Deploying 1E Client on Solaris

On this page:

Summary

Guidance for deploying 1E Client onto **Solaris** devices, including installation and uninstallation. Only the Tachyon features of 1E Client are available on Solaris and other non-Windows OS.

Requirements

Please review Design Considerations and Requirements pages. After installation please review the Verifying page.

For details of supported OS platforms please refer to Supported Platforms reference page.

Guidance provided below is for installation on Solaris.

Please contact 1E if guidance is required for installation on other non-Windows OS and for Android.

1E Client does not have a license key. Even so, you must adhere to the terms of your license agreement.

Deployment choices

You must decide how you will configure the 1E Client and deploy to devices. For more information about configuring the 1E Client properties during and after installation, please refer to 1E Client configuration settings and installer properties.

Deploying the 1E Client is normally achieved using your existing software deployment tool.

Non-Windows installation account

To install the 1E Client on a non-Windows client the installation account must have privileges to run the **sudo** command.

Certificate files

Each Tachyon client device requires a **.pfx** certificate file. Please refer to Client certificates below, for steps to create and use the **.pfx** file.

For Solaris devices, the Tachyon client does not use proprietary certificate stores. Instead, the client requires the certificate exists as a .PFX file in the client installation folder structure.

If you have configured Tachyon Server to require client certificates (Tachyon Setup: Client certificates) then each device requires a certificate with the following properties so the Tachyon client be authenticated by the Tachyon Switch.

1. Issued by a trusted Certificate Authority (CA)
   - The certificate for the Root CA in the Certification Path must exist in the Trusted Root CA store of the client
   - If the issuing CA is not the Root CA then the certificate for the issuing CA and any intermediate CA in the Certification Path must exist in the Intermediate CA store of the client
   - If either of these CA certificates are different to those used by the Tachyon Web Server, they will need to be exported and imported on the Tachyon Web Server
   - Most organizations have automated distribution of these CA certificates to clients and servers, using Group Policy for example.
2. Has at least the following Enhanced Key Usage
   - Client Authentication
3. Has at least the following Key Usage
   - Digital Signature
   - Key encipherment

4. Has a private key
    - For workgroup and non-Windows devices, the **private key must be exportable**
5. Revocation information is included.
    - References at least one CRL Distribution point that uses HTTP.
6. Has a Subject Name of type Common Name (**CN=<computername>**) or Subject Alternative Name (**DNS Name=<computername>**) where <computername> depends on the type of device:
    - On domain-joined Windows PCs this must be the **computername FQDN** of the computer, for example **W701.ACME.LOCAL**
    - On workgroup Windows PCs and non-Windows devices, this must be the computername of the computer - as returned by the **hostname** command, for example on Windows PC this could be **W701**, and on a Mac this could be **MAC01.local**

> Tachyon clients and Switches use OpenSSL and its validation process to verify certificates.

Preparation

The Windows and non-Windows versions of the 1E Client are available for download from the 1E Support Portal.

Installation source files for 1E Client for non-Windows are available in a zip file called **1EClient-Non-Windows.v5.1.x.x.zip**

Within the zip, the Solaris 1E Client is provided as the following **.p5p** files:

- 1e.client-Solaris_11.3_SPARC_v5.1.x.x.p5p
- 1e.client-Solaris_11.4_x64_v5.1.x.x.p5p

Installation

> The following specific libraries are required, but are usually installed by default:
>
> - libcurl
> - zlib

The Solaris 1E Client is provided as an Image Packaging System (IPS) package archive file, with the suffix **.p5p**. The basic command to install a package archive is:

```
pkg install -g package_archive_file package_name
```

Unfortunately, IPS does not support pre- and post-install actions of any sort, so after installation separate commands must be run to configure the client, enable it as a service, and start it. So, if the Tachyon Server (assuming that the switch and the background channel are both installed on the Tachyon Server) has the DNS Name FQDN **tachyon.acme.local** then the 3 commands needed are as follows:

```
sudo pkg install -g 1e.client-Solaris_11.3_x64_v4.0.0.573.p5p 1e.client
sudo /usr/sbin/1e.client.updateconf.sh /etc/1E/Client/1e.client.conf SWITCH=tachyon.acme.local:4000
BACKGROUNDCHANNELURL=https://tachyon.acme.local:443/Background/
sudo svcadm enable n1e-client
```

Please refer to 1E Client configuration settings and installer properties for a list of other configuration properties that can be configured in the same way.

> **Package repository errors**
>
> If you get an error telling you that you can't get to the Solaris package repository while installing, e.g. **Unable to contact any configured publishers** - which you'll probably get if your machine doesn't have internet access - you may need to temporarily disable the Solaris publisher from the package repository, by running the following command:
>
> ```
> sudo pkg set-publisher --disable solaris
> ```
>
> Then the pkg install should succeed. You should be able to re-enable the Solaris publisher (should you need to) by doing an `--enable` instead of `--disable` on the command line above.

Another restriction of Solaris IPS is that files and directories cannot be tagged so that they are not deleted on uninstall. If you want the client's persistent storage to survive after uninstall (e.g. because you are likely to install the client again in future), installing the additional **1e.client.persist** package will ensure this. This is included in the same package archive file as the client package itself. This command can be run either before or after installation of the client itself:

```
sudo pkg install -g 1e.client-Solaris_11.3_x64_v4.0.0.573.p5p 1e.client.persist
```

> **Warning: extra configuration for Solaris**
>
> Add a pair of entries for `https/443` (one each for `udp` and `tcp`) to the `/etc/services` file if they are not there already. Likewise `http/80` for consistency. The Tachyon background channel does not work without the `https` entries, and we get "`ERROR - failed to get host IP address for name '<background channel server>' because: service name not available for the specified socket type`", and file downloads for example fail.

Certificate Files for the Tachyon Solaris client

The client certificate **Tachyon.pfx** and a **cacert.pem** file are required in the hidden directory:*/etc/1E/Client/.sslcerts* (to create these files see Client certificates below).

If **Tachyon.pfx** contains the same certificate trust chain as the Tachyon Switch, then **cacert.pem** is optional. This is because the client will have already cached the public certificates when it parses **Tachyon.pfx** and **cacert.pem** will be duplicating the public certificate information. If **Tachyon.pfx** is using a different certificate trust chain from the Tachyon Switch, then **cacert.pem** is always required.

Reconfiguration

Please refer to 1E Client command-line parameters if you would like to see details of other CLI commands.

> This method is suitable for reconfiguring all 1E Client settings on non-Windows devices because only Tachyon client features are available and all settings are stored in the 1E Client configuration file.

Example:

```
/usr/sbin/1E.Client -reconfigure Switch=ACME-DMZ01.ACME.LOCAL:4000 BackgroundChannelUrl=https://ACME-DMZ01.
ACME.LOCAL:443/Background/ -restart
```

Client certificates

Each client device requires its own certificate, which must be created as a **.pfx** file.

Using OpenSSL to create the .pfx file

Each non-Windows devices requires its own certificate. Below is a guide for using a Microsoft CA to issue a certificate (which is the same for Windows computers), then exporting it and using OpenSSL to prepare it before installing it on the non-Windows device.

First, you will need to have created a new Certificate template on your Certificate Authority by making a duplicate of either the Computer or Workstation template and configuring it with at least the following properties:

- **General** - use a suitable name such as **Tachyon Devices** and validity period
- **Request Handling** - Allow private keys to be exported
- **Subject Name** - Allow information to be supplied in the certificate request, rather than being built from Active Directory information
- **Extensions** - Application Policies should contain only Client Authentication
- **Security** - ensure relevant users and computers will be able to request certificates.

Once the new template is created on the CA, issue it.

Using the issued template, request a certificate for a target device, and export it in **.pfx** form and remember the password.

The target device requires a copy of the basic **cacert.pem** and the **.pfx** file with its password removed. You can do this using the following steps. Use the relevant OpenSSL version for the OS. OpenSSL is normally available by default on Linux and Mac devices. If you want to follow these steps on Windows, you will need to download the open source version appropriate to your OS.

1. First, extract the certificate:

   ```
   openssl pkcs12 -clcerts -nokeys -in <YourPKCSFile>.pfx -out certificate.crt
   ```

2. Second, the CA key:

   ```
   openssl pkcs12 -cacerts -nokeys -in <YourPKCSFile>.pfx -out ca-cert.ca
   ```

3. Now, the private key:

   ```
   openssl pkcs12 -nocerts -in <YourPKCSFile>.pfx -out private.key -passout pass:TemporaryPassword
   ```

4. Remove the passphrase:

   ```
   openssl rsa -in private.key -out new.key -passin pass:TemporaryPassword
   ```

5. Put things together for the new PKCS-File (on Windows, **type** can be used instead of **cat**):

   ```
   cat new.key > PEM.pem
   cat certificate.crt >> PEM.pem
   cat ca-cert.ca >> PEM.pem
   ```

6. And create the new **.pfx** file, when prompted for a password ensure that you enter an empty password (that is press enter when prompted for the password and confirmation without entering any text):

   ```
   openssl pkcs12 -export -nodes -CAfile ca-cert.ca -in PEM.pem -out Tachyon.pfx
   ```

Now you have a new PKCS12 key file without passphrase on the private key part. This **Tachyon.pfx** file and the **cacert.pem** file, must be placed in one of the following locations - depending on the OS. These are hidden folders.

Storing the .pfx on the client

MultiExcerpt named '**SSLcertificateFileMethod**' was not found

The page: **Deploying 1E Client on macOS** was found, but the multiexcerpt named '**SSLcertificateFileMethod**' was not found. Please check/update the page name used in the 'multiexcerpt-include macro.

Uninstallation

The Solaris IPS packaging system does not support pre- and post-install actions, so the 1E Client must be disabled before uninstallation.

```
sudo svcadm disable 1e.client
sudo pkg uninstall 1e.client
```

If you protected the 1E Client's persistent storage, as described in Solaris installation, but no longer need it, then you will also need to run this command:

```
sudo pkg uninstall 1e.client.persist
```

Even if you did not protect the 1E Client's persistent storage, when the 1E Client is uninstalled the IPS system saves the directory at `/var/pkg/lost+found/etc/1E/1e.client/Persist-`*`timestamp`* , so it can be restored if necessary.