Alfa & Ariss
Network Security Solutions

# SecureW2 Client for Windows
# Administrator Guide

Version 2.2

Released: August 2004

# Table of Contents

# Preface

SecureW2 2.2.x now supports the use of a pre-configuration file. This file allows administrators to distribute a copy of a pre-configured SecureW2 that also automatically installs the correct certificates. Further more it is possible to query the user for their credentials during installation.

This guide shows how to make use of the pre-configuration script supported by SecureW2 2.2.x.

# Configuration

The SecureW2 pre-configuration file is based on a Microsoft INF File. There are different sections each depicting how SecureW2 is to be configured.

## Basic INF File

Each INF file must contain the following, if not SecureW2 will not be able to read the file:

```
[Version]
Signature  = "$Windows NT$"
Provider    = "Alfa & Ariss"
Config     = 3

Comments can be added in the INF file using a semicolon.
A semicolon is also used to disable lines. Attributes
that are not defined will be set to their Default value.

Example of a comment in the INF file:

; This is a comment

Example of a disabled line in the INF file:

; attribute1 = I am disabled
attribute2 = I am not disabled
```

# Global Configuration

Currently the global configuration only depicts which certificates must be installed.

## Certificate configuration

The Certificates section contains the certificate chain that is to be installed on the local computer. The Certificates section is defined using the following tag:

```
[Certificates]
```

In the Certificates section you must define your certificate chain. The following attributes define a certificate:

| Attribute | Value |
|---|---|
| `Certificate.n`<br>The value "n" should start at 0 and be incremented with each new certificate. | The location of the certificate.<br>The path is relative to where the installer is executed.<br>Currently only DER encoded X.509 certificates are supported.<br>**Value: String** |

"`Certificate.0`" must always refer to the TTLS Server certificate. The rest of the chain ("`Certificate.n`") should refer to certificates that are either Subordinate CA's or Root CA's.

The following example shows a certificate chain containing a TTLS certificate, a Subordinate CA certificate and the Root CA certificate:

```
[Certificates]
Certificate.0 = ttls.cer
Certificate.1 = subca.cer
Certificate.2 = rootca.cer
```

## Retrieving/Converting Certificates

To retrieve your CA and TTLS certificates and convert them to DER encoding you can use the following options:

1. On your radius server simply use the following openssl command to convert your CA and TTLS PEM certificates to DER encoding:

   openssl x509 -inform PEM -outform DER -in ttls.pem –out ttls.der

2. On a computer (Windows) running SecureW2 that already trusts your TTLS server the certificates have already been installed in the local certificate store of that computer. To retrieve the certificates you can use the "Microsoft Management Console" and the "Certificates" snap-in:

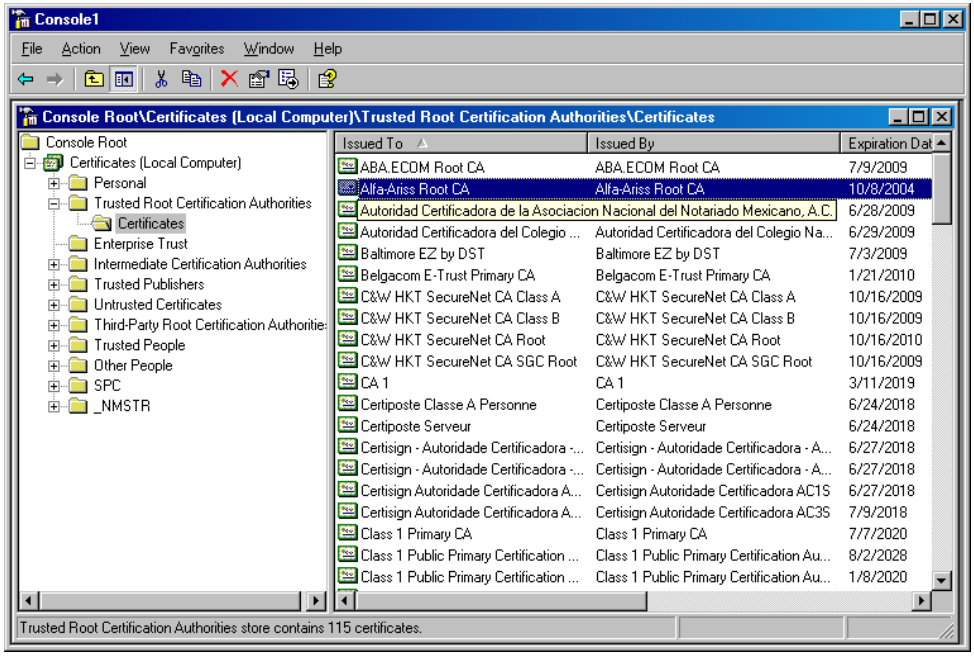


**Figure 1 MMC Certificate snap-in**

To use the MMC do the following:

a. Click on the "Start" menu
b. Click on the "Run" option
c. Enter the following command: mmc
   This will start-up a new Microsoft Management Console in which you can add snap-ins allowing you to control the different aspects of your computer.
d. Select "File" in the top menu
e. Select "Add/Remove snap-in"
   You are presented with the "Add/Remove snap-in" window in which you can select the snap-in's you wish to use.
f. In "Standalone" tab click on the "Add" button
   You are now presented with the "Add snap-in" window showing the different snap-in's that are available.
g. Select the "Certificates" snap-in and click on the "Add" button.
h. When asked which certificates are to be managed select "Computer account".
i. When asked for which computer the certificates are to be managed select "Local computer" and click on "Finish".
j. Click on "Close" to return to the "Add/Remove snap-in" window that now shows the "Certificates" snap-in.
k. Click on "Ok" to return to the main MMC window.
l. To find the CA certificate installed by SecureW2 expand the certificates snap-in so you can view the certificates in the "Trusted Root Certification Authorities".
m. Look for your CA certificates and right click on the certificate and select "All Tasks" and then "Export…".
n. You are now presented with the "Certificate Export Wizard". Run through the wizard and export the certificates using DER encoding to a location of your choosing.

## SSID Configuration

A profile has to be assigned to each SSID that is going to use SecureW2. A SSID section is defined using the following tag:

[SSID.n]

Where n is the number of the SSID section. This number must start at 1 and be incremented with each new SSID section.

Per SSID section you must define the following attributes:

| Attribute | Value |
|---|---|
| NAME | Name of the SSID.<br>**Value: String** |
| Profile | Name of the Profile that this SSID should use.<br>**Value: String** |

The following example shows 2 SSIDs and the profiles that they will use:

```
[SSID.1]
Name = MYSSID
Profile = "AENA"

[SSID.2]
Name = HOMESSID
Profile = "HOME"
```

## Profile Configuration

A SecureW2 profile contains the configuration used by SecureW2 when connecting to an SSID. A Profile section is defined using the following tag:

```
[Profile.n]
```

Where n is the number of the Profile section. This number must start at 1 and be incremented with each new Profile section.

Each profile must have the following attribute defined:

| Attribute | Value |
|---|---|
| NAME | Name of the Profile.<br>**Value: String** |

Attributes that are not defined in a Profile section will be set to their Default value. Per Profile section you can define the following attributes:

### Connection attributes

| Attribute | Value |
|---|---|
| `UseAlternateIdentity` | Enabling this option instructs SecureW2 to use an alternate outer identity.<br>**Value: TRUE or FALSE**<br>**Default Value: TRUE** |
| `UseAnonymousIdentity` | Enabling this option instructs SecureW2 to use an Anonymous outer identity.<br>**Value: TRUE or FALSE**<br>**Default Value: TRUE** |
| `AlternateOuterIdentity` | The value defined by this attribute is used as the outer identity. This option is only valid if `UseAnonymousIdentity` is FALSE.<br>**Value: String**<br>**Default Value: EMPTY** |
| `EnableSessionResumption` | Enabling this option instructs SecureW2 to use session resumption (quick connect).<br>**Value: TRUE or FALSE**<br>**Default Value: FALSE** |

### Certificate attributes

| Attribute | Value |
|---|---|
| `VerifyServerCertificate` | Enabling this option instructs SecureW2 verify the TTLS server certificate.<br>**Value: TRUE or FALSE**<br>**Default Value: TRUE** |
| `SpecifyRootCA` | Enabling this option instructs SecureW2 to only accept certificates issued by the CA defined in `TrustedRootAuthority`. If FALSE SecureW2 will use the default Windows Certificate trust.<br>**Value: TRUE or FALSE**<br>**Default Value: FALSE** |
| `TrustedRootAuthority` | The value defining by this attribute is the hexadecimal string of the SHA1 hash of the Trusted Root CA certificate. SecureW2 uses this to find the correct Root CA certificate installed on the local computer.<br><br>To retrieve the hexadecimal SHA1 value of a certificate in Windows, double-click on the certificate. In the Certificate window select the Details tab. The SHA1 value is listed as the Thumbprint.<br><br>Using openssl use the following command:<br>openssl sha1 < ttls.cer<br><br>The hexadecimal string should not contain spaces.<br>**Value: String**<br>**Default Value: EMPTY** |
| `VerifyServerName` | The value defining by this attribute is the string used to verify the common name in the certificate of the TTLS server.<br>**Value: String**<br>**Default Value: EMPTY** |

### Authentication attributes

| Attribute | Value |
|---|---|
| `AuthenticationMethod` | The value defining by this attribute is the authentication method used by SecureW2 to authenticate the user. Currently this can be two values:<br>PAP or EAP<br>**Value: String**<br>**Default Value: PAP** |
| `EAPType` | If EAP has been selected as the `AuthenticationMethod` the value defined by this attribute is the EAP-Type that is to be used. The following EAP methods are available:<br>▪ `4 = EAP-MD5`<br>▪ `26 = EAP-MSCHAP v2`<br>**Value: Numeric**<br>**Default value: 0** |

### User attributes

| Attribute | Value |
|---|---|
| `PromptUserForCredentials` | Enabling this option instructs SecureW2 to prompt the user for credentials during authentication.<br>**Value: TRUE or FALSE**<br>**Default Value: TRUE** |
| `UserName` | If the attribute `PromptUserForCredentials` is FALSE then setting this value to PROMPTUSER instructs the SecureW2 installer to prompt the user for credentials during installation.<br>**Value: PROMPTUSER**<br>**Default Value: EMPTY** |

### Advanced attributes

| Attribute | Value |
|---|---|
| `ServerCertificateOnLocalComputer` | Enabling this option instructs SecureW2 to verify if the TTLS certificate is installed on the local computer.<br>**Value: TRUE or FALSE**<br>**Default Value: FALSE** |
| `CheckForMicrosoftExtension` | Enabling this option instructs SecureW2 to verify if the TTLS certificate contains the correct Microsoft Extended Key Usage.<br>**Value: TRUE or FALSE**<br>**Default Value: FALSE** |
| `AllowNewConnections` | Enabling this option instructs SecureW2 to allow users to setup new connections.<br>**Value: TRUE or FALSE**<br>**Default Value: FALSE** |
| `RenewIPAddress` | Enabling this option instructs SecureW2 renew the DHCP IP Address of the authenticating adapter.<br>**Value: TRUE or FALSE**<br>**Default Value: FALSE** |

The following example shows all options that can be configured for a profile:

```
[Profile.1]
Name = "AENA"

;
; Connection
;
UseAlternateIdentity = FALSE
UseAnonymousIdentity = FALSE
AlternateOuterIdentity = username@domain.com
EnableSessionResumption = TRUE

;
; Certificates
;
VerifyServerCertificate = TRUE
SpecifyRootCA = TRUE
TrustedRootAuthority =
ea7b4e3ed43f095cc0763f3cd851d977e33be0f9
VerifyServerName = .domain.com

;
; Authentication
;
AuthenticationMethod = EAP
EAPType = 4; MD5-Challenge (EAP-MD5)

;
; User Account
;
PromptUserForCredentials = FALSE
UserName = PROMPTUSER


;
; Advanced
;
ServerCertificateOnLocalComputer = TRUE
CheckForMicrosoftExtension = TRUE
AllowNewConnections = TRUE
RenewIPAddress = TRUE
```

# Setup

Once you have setup your configuration file the SecureW2 installer can use this file to automate the installation process.

The first step is to rename your configuration file to SecureW2.INF.

The SecureW2 installer searches for a SecureW2.INF file during installation. If found it executes the script. If not found it will continue as usual. SecureW2 uses the current directory in which it was executed to search for the SecureW2.INF file.

## Building your own setup

For customers with a site license allowing them to distribute SecureW2 in their organization it is possible to package the SecureW2.inf, SecureW2_220.exe and certificates in an installation program that unpacks the file to a temporary directory and then runs the installer.

NOTE: it is important that the SecureW2 installation program is visible to the user due to the license agreement.

An example of such an installation program is NSIS, a free open source installer from NULLSoft. The following link is an NSIS installer example script for SecureW2 that allows you to create your own SecureW2 installer: http://www.securew2.com/uk/resources/securew2/v2/SecureW2_example.NSI