ITarian Endpoint Manager

On-Premise Deployment Guide

Table of Contents

1. Endpoint Manager On-Premise Deployment Guide

This guide explains how to deploy Endpoint Manager by ITarian on customer premises.

OS support

- Windows
- OS X
- Linux
- iOS
- Android

Features

- Manage endpoints (Windows/Linux/Mac) and mobile devices (iOS/Android)
- Remote package installation
- Antivirus and other advanced security features
- Manage users and user groups
- Easily deploy configuration templates
- File log / activity / verdict management
- Remote file and process management (Windows)
- Remote control of managed endpoints (Windows/Mac)
- Run remote procedures and monitor endpoint events
- Wipe Apple and Android devices
- Update CCC and CCS agents from internal cache in order to optimize internet bandwidth and accelerate updates in large networks.

See the full list of features at **https://dm.comodo.com/**

Guide Structure

This guide will take you through the installation and configuration of Endpoint Manager.

- **How it works**
- **Hardware requirements**
- **Network communication**
- **Firewall requirements**
- **Software requirements**
- **DNS requirements**
- **SSL requirements**
- **Export certificate for use on Endpoint Manager and Tigase server**
- **Installation via installer**
- **Manual installation**
- **Manual upgrade**
- **SMTP settings**

2. How it Works

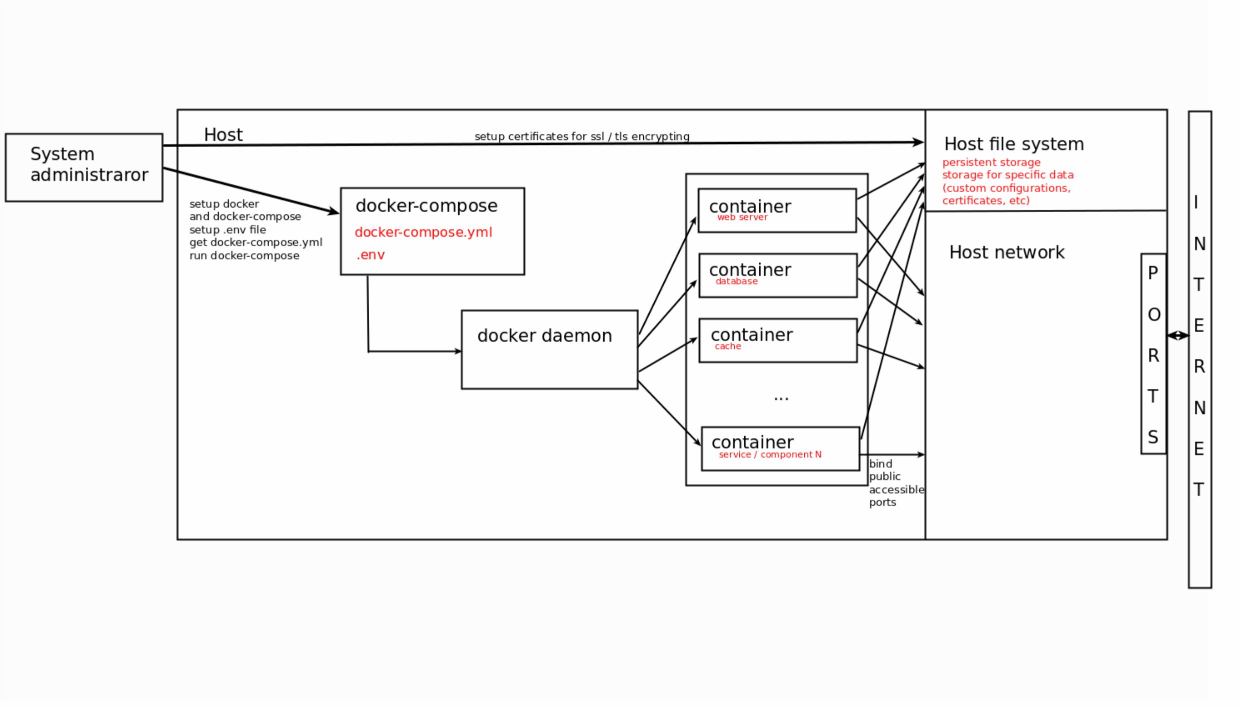
Endpoint Manager (EM) on-premise solution is distributed as a set of docker images and can be run with **docker**. Docker containers and **docker-compose** are mandatory to deploy EM on your premises.

Docker compose tool facilitates easy setup all components together and to maintain the environment.

To deploy EM on premises, system administrators should:

- Install docker
- Install docker-compose
- Setup configuration (basic domain name)
- Setup certificates (for domain above)
- Run all together with single docker-compose command

Scheme of docker-compose usage:



Docker-compose supports a stand-alone configuration for a single server. Docker-compose supports up to 1000 endpoints.

For larger deployments with multiple servers, use a more complicated docker management system like **Kubernetes**.

- Note - Support for Kubernetes is currently in-development.

3. Hardware Requirements

Hardware requirements / recommendations (1000 endpoints):

Minimum 2 servers for docker-compose configuration.

ITSM-server (EM server)

- 8 Cores Cpu | 8 Gb Ram | 50 Gb Hdd

Tigase-server

- 4 Cores Cpu | 4 Gb Ram | 30 Gb Hdd

Statistics: 1 endpoint produces 0.015 requests per second. This means we can handle about 65 sequential requests from different endpoints per seconds.

A server can handle 50-100 connections simultaneously. Therefore, the average endpoint count that can be handled is 50 * 65 > 3000.

4. Network Communication

The on-premise installation consists of multiple services and components which communicate with each other.

Public listen ports

Endpoint Manager (ITSM) Server

- **80 HTTP** - web port (redirects to https port 443 by default). Port 80 is only used for non-https browser connections.
- 443 HTTPS - common port which handles all incoming connections with TLS encryption

Tigase (xmpp) Server

- **443 TCP** – Secured TCP connection for endpoints and remote control tools.
- **5222 TCP** - Default XMPP port with the same purposes but not used. Might be used as a fallback option for 443.
- **8080 HTTP** - Service port for sending push messages. It is only used by the Endpoint Manager server and can be closed for external connections.

Turn server

- **49152 - 65535 UDP** – Dynamically allocated port range for remote control connections to endpoints located behind the NAT

Private network

Besides public ports most services expose specific ports to internal network which is closed to external world. These ports could be exposed just for debug purposes, but by default all service ports are closed including databases, message brokers and microservices which are the part of all system.

5. Firewall Requirements

- The Endpoint Manager system is designed for restricted environments which have an almost fully closed network.
- Therefore, it only exposes 443 port as main secure channel.
- Port 80 is used only for convenient redirects as the most popular default web port for each domain.

- Port 443 is also used for XMPP connections to Tigase, handling TCP traffic rather than HTTP.
- To summarize, we need to have port 443 open on the firewall as a minimum requirement. We also recommend Port 80 is left open for compatibility reasons.

6. Software Requirements

- The on-premise version was tested on Ubuntu Desktop and Ubuntu Server (Ubuntu 16.04.4 LTS). The scripts in this document were prepared for and tested on Ubuntu 16.
- Other versions of Ubuntu were not tested, but should deployment should still work on Ubuntu 14 and up (...maybe even Ubuntu 12 and up).
- The deployment will most likely work on other versions of Linux too (Debian, CentOS, etc). The only real difference is how to install the docker.
- For docker-compose configuration it doesn't matter which hostname is specified for each server.

7. DNS Requirements

- Endpoint Manager (ITSM) requires several domain names which should be resolved by different components.
- Endpoint Manager (ITSM) requires a minimum of one base domain and about 10 subdomains on the same level that should be resolved by different components. Otherwise you have to specify each required subdomain on every endpoint according to infrastructure.
- Basic DNS domain should be set by customer. But there are few requirements for existing domains / subdomains..

List of required domains:

- Base domain
- Itsm-domain
- Xmpp-domain
- Rmm-domain
- Patch-Management-domain
- Audit-log-domain
- Download-domain
- RealtimeDeviceCommunication-API-domain
- RealtimeDeviceCommunication-Relay-domain
- BulkInstallationPackage-domain

Base domain is just a pointer for all another subdomains.

Example:

ITSM-server IP	**10.0.5.1**
Tigase-server IP	**10.0.5.2**
Turn-server IP	**10.0.5.3**

Assume we have ITSM domain **on-prem.company.local** on IP **10.0.5.1** (ITSM-server).

It means that base domain is **company.local** (doesn't matter which ip it has. This entry not used in the system).

Next subdomains must be related to base domain.

Rmm-domain - **rmm-api.company.local** (IP **10.0.5.1** same as ITSM-server)

Patch-Management-domain - **plugins-api.company.local** (IP **10.0.5.1** same as ITSM-server)

Audit-log-domain - **auditlogs-api.company.local** (IP **10.0.5.1** same as ITSM-server)

Download-domain - **dl.company.local** (IP **10.0.5.1** same as ITSM-server)

Xmpp-domain - **xmpp.company.local** (IP **10.0.5.2** tigase-server)

RealtimeDeviceCommunication-API-domain - **rtdc-api.company.local** (IP **10.0.5.1** same as ITSM-server)

RealtimeDeviceCommunication-Relay-domain - **rtdc-relay-01.company.local** (IP **10.0.5.1** same as ITSM-server)

BulkInstallationPackage-domain - **bip.company.local** (IP **10.0.5.1** same as ITSM-server)

Required subdomain list which should be resolved

From Endpoint Manager (ITSM) -server

- xmpp (to tigase-server)

From Tigase-server

- <ITSM> - customer specified ITSM_DOMAIN (to ITSM-server)

From administrator endpoint (web access)

- <ITSM> - customer specified ITSM_DOMAIN (to ITSM-server)
- rtdc-api (to ITSM-server)
- rtdc-relay-01 (to ITSM-server)

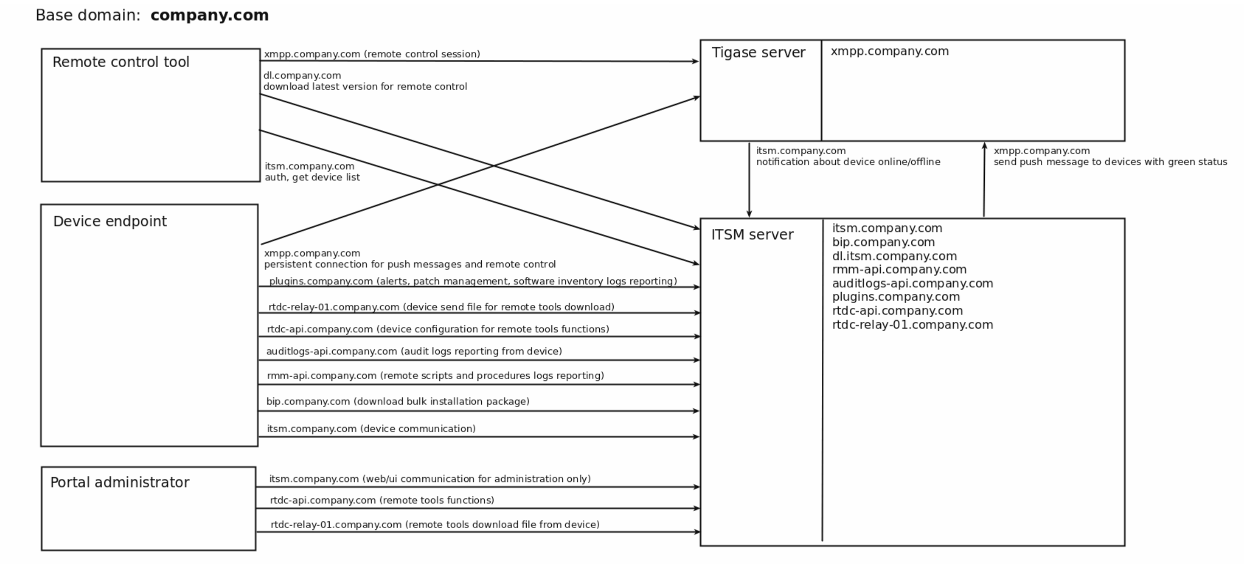
From enrolled device endpoint

- <ITSM> - customer specified ITSM_DOMAIN (to ITSM-server)
- bip - bulk installation package download host (to ITSM-server)
- rmm-api - rmm logs reporting (to ITSM-server)
- auditlogs-api - audit logs reporting (to ITSM-server)
- rtdc-api - remote tools configuration (to ITSM-server)
- rtdc-relay-01 - remote tools download file (to ITSM-server)
- plugins - alerts, patch management, software inventory logs reporting (to ITSM-server)
- xmpp - persistent connection for receiving push messages and remote control commands (to tigase-server)

From remote control tool

- <ITSM> - customer specified ITSM_DOMAIN (to ITSM-server)
- dl - check and download updates (to ITSM-server)
- xmpp - remote control communication (to tigase-server)

Scheme example:


Base domain: **company.com**
Remote control tool
xmpp.company.com (remote control session)
dl.company.com
download latest version for remote control
itsm.company.com
auth, get device list
Tigase server
xmpp.company.com
itsm.company.com
notification about device online/offline
xmpp.company.com
send push message to devices with green status
Device endpoint
xmpp.company.com
persistent connection for push messages and remote control
plugins.company.com (alerts, patch management, software inventory logs reporting)
rtdc-relay-01.company.com (device send file for remote tools download)
rtdc-api.company.com (device configuration for remote tools functions)
auditlogs-api.company.com (audit logs reporting from device)
rmm-api.company.com (remote scripts and procedures logs reporting)
bip.company.com (download bulk installation package)
itsm.company.com (device communication)
ITSM server
itsm.company.com
bip.company.com
dl.itsm.company.com
rmm-api.company.com
auditlogs-api.company.com
plugins.company.com
rtdc-api.company.com
rtdc-relay-01.company.com
Portal administrator
itsm.company.com (web/ui communication for administration only)
rtdc-api.company.com (remote tools functions)
rtdc-relay-01.company.com (remote tools download file from device)

8. SSL Requirements

- We recommend you to use a wildcard certificate.
- You may to have certificates for each subdomain specified above.
- Currently required to have wildcard ssl certificate for ITSM-server *.company.local and the same certificate or specific one for tigase xmpp.company.local.
- If you don't have real domain and trusted SSL certificate you can generate self-signed certificates by yourself.
- **Note**: Endpoints couldn't work with self-signed certificate on tigase. In this case you will never get green online status on ITSM-server for endpoints and also remote control will not work.

For minimal configuration it is required to have set of certificates and keys for each server (ITSM, tigase).

Setup SSL certificates for Endpoint Manager (ITSM)

Place valid SSL certificate and key into /opt/itsm/web/certs under the names cert.crt and cert.key.

Note: Private key must be without passphrase as web server could not work with those.

```
# create directory
sudo mkdir -p /opt/itsm/web/certs

# copy prepared certificate and key to destination
cp /path/to/your/certificate.crt /opt/itsm/web/certs/cert.crt
cp /path/to/your/certificate.key /opt/itsm/web/certs/cert.key
```

If you don't have valid certificates:

It is possible to issue self-signed certificate key-pair.

But in this case you need to allow unsecured access in the browser and some features will be dropped.

Next commands create self-signed certificates:

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
/opt/itsm/web/certs/cert.key -out /opt/itsm/web/certs/cert.crt
```

Note: Your certificate CN (common name) domain should be the same as ITSM_DOMAIN variable value specified in .env.

Setup SSL certificates for TIGASE

Place valid pem certificate for domain specified in TIGASE_DOMAIN into /opt/tigase/certs.

Filename should be in following pattern {TIGASE_DOMAIN}.pem.

Note:

- Certificate name should be exactly as TIGASE_DOMAIN value specified in .env file. For example, above certificate filename should be yourdomain.com.pem without prefix “xmpp.”
- Certificate bundle must contain root CA certificate. For creation valid certificate need to concatenate private.key + certificate.crt + chain.crt + root.crt

```
sudo mkdir -p /opt/tigase/certs
cat cert.key cert.crt chain.crt root.crt > your.domain.pem
sudo mv your.domain.pem /opt/tigase/certs/
```

9. Export Certificate for Use on Endpoint Manager and Tigase Server

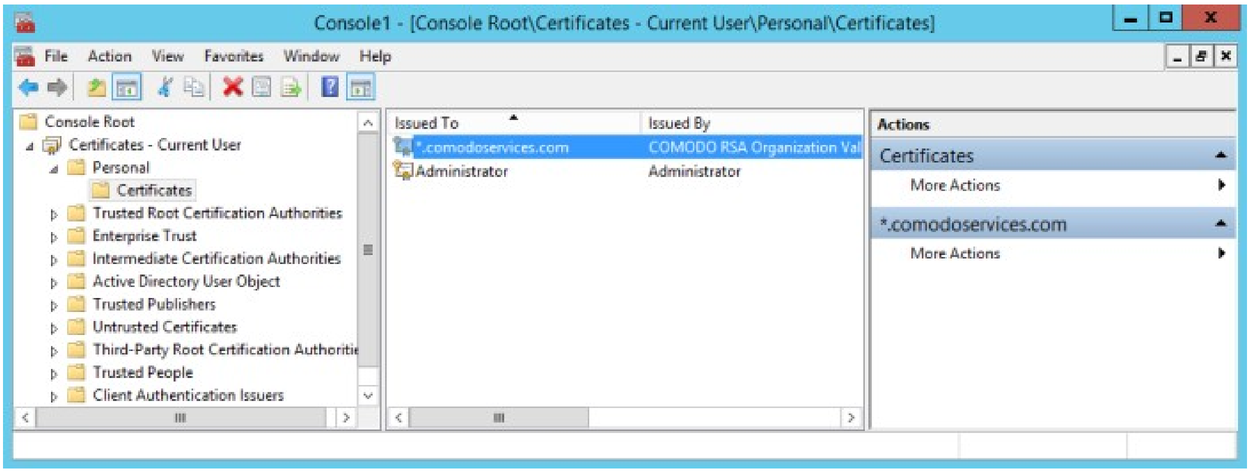
Export Certificate from Windows to .pfx format:

- **Step 1: View certificate information using mmc.exe**
- **Step 2: Export the certificate to .pfx format**

Step 1 - View certificate information using mmc.exe

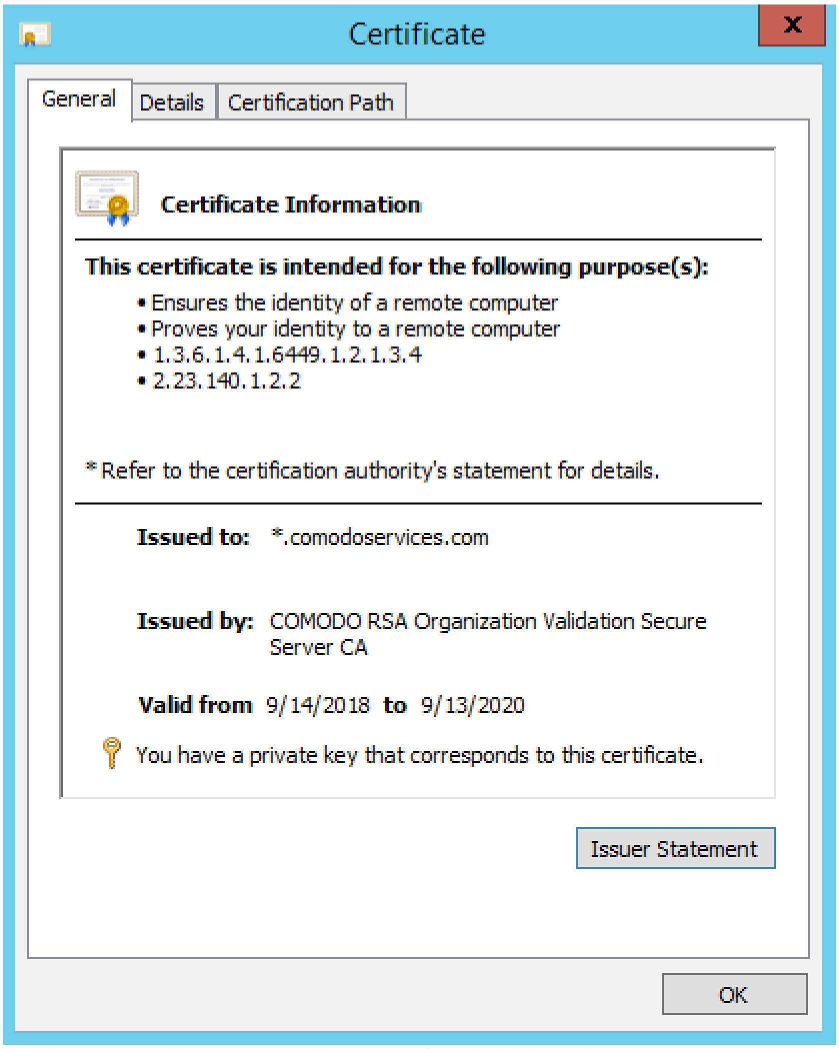
- Note: Assume you have installed a certificate onto your machine where
  - Right-click the 'Start' > 'Run'
  - Type mmc.exe > hit 'Enter'

  From 'Console Root'
  - Click 'File' then 'Add' / 'Remove Snapin…'
  - Select the 'Certificates' from the list and click 'Add'
    If you are not sure whether or not the certificate is under a user or a computer account, add them both
  - Click 'OK' to load the interface
  - Browse the certificate you want to use (usually is under the 'Personal' > 'Certificates' folder


Console1 - [Console Root\Certificates - Current User\Personal\Certificates]
File
Action
View
Favorites
Window
Help
Console Root
Certificates - Current User
Personal
Certificates
Trusted Root Certification Authorities
Enterprise Trust
Intermediate Certification Authorities
Active Directory User Object
Trusted Publishers
Untrusted Certificates
Third-Party Root Certification Authoritie
Trusted People
Client Authentication Issuers
Actions
Certificates
More Actions
*.comodoservices.com
More Actions

Issued To	Issued By
*.comodoservices.com	COMODO RSA Organization Val
Administrator	Administrator

- Double-click the certificate name to open to view its detailed information



Certificate Information

This certificate is intended for the following purpose(s):

- Ensures the identity of a remote computer
- Proves your identity to a remote computer
- 1.3.6.1.4.1.6449.1.2.1.3.4
- 2.23.140.1.2.2

* Refer to the certification authority's statement for details.

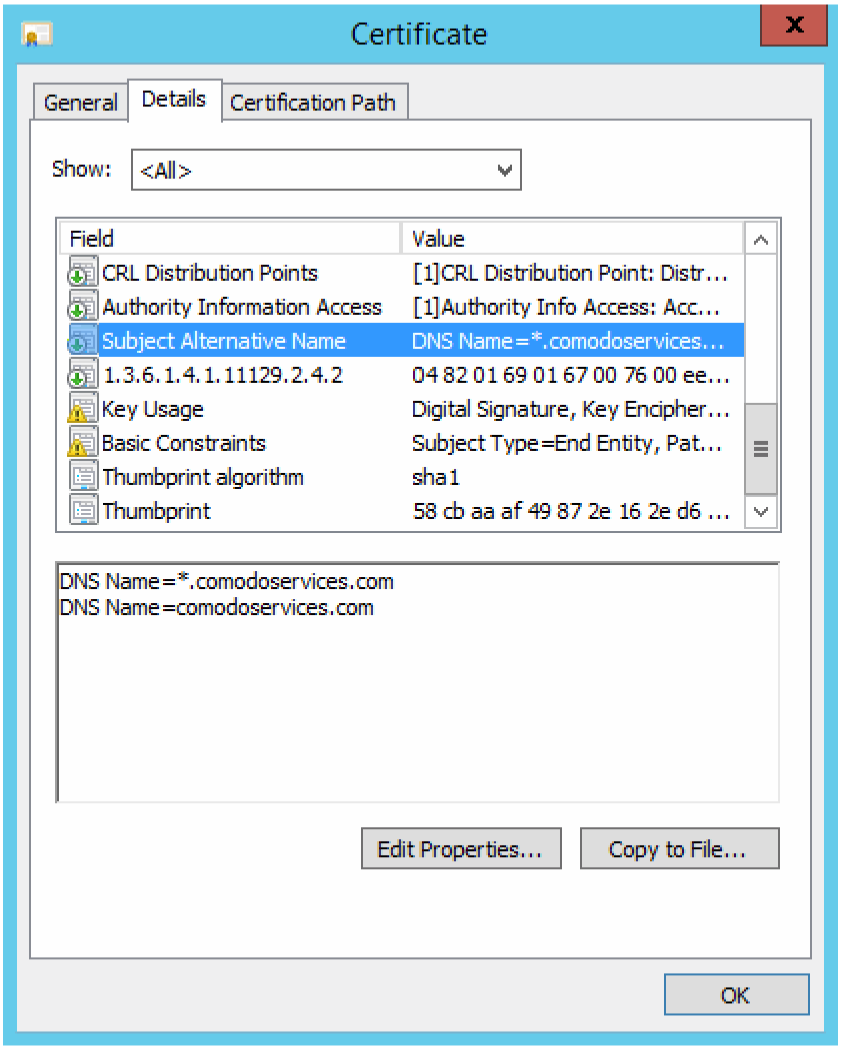
Issued to: *.comodoservices.com

Issued by: COMODO RSA Organization Validation Secure Server CA

Valid from 9/14/2018 **to** 9/13/2020

You have a private key that corresponds to this certificate.

- 'General' tab - view the certificate private key associated with it and validate the certificate, anything.comodoservices
  - You will not be able to validate comodoservices.com
- 'Details' tab > select 'Subject Alternative Name' section to confirm the validation

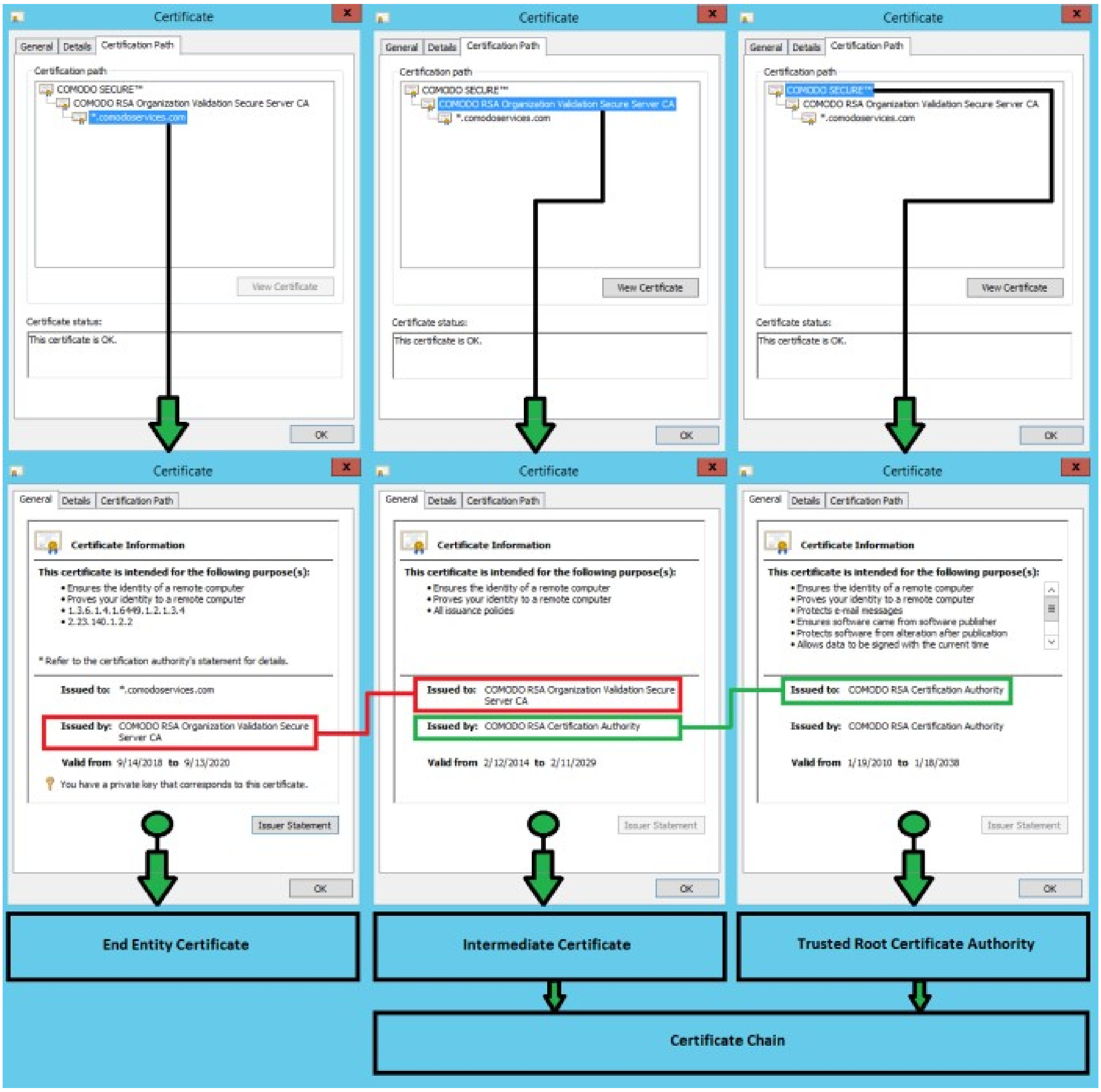


Show: <All>

Field	Value
CRL Distribution Points	[1]CRL Distribution Point: Distr...
Authority Information Access	[1]Authority Info Access: Acc...
Subject Alternative Name	DNS Name=*.comodoservices...
1.3.6.1.4.1.11129.2.4.2	04 82 01 69 01 67 00 76 00 ee...
Key Usage	Digital Signature, Key Encipher...
Basic Constraints	Subject Type=End Entity, Pat...
Thumbprint algorithm	sha1
Thumbprint	58 cb aa af 49 87 2e 16 2e d6 ...

```
DNS Name=*.comodoservices.com
DNS Name=comodoservices.com
```

- There are two entries:
  - The first entry - validate anything.comodoservices.com
  - The second entry - validate the main domain comodoservices.com.
  - If you are using a multidomain certificate, you can see all the FQDN's/IP's that the certificate is able to cover. In our case this wildcard certificate will suit our needs.
- 'Certificate Path' tab - view the certificate chain and confirm that the End Entity Certificate is able to link to a trusted root certificate using one or two intermediate certificates.
- In our example, one intermediate certificate:



Certification path

- COMODO SECURE™
  - COMODO RSA Organization Validation Secure Server CA
    - *.comodoservices.com

Certificate status:

This certificate is OK.

Certificate Information

This certificate is intended for the following purpose(s):

- Ensures the identity of a remote computer
- Proves your identity to a remote computer
- 1.3.6.1.4.1.6449.1.2.1.3.4
- 2.23.140.1.2.2

* Refer to the certification authority's statement for details.

Issued to: *.comodoservices.com

Issued by: COMODO RSA Organization Validation Secure Server CA

Valid from 9/14/2018 **to** 9/13/2020

You have a private key that corresponds to this certificate.

Certificate Information

This certificate is intended for the following purpose(s):

- Ensures the identity of a remote computer
- Proves your identity to a remote computer
- All issuance policies

Issued to: COMODO RSA Organization Validation Secure Server CA

Issued by: COMODO RSA Certification Authority

Valid from 2/12/2014 **to** 2/11/2029

Certificate Information

This certificate is intended for the following purpose(s):

- Ensures the identity of a remote computer
- Proves your identity to a remote computer
- Protects e-mail messages
- Ensures software came from software publisher
- Protects software from alteration after publication
- Allows data to be signed with the current time

Issued to: COMODO RSA Certification Authority

Issued by: COMODO RSA Certification Authority

Valid from 1/19/2010 **to** 1/18/2038

End Entity Certificate

Intermediate Certificate

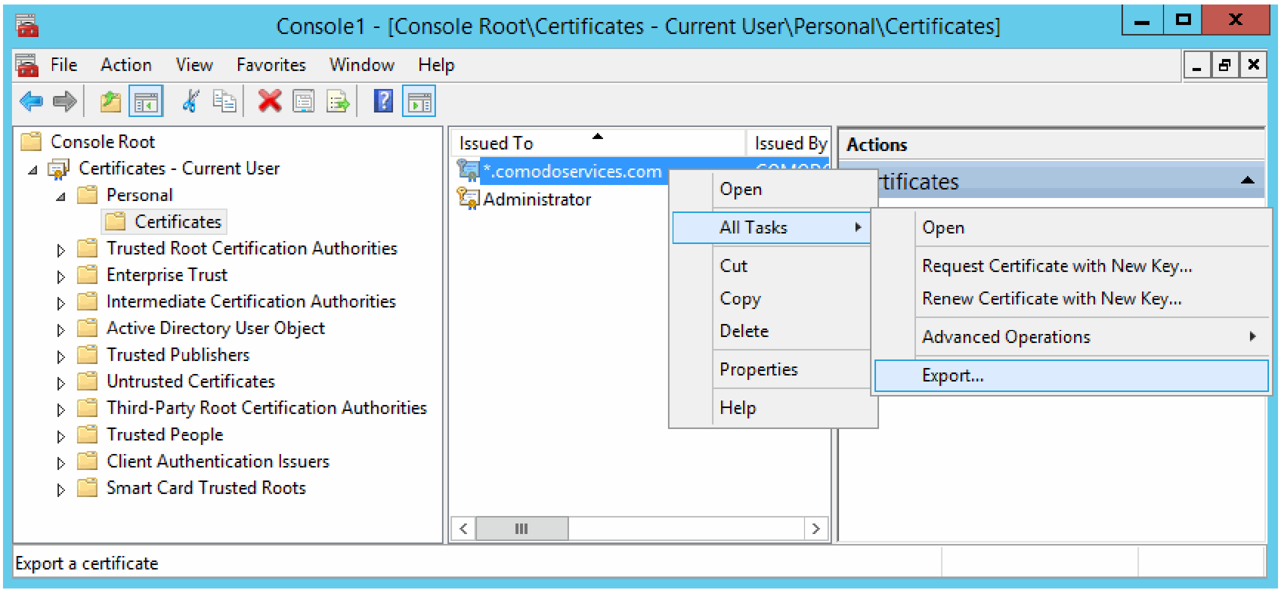
Trusted Root Certificate Authority

Certificate Chain

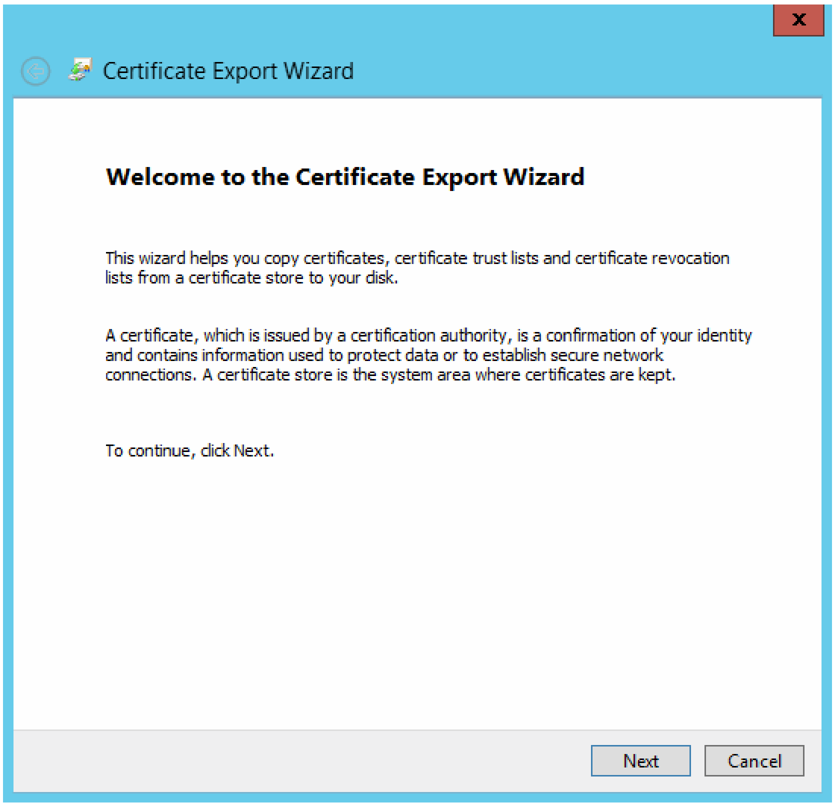
Step 2: Export the certificate to .pfx format

From the 'Certificates' window

- Right-click the certificate name > 'All Tasks' > 'Export...'


Console1 - [Console Root\Certificates - Current User\Personal\Certificates]
File Action View Favorites Window Help
Console Root
Certificates - Current User
Personal
Certificates
Trusted Root Certification Authorities
Enterprise Trust
Intermediate Certification Authorities
Active Directory User Object
Trusted Publishers
Untrusted Certificates
Third-Party Root Certification Authorities
Trusted People
Client Authentication Issuers
Smart Card Trusted Roots
Issued To
Issued By
*.comodoservices.com
Administrator
Actions
rtificates
Open
All Tasks
Cut
Copy
Delete
Properties
Help
Open
Request Certificate with New Key...
Renew Certificate with New Key...
Advanced Operations
Export...
Export a certificate

- Click Next


Certificate Export Wizard

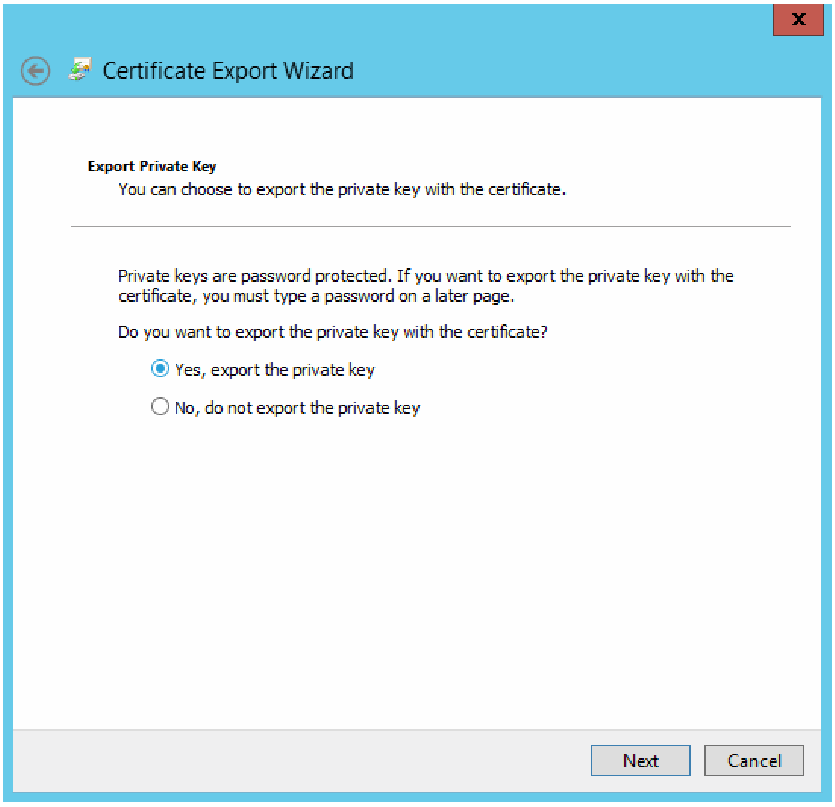
Welcome to the Certificate Export Wizard

This wizard helps you copy certificates, certificate trust lists and certificate revocation lists from a certificate store to your disk.

A certificate, which is issued by a certification authority, is a confirmation of your identity and contains information used to protect data or to establish secure network connections. A certificate store is the system area where certificates are kept.

To continue, click Next.

- Please select “Yes, export the private key” > click 'Next'



Certificate Export Wizard

Export Private Key

You can choose to export the private key with the certificate.

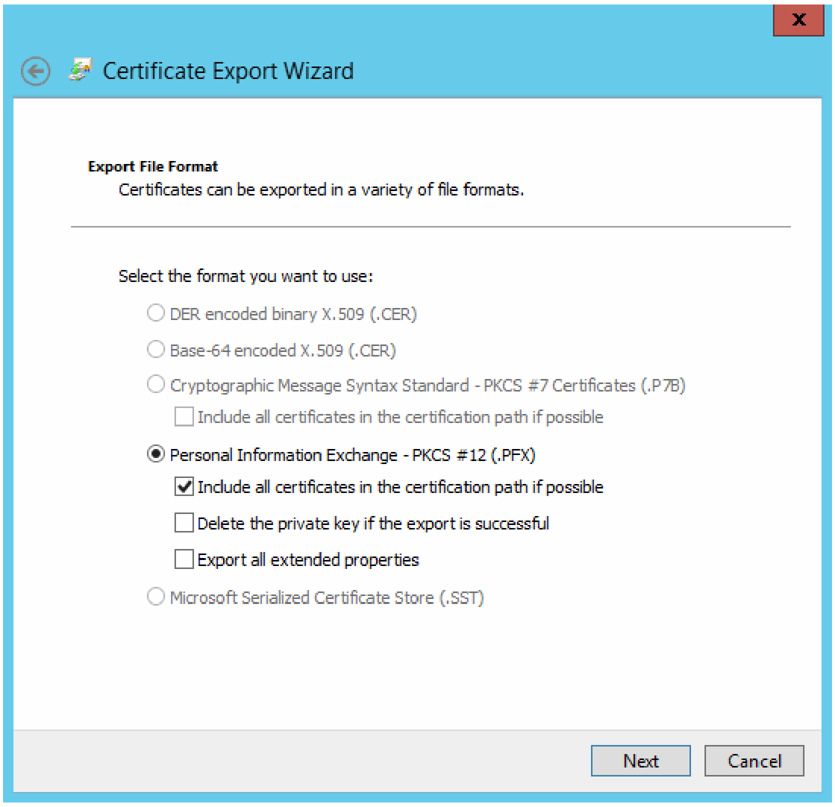
Private keys are password protected. If you want to export the private key with the certificate, you must type a password on a later page.

Do you want to export the private key with the certificate?

● Yes, export the private key

○ No, do not export the private key

- Select “Include all certificates in the certification path if possible” > click 'Next' to include the certificate chain



Certificate Export Wizard

Export File Format

Certificates can be exported in a variety of file formats.

Select the format you want to use:

○ DER encoded binary X.509 (.CER)

○ Base-64 encoded X.509 (.CER)

○ Cryptographic Message Syntax Standard - PKCS #7 Certificates (.P7B)

□ Include all certificates in the certification path if possible

● Personal Information Exchange - PKCS #12 (.PFX)

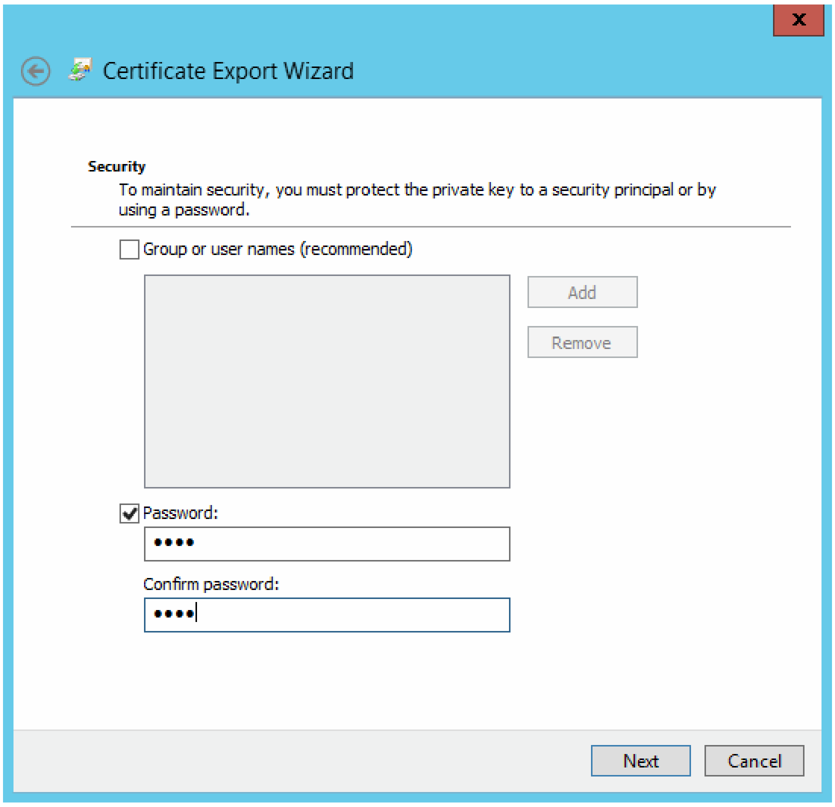
☑ Include all certificates in the certification path if possible

□ Delete the private key if the export is successful

□ Export all extended properties

○ Microsoft Serialized Certificate Store (.SST)

- Provide a password > click 'Next'



Certificate Export Wizard

Security

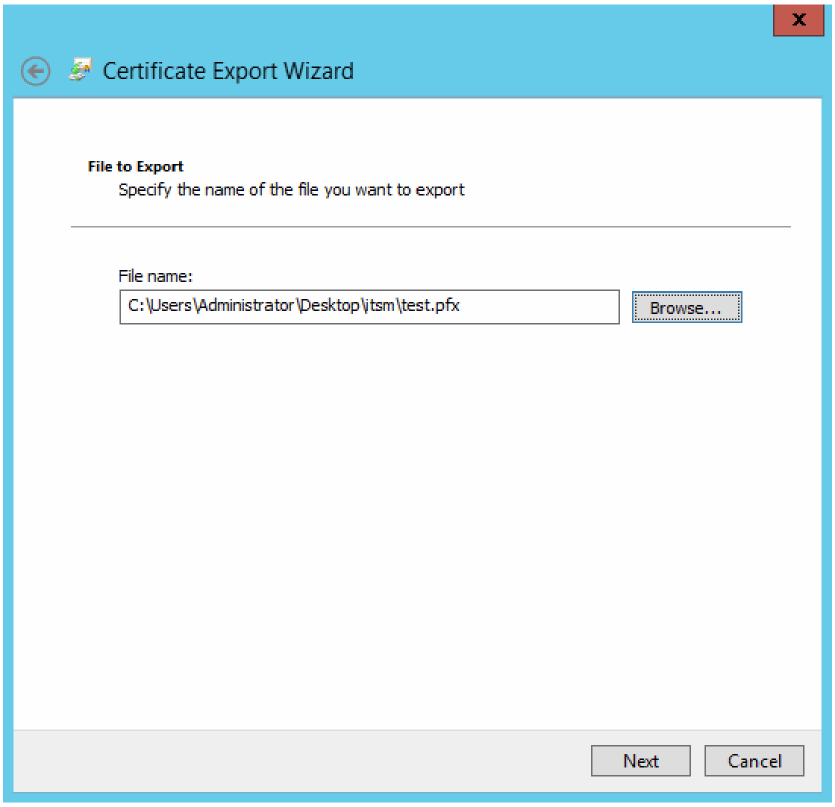
To maintain security, you must protect the private key to a security principal or by using a password.

☐ Group or user names (recommended)

☑ Password: ••••

Confirm password: ••••

- Specify a name, select place to save it > click 'Next'



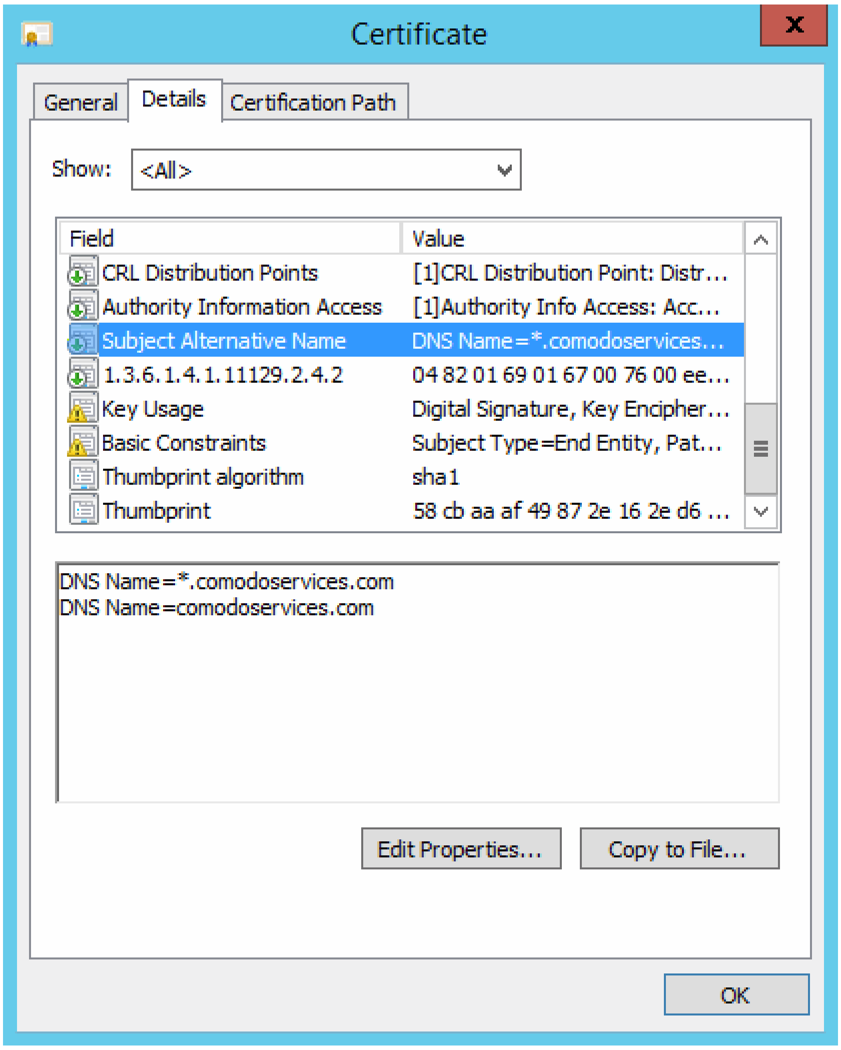
Certificate Export Wizard

File to Export

Specify the name of the file you want to export

File name: C:\Users\Administrator\Desktop\itsm\test.pfx

- The last step, click 'Finish' to export the certificate to a .pfx file



Show: <All>

Field	Value
CRL Distribution Points	[1]CRL Distribution Point: Distr...
Authority Information Access	[1]Authority Info Access: Acc...
Subject Alternative Name	DNS Name=*.comodoservices...
1.3.6.1.4.1.11129.2.4.2	04 82 01 69 01 67 00 76 00 ee...
Key Usage	Digital Signature, Key Encipher...
Basic Constraints	Subject Type=End Entity, Pat...
Thumbprint algorithm	sha1
Thumbprint	58 cb aa af 49 87 2e 16 2e d6 ...

DNS Name=*.comodoservices.com
DNS Name=comodoservices.com

Options to prepare Endpoint Manager (ITSM) server and Tigase certificates using openssl

It has 2 options to extract certificates to use them in the Endpoint Manager (ITSM) server and Tigase server:

1. **Option 1: Use the script scriptpfx.sh to create cert.crt, cert.key and in this case comodoservices.com.pem**
2. **Option 2: Manually create cert.crt, cert.key and comodoservices.com.pem from a .pfx file**

Option 1: Use the script scriptpfx.sh to create cert.crt, cert.key and in this case comodoservices.com.pem

- Create a folder on the itsm or tigase server using for example FileZilla
- Copy the .pfx file to the folder
- Create file `scriptpfx.sh` in the created folder (near .pfx file) with following content:

```
#!/bin/bash
openssl pkcs12 -in $1 -nocerts -nodes -passin pass:$2 | sed -ne '/-BEGIN PRIVATE
KEY-/,/-END PRIVATE KEY-/p' > clientcert.key
openssl pkcs12 -in $1 -clcerts -nokeys -passin pass:$2 | sed -ne '/-BEGIN
CERTIFICATE-/,/-END CERTIFICATE-/p' > clientcert.crt
openssl pkcs12 -in $1 -cacerts -nokeys -chain -passin pass:$2 | sed -ne '/-BEGIN
CERTIFICATE-/,/-END CERTIFICATE-/p' > cacerts.crt
a="$(openssl crl2pkcs7 -nocrl -certfile cacerts.crt | openssl pkcs7 -print_certs -text -
noout | sed -n 's/^.*CN=//p' | sed -n 1p)"
b="$(openssl crl2pkcs7 -nocrl -certfile cacerts.crt | openssl pkcs7 -print_certs -text -
noout | sed -n 's/^.*CN=//p' | sed -n 2p)"
```

```
if [ "$a" == "$b" ]; then
        cabundle="$(cat cacerts.crt | wc -l)"
        if [ "$cabundle" -gt 1 ]; then
                cat cacerts.crt | sed -ne '/-BEGIN CERTIFICATE-/,/-END 
CERTIFICATE-/p; /-END CERTIFICATE-/q' > rootca.crt
                cat cacerts.crt > intermediatefile.crt
                nr="$(cat rootca.crt | wc -l)"
                sed -i 1,"${nr}"d intermediatefile.crt
                cat rootca.crt > newcertificatechain.crt
                cabundle1="$(cat intermediatefile.crt | wc -l)"
                if [ "$cabundle1" -gt 1 ]; then
                        cat intermediatefile.crt | sed -ne '/-BEGIN CERTIFICATE-/,/-END 
CERTIFICATE-/p; /-END CERTIFICATE-/q' > intermediate1.crt
                        cat intermediate1.crt >> newcertificatechain.crt
                        nr1="$(cat intermediate1.crt | wc -l)"
                        sed -i 1,"${nr1}"d intermediatefile.crt
                        cabundle2="$(cat intermediatefile.crt | wc -l)"
                        if [ "$cabundle1" -gt 1 ]; then
                                cat intermediatefile.crt | sed -ne '/-BEGIN 
CERTIFICATE-/,/-END CERTIFICATE-/p; /-END CERTIFICATE-/q' > intermediate2.crt
                                cat intermediate2.crt intermediate1.crt rootca.crt > 
newcertificatechain.crt
                                rm intermediate2.crt
                                rm intermediate1.crt
                                rm rootca.crt
                                rm intermediatefile.crt
                        else
                                cat intermediate1.crt rootca.crt > 
newcertificatechain.crt
                                rm intermediate1.crt
                                rm rootca.crt
                                rm intermediatefile.crt
                        fi
                else
                        cat rootca.crt > newcertificatechain.crt
                        rm rootca.crt
                        rm intermediatefile.crt
                fi
                cat clientcert.key > cert.key
                cat clientcert.crt newcertificatechain.crt > cert.crt
                cat clientcert.key clientcert.crt newcertificatechain.crt > $3
                rm clientcert.key
                rm clientcert.crt
                rm cacerts.crt
                rm newcertificatechain.crt
        else
                echo The certificate chain is not included in the $1.
                echo Please create again the $1 and include the certificate chain.
        fi
else
        cat clientcert.key > cert.key
        cat clientcert.crt newcertificatechain.crt > cert.crt
        cat clientcert.key clientcert.crt cacerts.crt > $3
        rm clientcert.key
        rm clientcert.crt
        rm cacerts.crt
fi
```

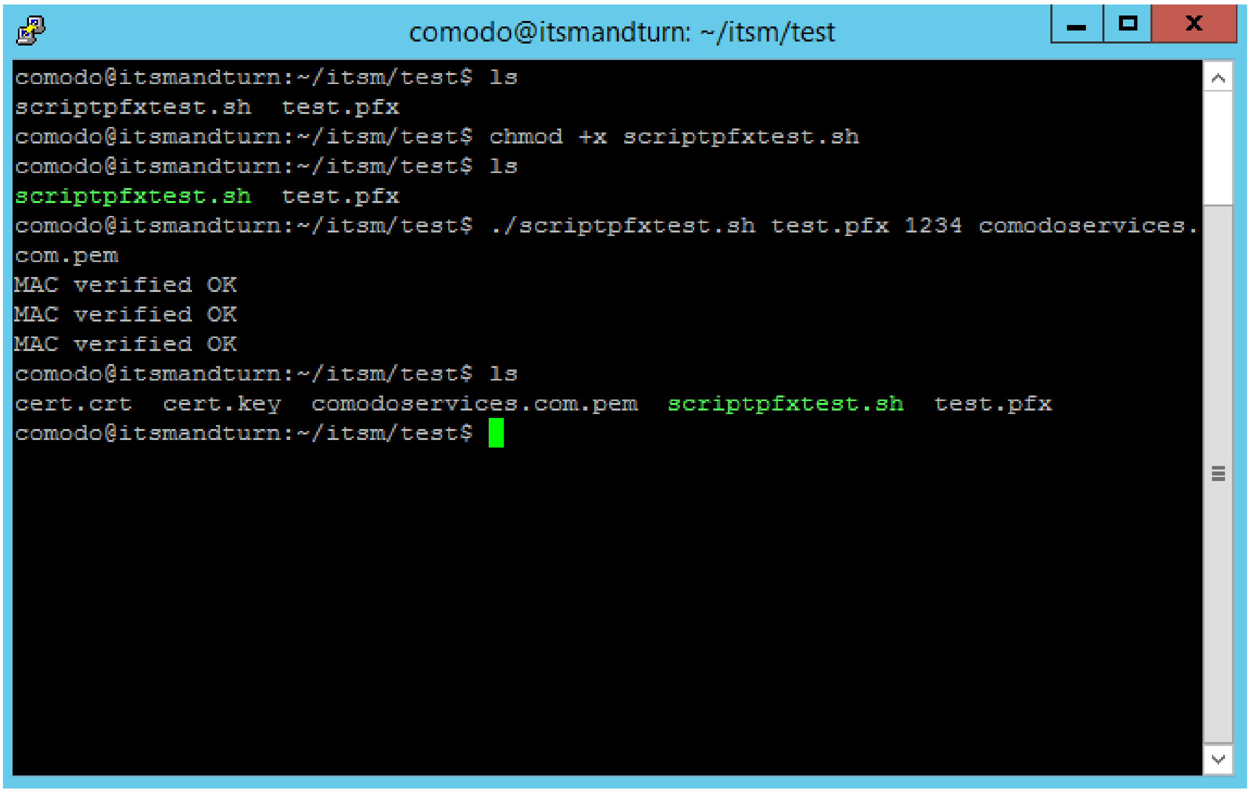
1. Run `chmod +x scriptpfxtest.sh` to make the file executable
2. Run `./scriptpfxtest.sh test.pfx 1234 comodoservices.com.pem` to generate cert.crt, cert.key and comodoservices.com.pem

   Format to use the command:

   ```
   ./scriptpfxtest.sh Parameter1 Parameter2 Parameter3
   ```

   Where:

- Parameter1: test.pfx – is the name of the .pfx file
- Parameter2: 1234 – is the password for the .pfx file
- Parameter3: comodoservices.com.pem – is the FQDN.pem that you want to use



```
comodo@itsmandturn:~/itsm/test$ ls
scriptpfxtest.sh  test.pfx
comodo@itsmandturn:~/itsm/test$ chmod +x scriptpfxtest.sh
comodo@itsmandturn:~/itsm/test$ ls
scriptpfxtest.sh  test.pfx
comodo@itsmandturn:~/itsm/test$ ./scriptpfxtest.sh test.pfx 1234 comodoservices.
com.pem
MAC verified OK
MAC verified OK
MAC verified OK
comodo@itsmandturn:~/itsm/test$ ls
cert.crt  cert.key  comodoservices.com.pem  scriptpfxtest.sh  test.pfx
comodo@itsmandturn:~/itsm/test$
```

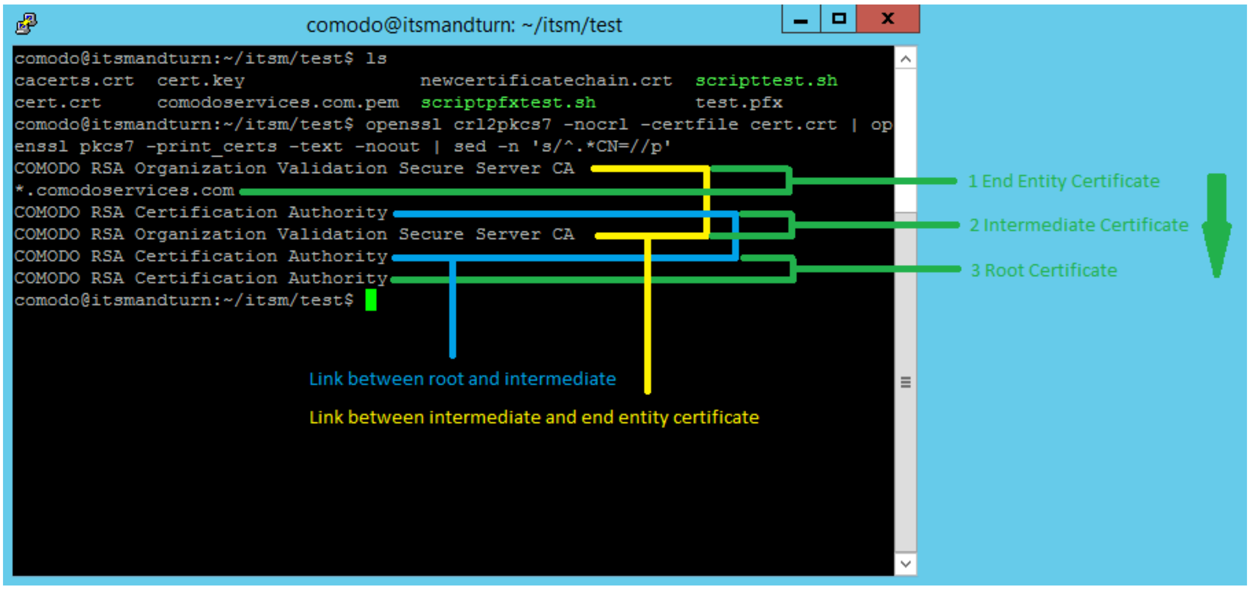
The script execution will create:

- cert.key
- cert.crt
- comodoservices.com.pem

To confirm that files are generated correctly, use the following commands:

```
openssl crl2pkcs7 -nocrl -certfile cert.crt | openssl pkcs7 -print_certs -text -noout |
sed -n 's/^.*CN=//p'
```

This confirms whether the certificate chain is in the correct order from top to bottom:



```
comodo@itsmandturn:~/itsm/test$ ls
cacerts.crt  cert.key                newcertificatechain.crt  scripttest.sh
cert.crt     comodoservices.com.pem  scriptpfxtest.sh         test.pfx
comodo@itsmandturn:~/itsm/test$ openssl crl2pkcs7 -nocrl -certfile cert.crt | op
enssl pkcs7 -print_certs -text -noout | sed -n 's/^.*CN=//p'
COMODO RSA Organization Validation Secure Server CA
*.comodoservices.com
COMODO RSA Certification Authority
COMODO RSA Organization Validation Secure Server CA
COMODO RSA Certification Authority
COMODO RSA Certification Authority
comodo@itsmandturn:~/itsm/test$
```

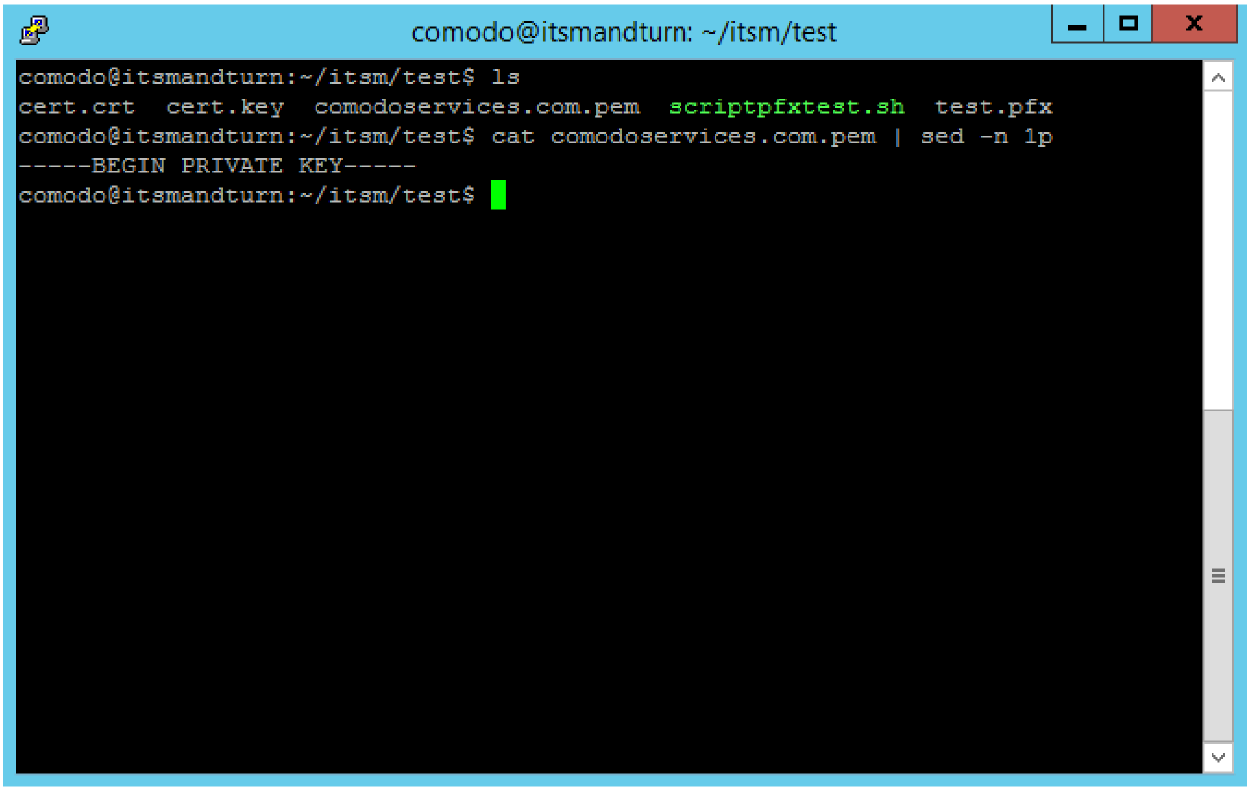
1 End Entity Certificate
2 Intermediate Certificate
3 Root Certificate
Link between root and intermediate
Link between intermediate and end entity certificate

You can use the same command on the Tigase certificate. Example: comodoservices.com.pem.

The difference between Endpoint Manager (ITSM) and Tigase certificates is that the Tigase certificate has a private key on top.

To verify this, we can use the following command:

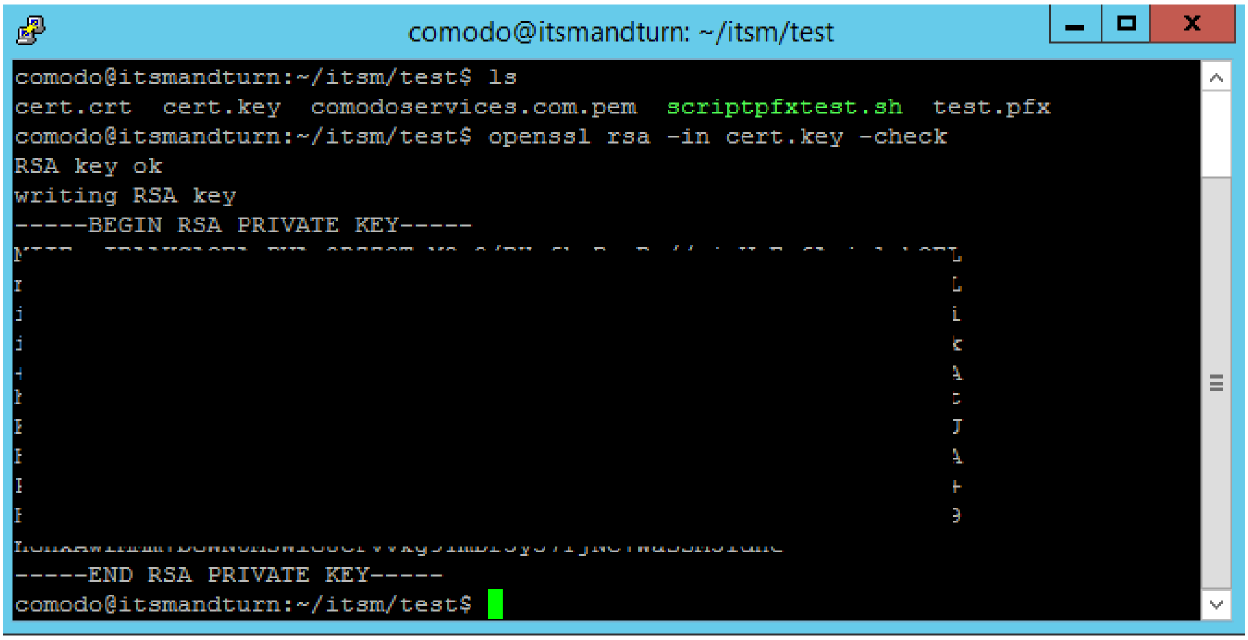
```
cat comodoservices.com.pem | sed -n 1p
```



```
comodo@itsmandturn:~/itsm/test$ ls
cert.crt  cert.key  comodoservices.com.pem  scriptpfxtest.sh  test.pfx
comodo@itsmandturn:~/itsm/test$ cat comodoservices.com.pem | sed -n 1p
-----BEGIN PRIVATE KEY-----
comodo@itsmandturn:~/itsm/test$
```

- Verify the private key using the command:

```
openssl rsa -in cert.key -check
```

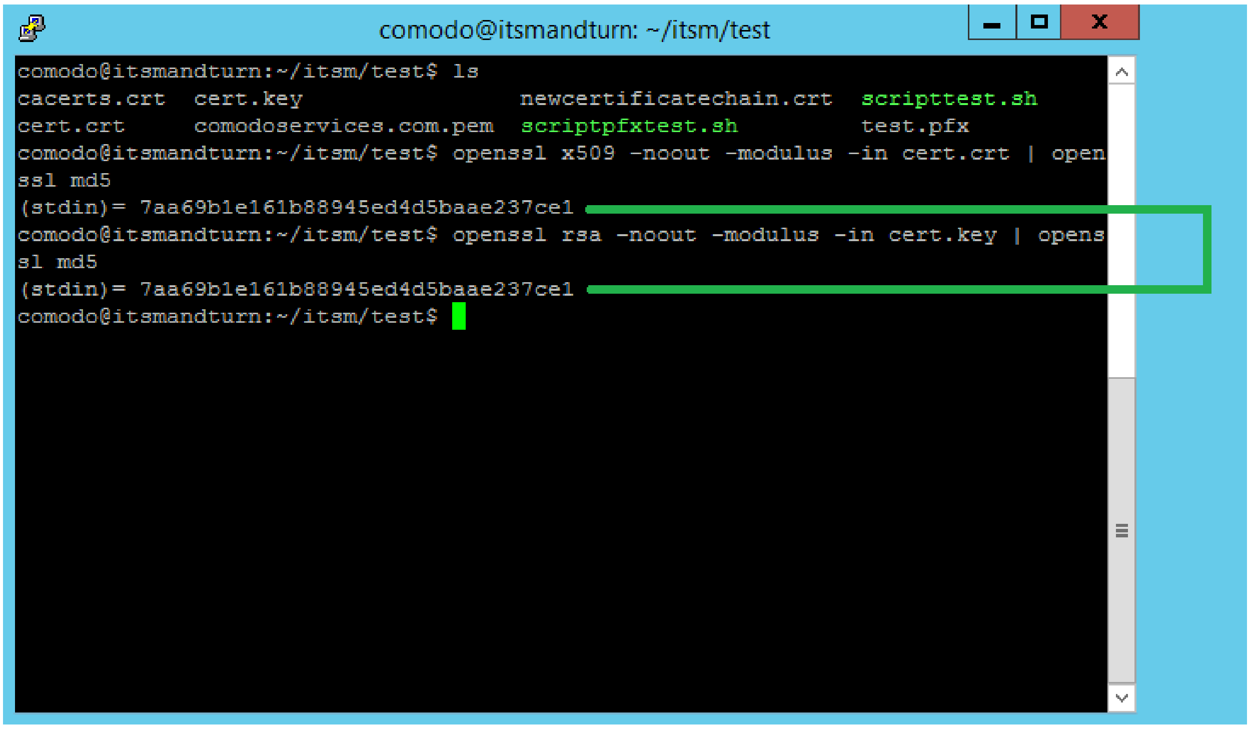


```
comodo@itsmandturn:~/itsm/test$ ls
cert.crt  cert.key  comodoservices.com.pem  scriptpfxtest.sh  test.pfx
comodo@itsmandturn:~/itsm/test$ openssl rsa -in cert.key -check
RSA key ok
writing RSA key
-----BEGIN RSA PRIVATE KEY-----
[redacted]
-----END RSA PRIVATE KEY-----
comodo@itsmandturn:~/itsm/test$
```

Use the following command to confirm that the private key is associated with the certificate :

```
openssl x509 -noout -modulus -in cert.crt | openssl md5
openssl rsa -noout -modulus -in cert.key | openssl md5
```

You will receive the same number on confirmation that private key is associated with the certificate.



```
comodo@itsmandturn:~/itsm/test$ ls
cacerts.crt  cert.key                 newcertificatechain.crt  scripttest.sh
cert.crt     comodoservices.com.pem  scriptpfxtest.sh         test.pfx
comodo@itsmandturn:~/itsm/test$ openssl x509 -noout -modulus -in cert.crt | open
ssl md5
(stdin)= 7aa69b1e161b88945ed4d5baae237ce1
comodo@itsmandturn:~/itsm/test$ openssl rsa -noout -modulus -in cert.key | opens
sl md5
(stdin)= 7aa69b1e161b88945ed4d5baae237ce1
comodo@itsmandturn:~/itsm/test$
```

Option 2: Manually create cert.crt, cert.key and comodoservices.com.pem from a .pfx file

- Copy the .pfx file on a folder on Endpoint Manager (ITSM) or Tigase server using for example FileZilla

Run this command to create cert.key:

```
openssl pkcs12 -in test.pfx -nocerts -nodes | sed -ne '/-BEGIN PRIVATE KEY-/,/-END
PRIVATE KEY-/p' > cert.key
```

To extract only the certificate, run the command:

```
openssl pkcs12 -in test.pfx -clcerts -nokeys | sed -ne '/-BEGIN CERTIFICATE-/,/-END
CERTIFICATE-/p' > clientcert.crt
```

- To extract the certificate chain, run the command:

```
openssl pkcs12 -in test.pfx -cacerts -nokeys -chain | sed -ne '/-BEGIN CERTIFICATE-/,/-
END CERTIFICATE-/p' > cacerts.crt
```

- To confirm the cacerts.crt has the correct order, run the command:

```
openssl crl2pkcs7 -nocrl -certfile cacerts.crt | openssl pkcs7 -print_certs -text -noout
| sed -n 's/^.*CN=//p'
```

If the certificate chain is as order, bottom to top instead of top to bottom, you can use script to reverse the order.

- Create file scriptorder.sh (in the folder near .pfx file) with following content:

```
#!/bin/bash
cabundle="$(cat $1 | wc -l)"
if [ "$cabundle" -gt 1 ]; then
        cat $1 | sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p; /-END
CERTIFICATE-/q' > rootca.crt
        cat $1 > intermediatefile.crt
        nr="$(cat rootca.crt | wc -l)"
        sed -i 1,"${nr}"d intermediatefile.crt
        cat rootca.crt > newcertificatechain.crt
        cabundle1="$(cat intermediatefile.crt | wc -l)"
        if [ "$cabundle1" -gt 1 ]; then
                cat intermediatefile.crt | sed -ne '/-BEGIN CERTIFICATE-/,/-END
CERTIFICATE-/p; /-END CERTIFICATE-/q' > intermediate1.crt
                cat intermediate1.crt >> newcertificatechain.crt
                nr1="$(cat intermediate1.crt | wc -l)"
                sed -i 1,"${nr1}"d intermediatefile.crt
                cabundle2="$(cat intermediatefile.crt | wc -l)"
                if [ "$cabundle1" -gt 1 ]; then
                        cat intermediatefile.crt | sed -ne '/-BEGIN CERTIFICATE-/,/-END
CERTIFICATE-/p; /-END CERTIFICATE-/q' > intermediate2.crt
                        cat intermediate2.crt intermediate1.crt rootca.crt >
newcertificatechain.crt
                        rm intermediate2.crt
                        rm intermediate1.crt
                        rm rootca.crt
                        rm intermediatefile.crt
                else
                        cat intermediate1.crt rootca.crt > newcertificatechain.crt
                        rm intermediate1.crt
                        rm rootca.crt
                        rm intermediatefile.crt
                fi
        else
                cat rootca.crt > newcertificatechain.crt
                rm rootca.crt
                rm intermediatefile.crt
        fi
else
        echo The file is empty.
        echo Lines = $cabundle
fi
```

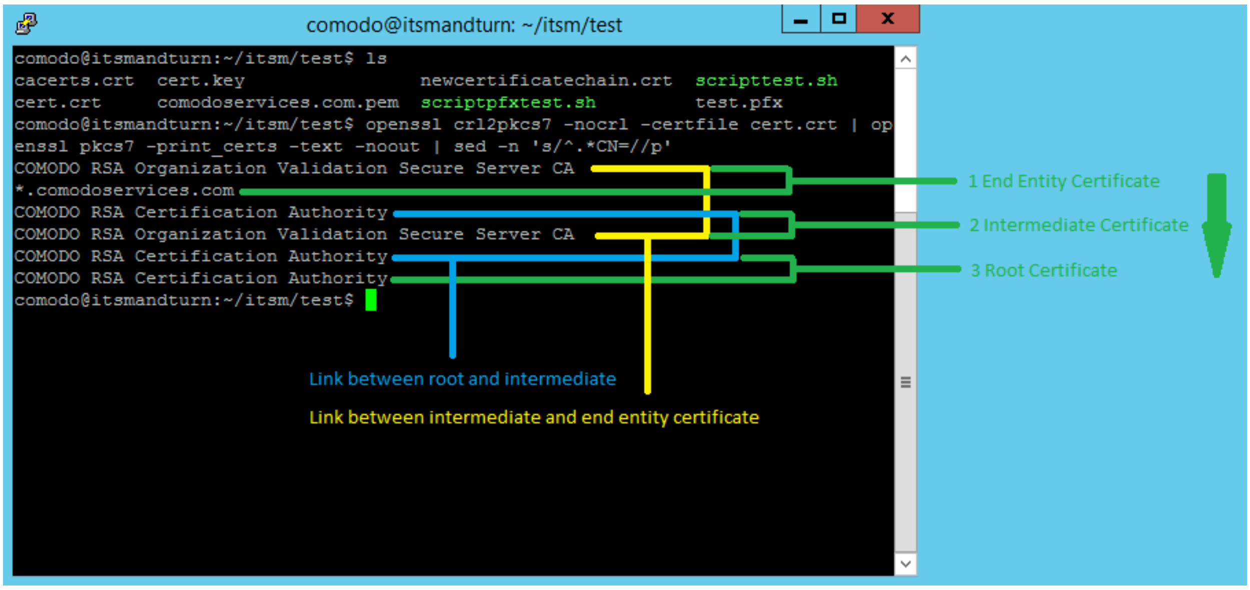
1. Run the script. To make the script executable use the command:

```
chmod +x scriptorder.sh
```

2. Once the script is executable, run it by providing the cacerts.crt as parameter:

```
./scriptorder.sh cacerts.crt
```

See example below:


comodo@itsmandturn: ~/itsm/test
1 End Entity Certificate
2 Intermediate Certificate
3 Root Certificate
Link between root and intermediate
Link between intermediate and end entity certificate

```
comodo@itsmandturn:~/itsm/test$ ls
cacerts.crt  cert.key                newcertificatechain.crt  scripttest.sh
cert.crt     comodoservices.com.pem  scriptpfxtest.sh         test.pfx
comodo@itsmandturn:~/itsm/test$ openssl crl2pkcs7 -nocrl -certfile cert.crt | op
enssl pkcs7 -print_certs -text -noout | sed -n 's/^.*CN=//p'
COMODO RSA Organization Validation Secure Server CA
*.comodoservices.com
COMODO RSA Certification Authority
COMODO RSA Organization Validation Secure Server CA
COMODO RSA Certification Authority
COMODO RSA Certification Authority
comodo@itsmandturn:~/itsm/test$
```

If you don't want to use the script, use the following commands to extract certificates in order:

- This command will extract the first certificate from the file, in our case the root:

```
cat cacerts.crt | sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p; /-END
CERTIFICATE-/q' > rootca.crt
```

- The second command deletes the certificate from the cacerts.crt to be able to extract the second available certificate

```
nr="$(cat rootca.crt | wc -l)";sed -i 1,"${nr}"d cacerts.crt
```

Use these three commands to extract your certificate chain and use cat to concatenate them in the correct order.

- Example to create comodoservices.com.pem:

```
cat cert.key clientcert.crt intermediate.crt root.crt > comodoservices.com.pem
```

- Example to create cert.crt:

```
cat clientcert.crt intermediate.crt root.crt > cert.crt
```

Use options from **Option 1** to verify them and confirm that the certificate files have been created/extracted correctly.

10. Installation via Installer

Installation

Note.

- To pull docker images you'll have to enter credentials (login, password) from your Docker Hub account (Link: **https://hub.docker.com/**).
- Be sure your account has access for pulling on-premise images.
- If you don't have Docker Hub account, create it and ask support for access.

Download installer and run it from root user:

```
curl -L -O http://get.on-premise.itarian.com/installer && chmod +x installer && sudo
./installer
```

Note. If you run not from root and started setup, please stop installation. Than run it from root user and start setup from very beginning.

Configuration files will be stored to folder `/home/[SUDO_USER]/itsm` (if it's possible to get SUDO_USER) or `/root/itsm`

Look to console output. There will be information for access to installation in browser:

- username (admin)
- password (always new, you don't need to save it)
- port

Open in browser: `http://{your_ip}:{port}/`

- Press the button "To start setup". You'll have to enter credentials from previous step. Than follow the instructions (Enter all necessary fields)
- The last step of installation (working with docker-compose) may take some time. After on-premise has been installed, you'll see the message
- Than you may stop running installer

- If something went wrong during installation or you see some errors, look to console output for more details

Updating

- On-premise installer also gives a possibility to update on-premise application if installation was executed by installer.
- In this case you'll see the "To update docker images" button.
- Press it.

11. Manual Installation

1. Prerequisites: install docker and docker-compose

- Login to remote server

```
ssh username@ip-or-hostname
```

- Get installation script (for ubuntu)

```
wget http://get-compose.on-premise.itarian.com/install-docker-compose.sh
```

- Make file executable

```
chmod +x install-docker-compose.sh
```

- Run script

```
sudo ./install-docker-compose.sh
```

- Setup local user permissions

```
sudo usermod -a -G docker $USER
```

- Logout from current session and login again to apply local user group changes

```
exit
ssh username@ip-or-hostname
```

- Perform docker login:

```
docker login
```

Note: Your docker account must be created on **hub.docker.com** and added by ITarian team to allow for on-premise storage.

2. Extra server setup

Only for itsm server it need to tune system settings:

```
sudo sysctl -w vm.max_map_count=262144
echo vm.max_map_count=262144 | sudo tee -a /etc/sysctl.conf
```

3. Get docker-compose.yml and configure settings:

    1. Create and navigate to itsm dir

    ```
    mkdir ~/itsm
    cd ~/itsm
    ```

    2. Get docker-compose.yml for specific server:

For Endpoint Manager (ITSM) server

```
wget https://get-compose.on-premise.itarian.com/docker-compose-with-
turn.yml -O docker-compose.yml
```

For Tigase server

```
wget https://get-compose.on-premise.itarian.com/tigase-docker-compose.yml -
O docker-compose.yml
```

3. Create file with name **.env** and fill it according to your server requirements:

For Endpoint Manager (ITSM) server

```
ITSM_DOMAIN=on-premise.itsm.local
ITSM_TURN_SERVERS=ip of turn server
ITSM_XMPP_HOST=xmpp.itsm.local
ITSM_XMPP_IP=ip of xmpp server
ITSM_WEB_HOST=same as ITSM_DOMAIN on-premise.itsm.local
ITSM_WEB_IP=ip of this host
```

Where:

```
ITSM_DOMAIN - domain name which must be the same as the certificate domain
used in setup
ITSM_TURN_SERVERS - list of ips where turn server is running separated by
comma or space ( if turn servers has been setup )
ITSM_XMPP_HOST - domain for tigase server ( if tigase has been setup )
ITSM_XMPP_IP - ip for host specified in ITSM_XMPP_HOST if dns record cannot
be resolved ( if tigase has been setup without dns )
ITSM_WEB_HOST - domain name which used by rmm microservices and points to
itsm-server.
ITSM_WEB_IP - ip for host specified in ITSM_WEB_HOST if dns record cannot
be resolved
```

For Tigase (xmpp) server

```
TIGASE_DOMAIN=itsm.local
ITSM_WEB_HOST=on-premise.itsm.local
ITSM_WEB_IP=ip of itsm server
```

4. Prepare and save SSL certificates

1. Prepare ASCII encoded certificate files:

- cert.crt - wildcard certificate for domain which specified as ITSM_DOMAIN
- cert.key - private key without password for cert.crt
- chain.crt - chaining certificate for cert.crt
- root.crt - CA root certificate for chain

2. Compose and save certificate files

For Endpoint Manager (ITSM) server

Save cert.crt and cert.key.

```
sudo mkdir -p /opt/itsm/web/certs
cat cert.crt chain.crt | sudo tee /opt/itsm/web/certs/cert.crt
sudo cp cert.key /opt/itsm/web/certs/cert.key
```

For tigase server

Save single certificate bundle under the name specified in TIGASE_DOMAIN and **.pem** extension.

```
sudo mkdir -p /opt/tigase/certs
cat cert.key cert.crt chain.crt root.crt | sudo tee
/opt/tigase/certs/itsm.local.pem
```

5. Starting up and basic management:

- Starting system

```
cd ~/itsm
docker-compose up -d
```

- Stopping system

```
docker-compose down
```

- View logs / debugging

```
docker-compose logs
```

12. Manual Upgrade

To perform upgrade it need to go on next steps:

1. Go to itsm directory which contains **docker-compose.yml** and **.env** files

```
cd ~/itsm
```

2. Stop the system

```
docker-compose stop
```

3. Get latest **docker-compose.yml file for regarding to your server type**

For Endpoint Manager (ITSM) server

```
wget https://get-compose.on-premise.itarian.com/docker-compose-with-turn.yml -O
docker-compose.yml
```

For tigase server

```
wget https://get-compose.on-premise.itarian.com/tigase-docker-compose.yml -O
docker-compose.yml
```

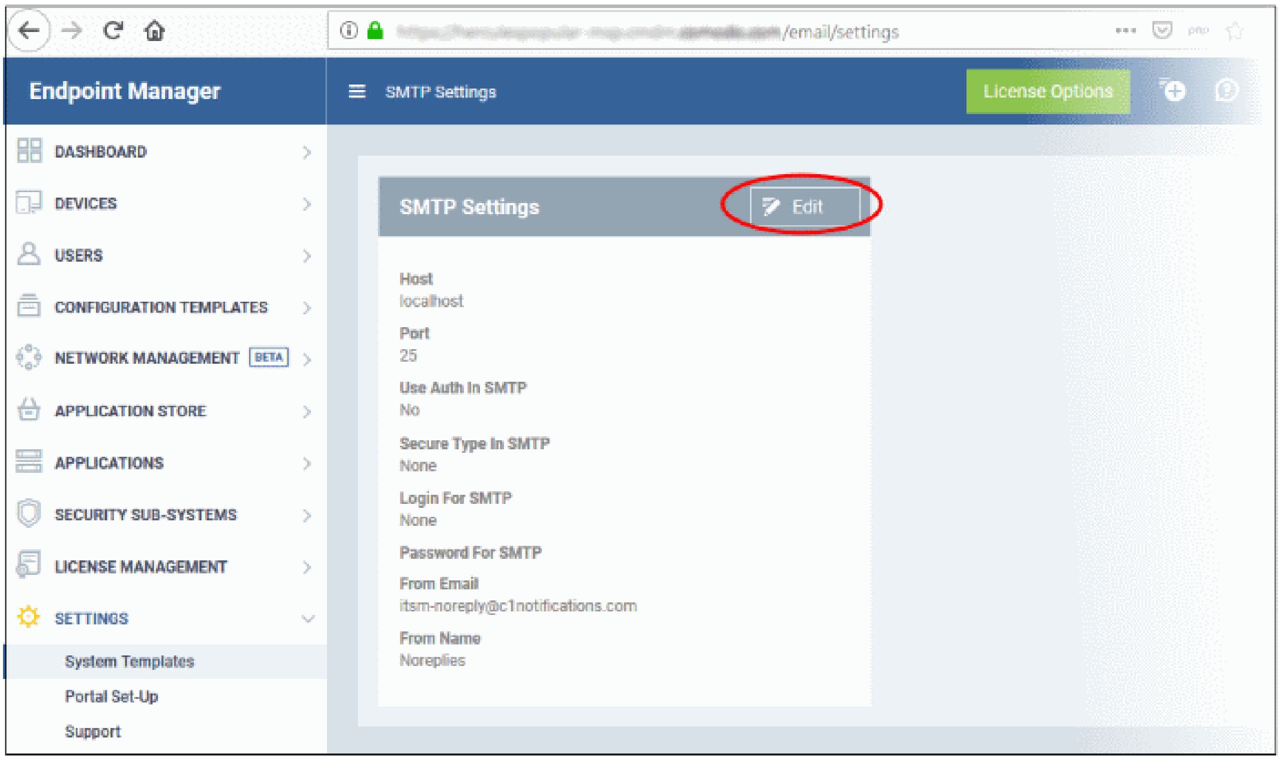
4. Run the system again

```
docker-compose up -d
```

13. SMTP Settings

After completing the installation you need to setup SMTP to have ability to receive email from Endpoint Manager (ITSM) server.

1. Login to to ITarian as admin
2. Click 'Applications' > 'Endpoint Manager' > 'Settings' > 'System Templates'
3. Open link of your ITSM server domain specified during installation (for example https://onpremise.itsm.mycompany.com/email/settings
4. Click the 'Edit' button.



Host
localhost

Port
25

Use Auth In SMTP
No

Secure Type In SMTP
None

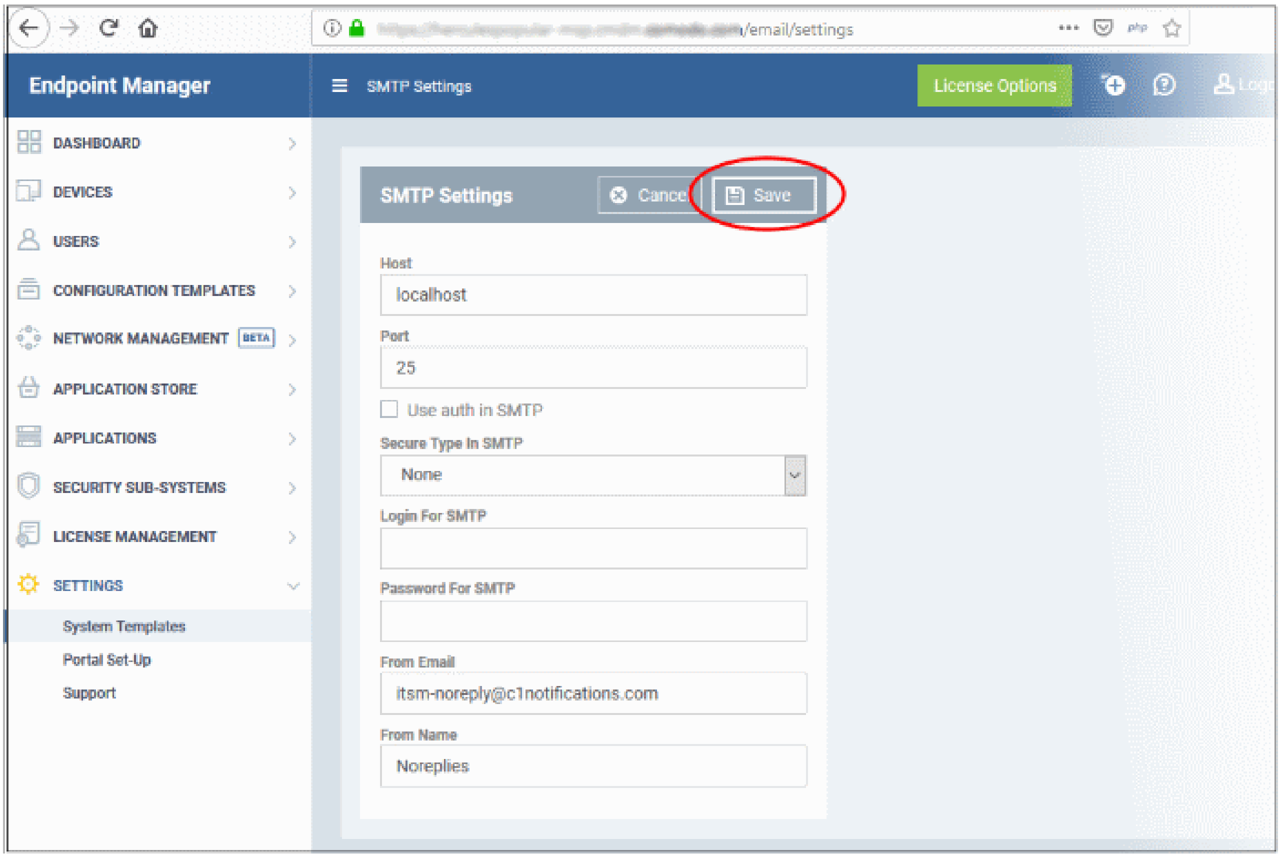
Login For SMTP
None

Password For SMTP

From Email
itsm-noreply@c1notifications.com

From Name
Noreplies

5. Fill all fields according to your corporate smtp settings.



Host
localhost

Port
25

□ Use auth in SMTP

Secure Type In SMTP
None

Login For SMTP

Password For SMTP

From Email
itsm-noreply@c1notifications.com

From Name
Noreplies

6. Click 'Save'.