Implementation of STARTTLS OVER SMTP PORT 25

At present the applications hosted under different environments like Java, PHP, Dot Net etc at NIC National Data Centers and State Data Centers are sending application generated mails by using SMTP protocol without any encryption while transferring the mails between client to relay MTA server and also without any authentication. The mails are generated by public IP numbers of NICNET / NKN or Private IP segments of various Data Centers.

For more secure approach, STARTTLS is implemented with authentication while sending mails from various applications using port 25. The following document gives information about how to make changes in the applications for implementation of STARTTLS over SMTP with authentication.

To relay the mails securely over port 25 from the application servers, the TLS communication and LDAP authentication are enforced to be implemented. To have a compatible communication between the relay clients and relay server, users are requested to incorporate the TLS as well as user credentials at the application level by making changes in the code.

The process of mail flow after enabling SSL over port 25 comprises of below steps:

1. The IP address of the relay client is verified against the white listed IP addresses in the relay server.
2. Client of relay server initiates the connection on TLSv1.2
3. Client of relay server passes the NIC LDAP credentials while connecting with relay server and in turn these credentials would get authenticated with NIC LDAP by relay server.
4. The mail from relay client is accepted by the relay server.

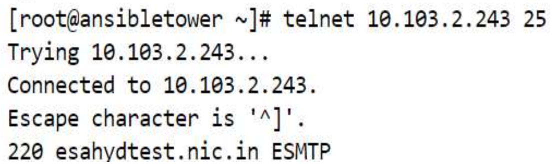
In the above 4 steps, each step would be implemented only on the successful implementation of the previous step.

Application Owner should test the following before making changes in the code

1) On confirmation from Relay team, application owners may have to test the connectivity to relay server from the command prompt as below and ensure the relay server is connected on port 25

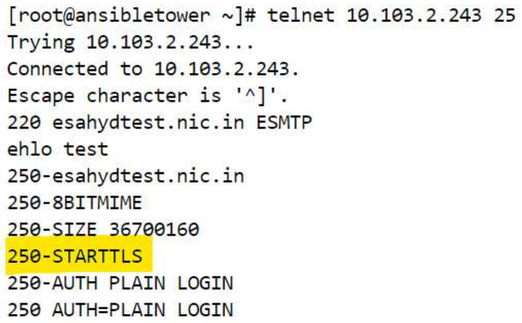
```
# telnet relay.nic.in 25
```

Please find the below example for successful output.
Note: IP provided here in the example is for testing purpose and not to be used for testing. Please use the above domain (relay.nic.in) for connectivity.



```
[root@ansibletower ~]# telnet 10.103.2.243 25
Trying 10.103.2.243...
Connected to 10.103.2.243.
Escape character is '^]'.
220 esahydtest.nic.in ESMTP
```

After the above line enter “elho test” and see the output as below.



```
[root@ansibletower ~]# telnet 10.103.2.243 25
Trying 10.103.2.243...
Connected to 10.103.2.243.
Escape character is '^]'.
220 esahydtest.nic.in ESMTP
ehlo test
250-esahydtest.nic.in
250-8BITMIME
250-SIZE 36700160
250-STARTTLS
250-AUTH PLAIN LOGIN
250 AUTH=PLAIN LOGIN
```

If the connection is successful, then we can see the line **“250-STARTTLS”.**

If the above step is not successful, then contact mail services at the mail address mailservices@nic.in to confirm about white listing of your IP in the relay MTA.

2) After successful connectivity to relay server, application owners need to test the SSL feature over port 25 by sending mail through the user specific application. The following are the attributes that are to be provided while sending the mail:
   - Hostname
   - LDAP enabled User name
   - LDAP User Password
   - From Address (Mail ID from where test mail is to be initiated)
   - To Address (Mail ID to whom the mail has to reach)
3) Application wise sample code is provided for easy testing of the functionality. Sample code for Java and PHP are provided here in this document. Go through the below content for the Steps to achieve the functionality based on the application used.

Steps while using Java Code:

1) Sample code for java is attached along with this document with name MailAuthSMTP.java. Fill the required details in the file which are marked in between the start and end tags (“<>”)

2) Compile the java file using the below command. Ensure the compilation is successful. If any exceptions arise, then rectify the issues based on the exception messages suggested.
   ```
   # javac MailAuthSMTP.java
   ```
3) After successful compilation of the file, execute the java class file using the below command.
   ```
   # java MailAuthSMTP
   ```
4) Check relevant logs for the mail status.

Note: Expecting that the relevant libraries are imported as per requirement to ensure that the Java code gets compiled without any exceptions.

Sample Java Code is provided as attachment for user reference.

Steps while using PHP Code:

a) Sample code for PHP is attached along with this document with names example.php and mail.php. The CLASS files for supporting SMTP feature through PHP are also included in the attachment. Place all the four files in relevant location. Fill the required details in the files example.php and mail.php in relevant locations (indicated inside the start and end tags "<>").
b) Execute the PHP example.php file. If any exceptions arise, then rectify the issues based on the exception messages suggested.
   ```
   # php example.php
   ```
c) Check relevant logs for the mail status.

Sample PHP Code is provided as attachment for user reference.

Steps while using DotNet Code:

d) Sample code for DotNet is attached along with this document with name DotNetSendEmailwithSSL_TLS_25.txt. Use the code in the relevant path and fill the required details in the file (indicated inside the start and end tags "<>").
e) Execute the file. If any exceptions arise, then rectify the issues based on the exception messages suggested.
f) Check relevant logs for the mail status.

Sample DotNet Code is provided as attachment for user reference.

Steps while using Python Code:

g) Sample code for Python is attached along with this document with name PythonSendEmailwithSSL_TLS_25.txt. Use the code in the relevant path and fill the required details in the file (indicated inside the start and end tags “<>”).

h) Execute the file. If any exceptions arise, then rectify the issues based on the exception messages suggested.

   #python3 –V ---To check version of Python

   #python3.6 <PythonFileName> (Ex: sendMail.py)

i) Check relevant logs for the mail status.

Sample Python Code is provided as attachment for user reference.