Sending SysLog with SSL in Nagios Log Server 2024

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
- Copy New Certificates
- Create Firewall Rule
- Create Input in Nagios Log Server using the certificates
- Create Filter in Nagios Log Server to break the syslog data into fields
- Configure syslog to use the CA certificate

Prerequisites

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

Monitoring A New Log Source

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 a certificate. This allows the client to trust that the CA certificate used by the destination is 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 syslog-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 syslog-ca.key -sha256 -days 1024 -
out syslog-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) []:syslog-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 syslog-nls.key 2048
```

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

```
openssl req -new -key syslog-nls.key -out syslog-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) []:syslog-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 syslog-nls.csr -CA syslog-ca.pem -CAkey syslog-ca.key -CAcreateserial -out syslog-nls.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:

CentOS/RHEL

```
cp syslog-ca.key /etc/pki/tls/private/
```

```
cp syslog-nls.key /etc/pki/tls/private/
```

```
cp syslog-ca.pem /etc/pki/tls/certs/
```

```
cp syslog-nls.crt /etc/pki/tls/certs/
```

Ubuntu/Debian

```
cp syslog-ca.key /etc/ssl/private
```

```
cp syslog-nls.key /etc/ssl/private
```

```
cp syslog-ca.pem /etc/ssl/certs
```

```
cp syslog-nls.crt /etc/ssl/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 | Oracle Linux

```
firewall-cmd --zone=public --add-port=7778/tcp
```

```
firewall-cmd --zone=public --add-port=7778/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 7778 -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 7778/tcp
```

```
sudo ufw reload
```

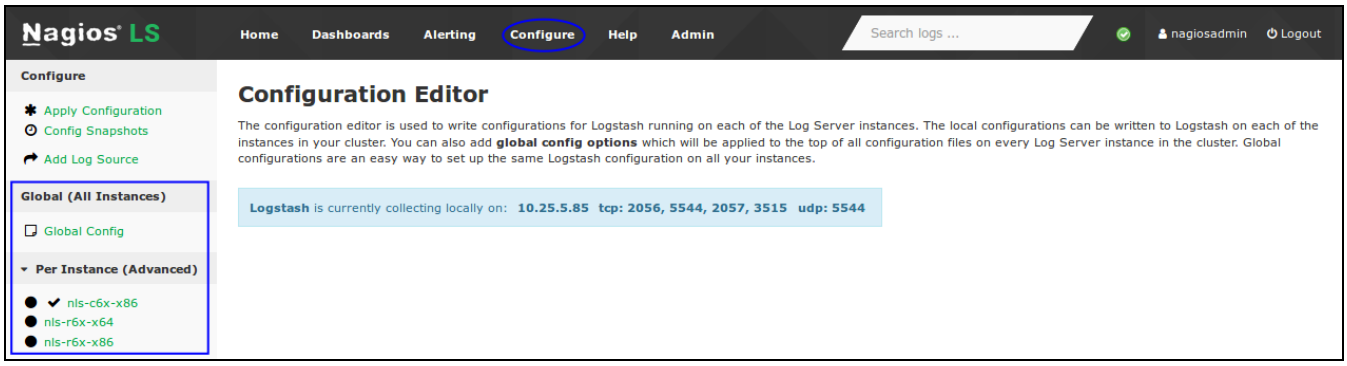
If you plan on creating the Input as part of the Global Config, you will need to create this firewall rule on all the instances in your Nagios Log Server cluster.

Create Input

> NOTE: The examples below use the certificate locations related to CentOS/RHEL systems. If you are working on a Ubuntu/Debian systems, see the Copy New Certificate section and replace with applicable locations in the commands.

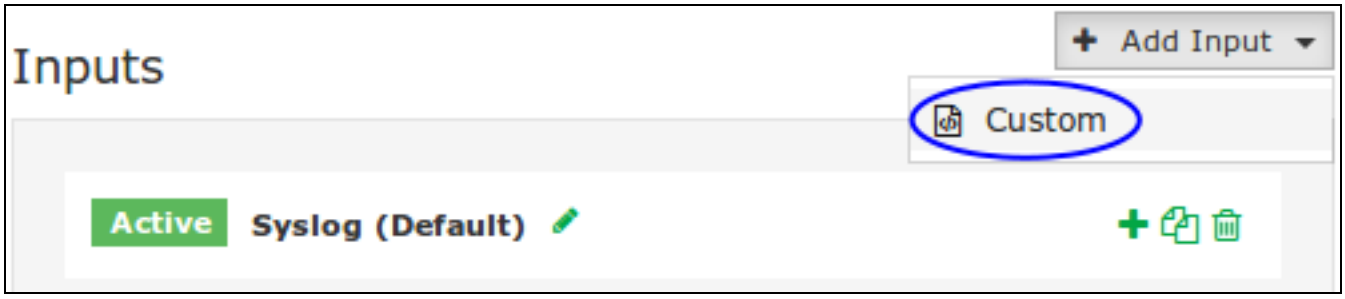
This creates an Input that uses the certificates you have created and will be listening on TCP port 7778.

1. Login to one of your Nagios Log Server instances as an Admin user Click Configure on the navigation bar.



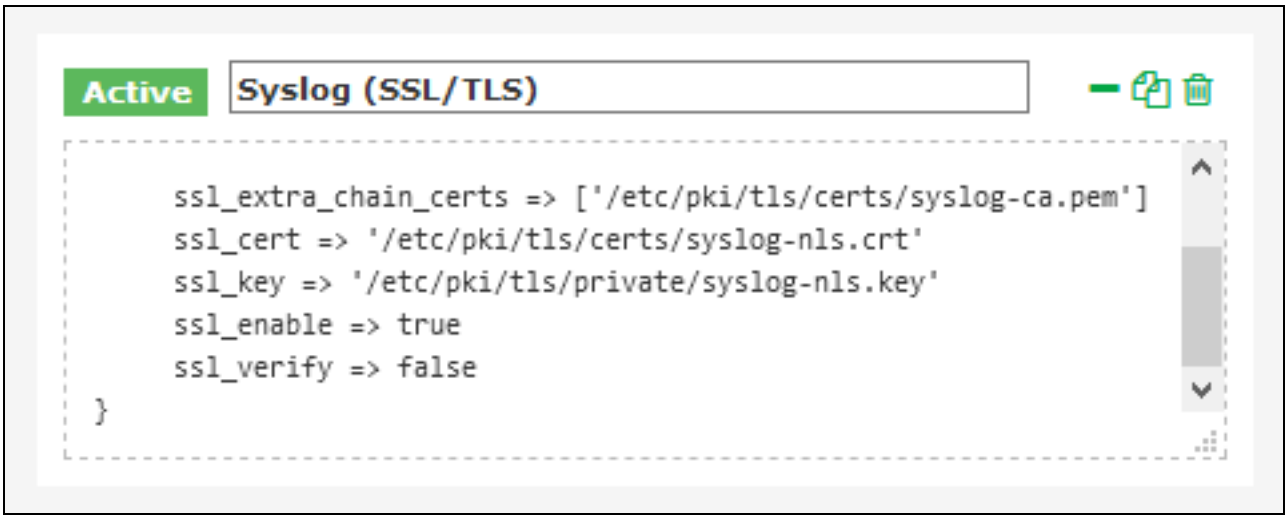
2. Select one of these options:

- 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.

3. 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.

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

```
tcp {
```

```
port => 7778
```

```
type => 'syslog_tls'
```

```
ssl_extra_chain_certs => ['/etc/pki/tls/certs/syslog-ca.pem']
```

```
ssl_cert => '/etc/pki/tls/certs/syslog-nls.crt'
```

```
ssl_key => '/etc/pki/tls/private/syslog-nls.key'
```

```
ssl_enable => true
```

```
ssl_verify => false
```

```
}
```

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:

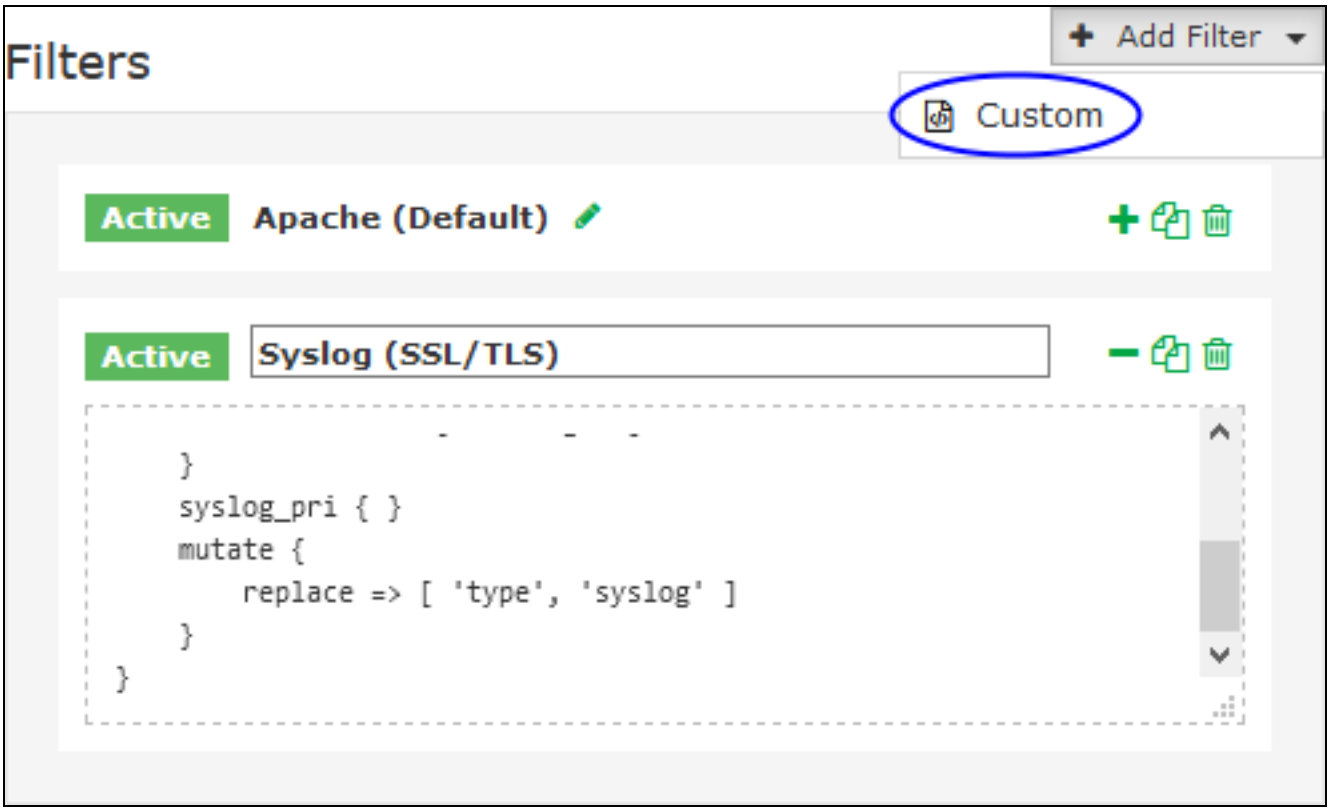
```
ssl_cacert => '/etc/pki/tls/certs/syslog-ca.pem'
```

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.

5. Click the Save button to create this input.

Create Filter

This creates a Filter that breaks the incoming syslog data into fields. While still on the Configure screen, on the right side of the page click the + Add Filter button and select Custom.



A new block appears at the bottom of the Filters table.

Type a unique name for the filter which will be Syslog (SSL/TLS). In the text area field enter the following code (you can copy and paste):

```
if [type] == 'syslog_tls' {
```

```
    grok {
```

```
        match => { "message" => "<%{POSINT:priority}>%
{SYSLOGTIMESTAMP:timestamp} %{SYSLOGHOST:hostname} %{DATA:program}(?:\
[%{POSINT:pid}\])?: %{GREEDYDATA:message}" }
```

```
        overwrite => [ 'message' ]
```

```
    }
```

```
    syslog_pri { }
```

```
    mutate {
```

```
        replace => [ 'type', 'syslog' ]
```

```
    }
```

```
}
```

The match => line and the two consecutive lines are one long line, it is wrapped over multiple lines in this documentation due to it being long (be careful with a copy and paste as it may be brought across as three separate lines).

Click the Save button to create this filter.

Verify And Apply Configuration

At this point you should click the Verify button to ensure the filter you just created is valid. Once the verify is successful you need to apply the configuration. In the left pane under Configure click Apply Configuration. Click the Apply button and then click

Yes, Apply Now when prompted.

Configuring syslog On Linux

You will need to install the TLS module for rsyslog on your client machine. In this example the client machine is CentOS 7 and the command to install these are:

```
yum install -y rsyslog-gnutls
```

Next the CA certificate needs to be copied to your Linux machine:

```
/root/syslog-ca.pem
```

copied to

```
/etc/pki/tls/certs/syslog-ca.pem
```

You could do this using the scp command, for example (executed on your Nagios Log Server):

```
scp /root/syslog-ca.pem root@client_address:/etc/pki/tls/certs/syslog-ca.pem
```

Keep in mind the scp command requires that openssh-clients is installed on both the source and destination machine.

Once you've done this, edit the /etc/rsyslog.d/99-nagioslogserver.conf file in vi with the following command:

```
vi /etc/rsyslog.d/99-nagioslogserver.conf
```

When using the vi editor, to make changes press i on the keyboard first to enter insert mode. Press Esc to exit insert mode.

Several lines need to be added to the beginning of the configuration and the port on the remote host line needs to be changed to 7778. Here is an example, the additions / changes are bolded:

```
### Begin forwarding rule for Nagios Log Server
```

```
$DefaultNetstreamDriver gtls
```

```
$ActionSendStreamDriverMode 1
```

```
$ActionSendStreamDriverAuthMode x509/certvalid
```

```
$DefaultNetstreamDriverCAFile /etc/pki/tls/certs/syslog-ca.pem
```

```
$WorkDirectory /var/lib/rsyslog
```

```
$ActionQueueFileName nlsFwdRule0
```

```
$ActionQueueHighWaterMark  8000
```

```
$ActionQueueLowWaterMark 2000
```

```
$ActionQueueMaxDiskSpace 1g
```

```
$ActionQueueSaveOnShutdown on
```

```
$ActionQueueType LinkedList
```

```
$ActionResumeRetryCount -1
```

```
# Remote host is: name/ip:port, e.g. 192.168.0.1:514, port optional
```

```
*.* @@nagios_log_server_address:7778
```

```
### End of Nagios Log Server forwarding rule
```

When you have finished, save the changes in vi by typing:

```
:wq
```

and press Enter.

Now you need to restart the syslog service on the Linux machine. On CentOS 7 this is accomplished by executing:

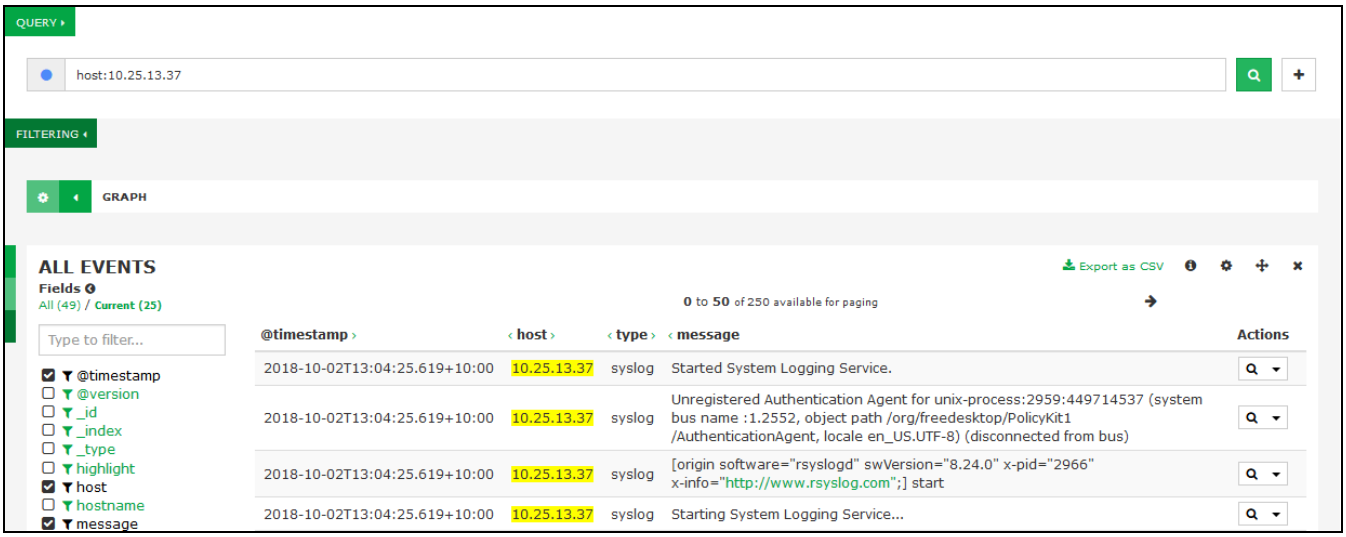
```
systemctl restart rsyslog
```

Verify Incoming Logs

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

host:<Linux 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 eth0 -nnvXSs 0 host 10.25.13.37
```

In that command, eth0 is the network interface on the Nagios Log Server and 10.25.13.37 is the IP address of the Linux machine.

Here is example output before implementing SSL/TLS.

13:22:42.122028 IP (tos 0x0, ttl 64, id 24258, offset 0, flags [DF], proto TCP (6), length 800)

10.25.13.37.40086 > 10.25.5.84.5544: Flags [P.], cksum 0x987e (correct), seq 2705226557:2705227305, ack 2060624355, win 229, options [nop,nop,TS val 202974717 ecr 428191577], length 748

0x0000: 4500 0320 5ec2 4000 4006 b26b 0a19 0d25 E...^.@.@..k...%

0x0010: 0a19 0554 9c96 15a8 a13e 7b3d 7ad2 a1e3 ...T.....>{=z...

0x0020: 8018 00e5 987e 0000 0101 080a 0c19 25fd .....~.....%.

0x0030: 1985 af59 3c33 303e 4f63 7420 2032 2031 ...Y<30>Oct..2.1

0x0040: 333a 3232 3a34 3220 6365 6e74 6f73 3139 3:22:42.centos19

0x0050: 2073 7973 7465 6d64 3a20 5374 6f70 7069 .systemd:.Stoppi

0x0060: 6e67 2053 7973 7465 6d20 4c6f 6767 696e ng.System.Loggin

You can see in the right hand side the plain text such as ".systemd:.Stopping.System.Loggin."

Here is example output after implementing SSL/TLS.

13:20:45.359313 IP (tos 0x0, ttl 64, id 50753, offset 0, flags [DF], proto TCP (6), length 1312)

10.25.5.84.7778 > 10.25.13.37.60328: Flags [P.], cksum 0x2bbd (incorrect -> 0xe766), seq 3447726428:3447727688, ack 1805938821, win 122, options [nop,nop,TS val 428074815 ecr 202857919], length 1260

```
0x0000:  4500 0520 c641 4000 4006 48ec 0a19 0554  E....A@.@.H....T
0x0010:  0a19 0d25 1e62 eba8 cd80 215c 6ba4 7085  ...%.b....!\k.p.
0x0020:  8018 007a 2bbd 0000 0101 080a 1983 e73f  ...z+..........?
0x0030:  0c17 5dbf 1603 0304 e702 0000 4d03 035b  ..].........M..[
0x0040:  b2e4 0d91 32b7 b042 7d58 5d39 1a18 1ab8  ....2..B}X]9....
0x0050:  b6e2 223d 0093 9e69 fe26 2bd0 f606 3b20  .."=...i.&+...;.
0x0060:  5bb2 e40d 830e e57c b620 6ffb 805d 20b4  [......|..o..]..
0x0070:  6dc5 956b 1245 e955 d014 a8a2 dada fef1  m..k.E.U........
0x0080:  c013 0000 05ff 0100 0100 0b00 0341 0003  .............A..
0x0090:  3e00 033b 3082 0337 3082 021f 0209 00d0  >..;0..70.......
0x00a0:  3c8a 52cc a810 b530 0d06 092a 8648 86f7  <.R....0...*.H..
```

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