     # Exec file_write('%ROOT%\data\nxlog_output.log', $raw_event + "\n");
#</Output>
```

In the example above, the `Host 10.25.5.99` line contains the IP Address of the Nagios Log Server. This address needs to be used in the new config section that you are going to add next.

The following is the new config section that needs to be put into the `nxlog.conf` file.

```
<Output out>
   Module          om_ssl
   Host            10.25.5.99
   Port            7777
   CertFile        C:\Program Files (x86)\nxlog\cert\device-nxlog.crt
   CAFile          C:\Program Files (x86)\nxlog\cert\ca.pem
   AllowUntrusted  TRUE
   Exec  $tmpmessage = $Message; delete($Message); rename_field("tmpmessage","message");
   Exec  $raw_event = to_json();
</Output>
```

Save the file and close Notepad.

Now you need to restart the nxlog service on the Windows machine. This can be done by executing the following commands in a Command Prompt with Administrative permissions:

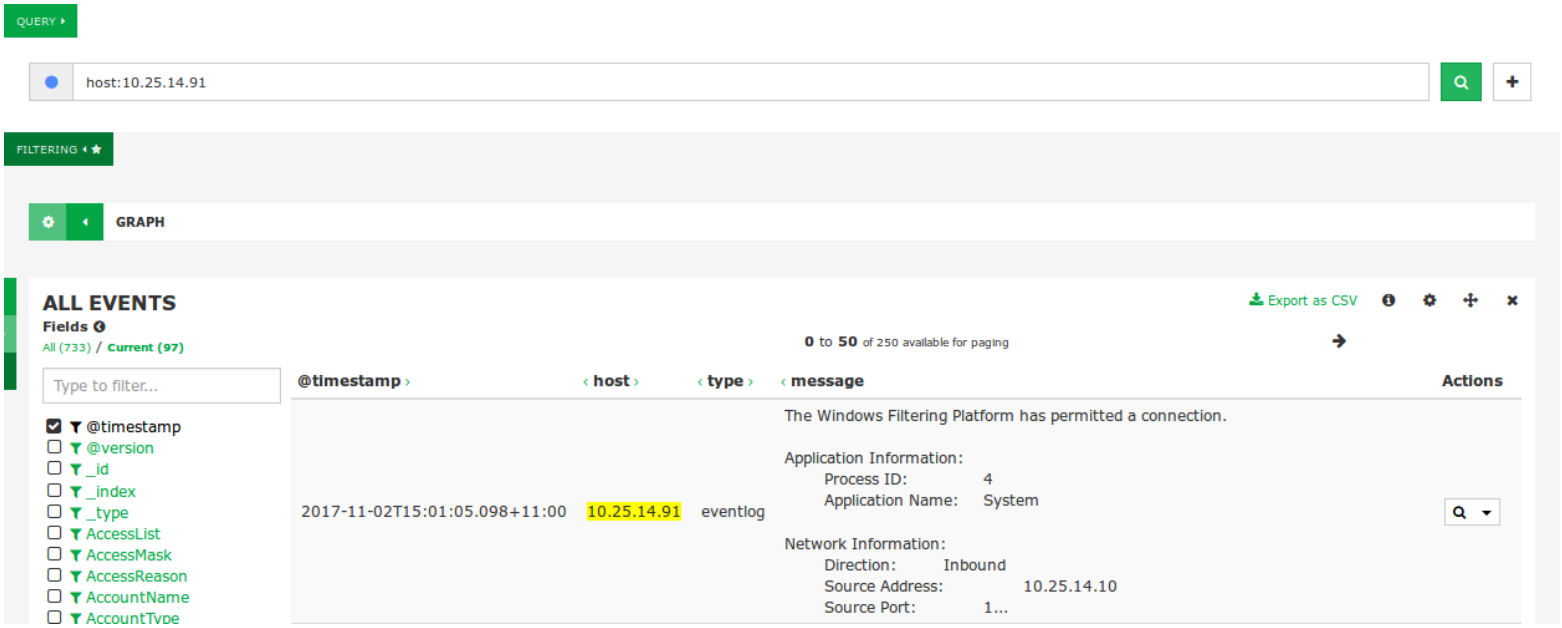
```
sc stop nxlog
sc start nxlog
```

Verify Incoming Logs

To confirm that Nagios Log Server is receiving data from the Windows server navigate to the **Dashboards** page. Perform a **Query** on the host field using the **IP Address** of your **Windows** host:

```
host:<Windows Host Address>
```

Here is an example that show the received logs appearing in the ALL EVENTS panel.



Additional Information

If you would like to verify that traffic is encrypted, you can verify this by using `tcpdump`. First you must have `tcpdump` installed on your Nagios Log Server which can be done with this command:

RHEL | CentOS | Oracle Linux

```
yum install -y tcpdump
```

Debian | Ubuntu

```
apt-get install -y tcpdump
```

Once installed execute the following command to observe the traffic:

```
tcpdump -i ens32 -nnvXSs 0 host 10.25.14.91
```

In that command, `ens32` is the network interface on the Nagios Log Server and `10.25.14.91` is the IP address of the Windows machine.

Here is example output **before** implementing SSL/TLS.

```
11:40:48.857072 IP (tos 0x0, ttl 128, id 31372, offset 0, flags [DF], proto TCP
(6), length 204)
    10.25.14.91.61978 > 10.25.5.99.3515: Flags [P.], cksum 0xc5dc (correct), seq
1015102624:1015102788, ack 1368467930, win 16425, length 164
     0x0000:  4500 00cc 7a8c 4000 8006 57c8 0a19 0e51  E...z.@...W....Q
     0x0010:  0a19 0555 f21a 0dbb 3c81 3ca0 5191 29da  ...U....<.<.Q.).
     0x0020:  5018 4029 c5dc 0000 7b22 4576 656e 7452  P.@)....{"EventR
     0x0030:  6563 6569 7665 6454 696d 6522 3a22 3230  eceivedTime":"20
     0x0040:  3137 2d30 342d 3138 2031 313a 3430 3a34  17-04-18.11:40:4
     0x0050:  3422 2c22 536f 7572 6365 4d6f 6475 6c65  4","SourceModule
```

You can see in the right hand side the plain text such as `"EventReceivedTime":"2017-04-18.11:40:44"`.

Here is example output **after** implementing SSL/TLS.

```
11:47:07.228206 IP (tos 0x0, ttl 128, id 1497, offset 0, flags [DF], proto TCP
(6), length 274)
    10.25.14.91.54713 > 10.25.5.99.7777: Flags [P.], cksum 0x3ac9 (correct), seq
4122608981:4122609215, ack 932892309, win 16074, length 234
     0x0000:  4500 0112 05d9 4000 8006 cc35 0a19 0e51  E.....@....5...Q
     0x0010:  0a19 0555 d5b9 1e61 f5ba 0555 379a ce95  ...U...a...U7...
     0x0020:  5018 3eca 3ac9 0000 1703 0100 204b f03e  P.>.:........K.>
     0x0030:  a312 5aa3 efc2 3cea 5830 4c8c 2983 f47a  ..Z...<.X0L.)..z
     0x0040:  dc67 3524 7961 dfb4 73de c64e b517 0301  .g5$ya..s..N....
```

```
0x0050:  00c0 459c 61e3 b309 b963 b3ab 599c 0b55  ..E.a....c..Y..U
0x0060:  221b c8dd 41e7 ffac 1b7a 6ba9 b5df 0dc5  "...A....zk.....
0x0070:  b902 827c 8076 5b83 7f6c 79f8 e57c ea6c  ...|.v[..ly..|.l
0x0080:  b628 e274 aa64 1b58 3348 39c2 856d 79ab  .(.t.d.X3H9..my.
0X0090:  1cc7 a825 016d 5b96 e155 4f6c 2b69 4fae  ...%.m[..UOl+iO.
0X00a0:  3704 d9f3 6302 39a9 fd4c 5020 839b 324f  7...c.9..LP...2O
```

You can see in the right hand side the data in encrypted and cannot be understood.

Finishing Up

This completes the documentation on how to send encrypted NXLogs with SSL/TLS to Nagios Log Server. If you have additional questions or other support related questions, please visit us at our Nagios Support Forums:

https://support.nagios.com/forum

The Nagios Support Knowledgebase is also a great support resource:

https://support.nagios.com/kb