Administrators guide

Disclaimer

The software described in this document is furnished under a license agreement and may be used only in accordance with the terms of the agreement.

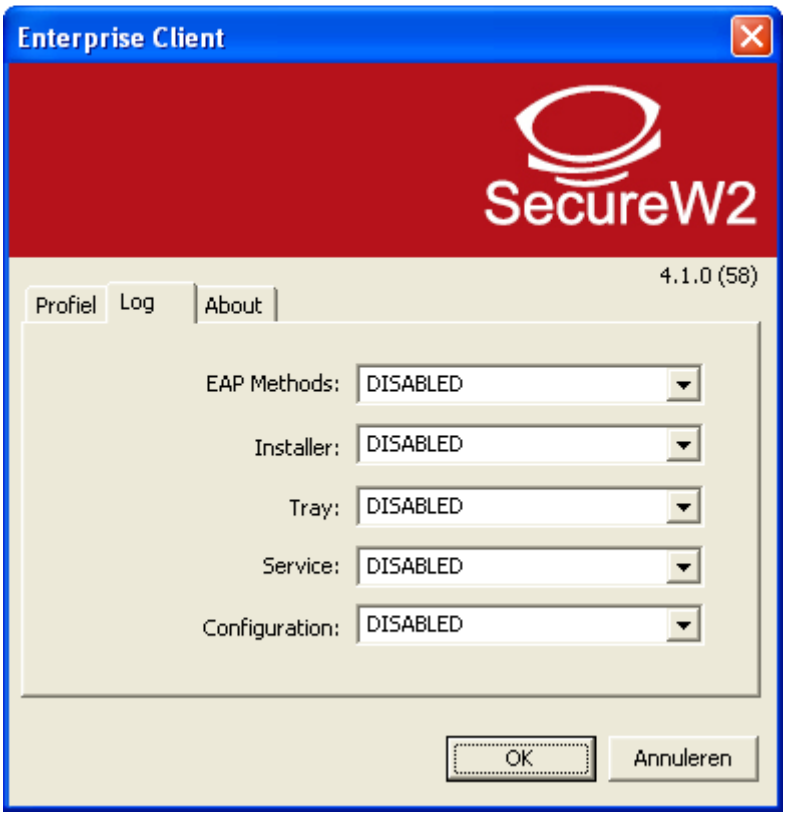
Released: July 2011

Every effort has been made to ensure the accuracy of this manual. However, SecureW2 makes no warranties with respect to this documentation and disclaims any implied warranties of merchantability and fitness for a particular purpose. SecureW2 shall not be liable for any errors or for incidental or consequential damages in connection with the furnishing, performance, or use of this manual or the examples herein. The information of this document is subject to change without notice.

Content

1. Log configuration

The SecureW2 Enterprise client offers an easy way of disabling and enabling the SecureW2 logging via by the configuration manager. The configuration manager can be invoked via the network properties of your adapter or by right clicking on the SecureW2 Tray icon and selecting “Configuration Manager”. Select the “Log” tab and you are presented with the available SecureW2 components and their current log level.



The trace files can be found in the following folder (see 1.1 Manual configuration):

%windir%\tracing (c:\WINDOWS\tracing).

1.1 Manual configuration

SecureW2 logging is controlled using the Microsoft Tracing functionality located in the Windows registry.

To access the log configuration open the following registry key:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\[EAP method]

For example: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\EAP-TTLS

The trace files can be found in the following folder:

%windir%\tracing (c:\WINDOWS\tracing).

The following registry entries control the logging:

<table>
<tr><th>Registry Entry</th><th colspan="4">Description</th></tr>
<tr><td>EnableFileTracing</td><td colspan="4">1 will enable the logging, 0 will disable the logging. The entry is a DWORD.</td></tr>
<tr><td>MaxFileSize</td><td colspan="4">This determines the size of the log file and when it should rotate the file. The entry is a DWORD. For example HEX 0x100000 would indicate a size of 1048576 bytes.</td></tr>
<tr><td>FileDirectory</td><td colspan="4">The directory for the log file. The name of the log file is the name of the EAP method. For example c:\windows\tracing\EAP-TTLS.LOG (EAP-TTLS.OLD for a rotated file).</td></tr>
<tr><td>FileTracingMask</td><td>SecureW2 offers 4 basic log levels. The entry is a DWORD.</td><td>Type</td><td>Flag (Hex)</td><td>Description</td></tr>
<tr><td></td><td></td><td>Error</td><td>0x10000</td><td>This level will log only error messages.</td></tr>
<tr><td></td><td></td><td>Warning</td><td>0x30000</td><td>This level will log additional warnings</td></tr>
<tr><td></td><td></td><td>Info</td><td>0x70000</td><td>This level will log information concerning the current state of the EAP method.</td></tr>
<tr><td></td><td></td><td>Debug</td><td>0xf0000</td><td>This will show information such as packet dumps.</td></tr>
<tr><td colspan="5">NOTE: Defining a level means you will receive the information defined by the specific level and all the levels below it. For example defining Info will instruct SecureW2 to log Errors, Warnings and Info messages.</td></tr>
<tr><td colspan="5">NOTE: FileTraceMask is available from version 2.x.x and on.</td></tr>
</table>

Other SecureW2 components that offer basic logging (ON or OFF) can be enabled by setting the following registry key to “1”:

Configuration:

- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\SECUREW2MGR\EnableFileTracing

Installation:

- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\SECUREW2PLUGIN\EnableFileTracing

Service:

- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\SECUREW2SERVICE\EnableFileTracing

2. Rebranding Client with Institution logo

It is highly recommended you utilize the SecureW2 rebranding option. This allows an institution to show a logo in all the SecureW2 user interfaces.

When SecureW2 is required to show a user interface, it will look for the following bitmap which must be 146x97 pixels:

%programfiles%\SecureW2\sw2_res_default.bmp

When found it will be displayed in the top left corner of the SecureW2 user interface as shown in this example:

This will require you to create an installer that adds the logo to the folder and runs the SecureW2 installer. This is usually done in combination with the pre-configuration file allowing you to create the necessary configuration for your users.

Examples of wrappers can be found in our learn section of the website and by downloading the WIX package (SecureW2_Enterprise_Client_WIX_00x.zip) from the SecureW2 portal. This contains the application and scripts necessary to build a MSI package with your own configuration and logo.

This zip file also contains a README.txt with the available option

3. Pre-Configuration

The SecureW2 installer searches for a pre-configuration file: *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.

3.1 Building your own setup

For organizations willing to distribute SecureW2 within their organization with their own SecureW2.inf it is possible to package the SecureW2.inf, SecureW2_xxx.exe and certificates in an installation program that unpacks the file to a temporary directory and then runs the installer.

3.1.1 Command line switches

The following command line switches can be used when calling the SecureW2_xx.exe installer from a third party installation program.

3.1.1.1 /PEAP <flag>

This flag controls whether the PEAP method is installed or not. Example ”/PEAP 1” will install the PEAP method. ”/PEAP 0” will not install the PEAP method.

3.1.1.2 /TTLS <flag>

This flag controls whether the TTLS method is installed or not. Example ”/TTLS 1” will install the TTLS method. ”/TTLS 0” will not install the TTLS method.

3.1.1.3 /GTC <flag>

This flag controls whether the GTC method is installed or not. Example ”/GTC 1” will install the GTC method. “/GTC 0” will not install the GTC method.

3.1.1.4 /Service <flag>

This flag controls whether the SecurW2 Service is installed or not. Example ”/Service 1” will install the service. “/Service 0” will not install the Service.

3.1.1.5 /Tray <flag>

This flag controls whether the SecurW2 Tray is installed or not. Example ”/Tray 1” will install the service. “/Tray 0” will not install the Tray.

3.1.1.6 /InstallLog <flag>

This flag controls whether the SecurW2 Installer log is enabled before installation or not. Example ”/InstallLog 1” will enable the Installer logging . “/Tray 1” will disable .

3.1.1.7 /S

SecureW2 supports a silent install via the command line. Calling the installer with the "/S" option will instruct SecureW2 to not show any screens. The restart required for setting up the EAP methods on Windows XP SP3 and Vista SP0 has to be done manually.

3.1.1.7.1 /S /Q

Calling the installer with the "/S /Q" options will instruct SecureW2 to not show any screens except for error and warning screens. The restart required for setting up the EAP methods on Windows XP SP3 and Vista SP0 has to be done manually.

3.1.2 NSIS

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

3.1.3 MSI

We also offer a MSI package that allows you to build a MSI package to deploy your installation. Please contact secureW2 for more info.

3.2 Basic INF File

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

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

```
1 [Version]
2 Signature   = "$Windows NT$"
3 Provider    = "SecureW2"
4 Config      = 7
```

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:

```
1 ; This is a comment
```

Example of a enabled/disabled line in the INF file:

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

3.3 Global Configuration

3.3.1 [WZCSVC]

Use this section to set the state of the *Wireless* Zero Configuration Service of Microsoft.

Option	Description	Type	Default value
Startup	This will start the WZCSVC service. **AUTO** indicates the service startup type will be changed to automatic and the service will be started. **START** indicates the service startup type will not be changed and the service will be started. **STOP** indicates the service startup type will not be changed and the service will be stopped. **DISABLED** indicates the service will be disabled and the service will be stopped.	String	*AUTO*
Restart	**true** indicates the service will be re-started before installation. **false**	Boolean	*false*
Both options will start the WZCSVC before installation.			

NOTE: Use the Restart option ONLY if you are having problems using a Cisco VPN client via the same adapter.

Example:

```
1 [WZCSVC]
2 Startup = AUTO
3 Restart = TRUE
```

3.3.2 [DOT3SVC]

Use this tag to set the state of the *Wired* Zero Configuration Service of Microsoft.

Option	Description	Type	Default Value
Startup	This will start the DOT3SVC service. **AUTO** indicates the service startup type will be changed to automatic and the service will be started. **START** indicates the service startup type will not be changed and the service will be started. **STOP** indicates the service startup type will not be changed and the service will be stopped. **DISABLED** indicates the service will be disabled and the service will be stopped.	String	*AUTO*
Restart	**true** indicates the service will be re-started before installation. **false**	Boolean	*false*
Both options will start the DOT3SVC before installation.			

NOTE: Use the Restart option ONLY if you are having problems using a Cisco VPN client via the same adapter.

Example:

```
[DOT3SVC]
Startup = AUTO
Restart = TRUE
```

3.3.3 Certificates

In the Certificates section you must define your certificate chain.

Option	Description	Type	Default Value
Certificate.n	Set the location of the certificate relative to the SecureW2.inf file. The value “n” should start at 0 or 1 and be incremented with each new certificate. Currently only DER encoded X.509 certificates are supported.	String	*empty*
Certificate.0 is optional but if defined 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
```

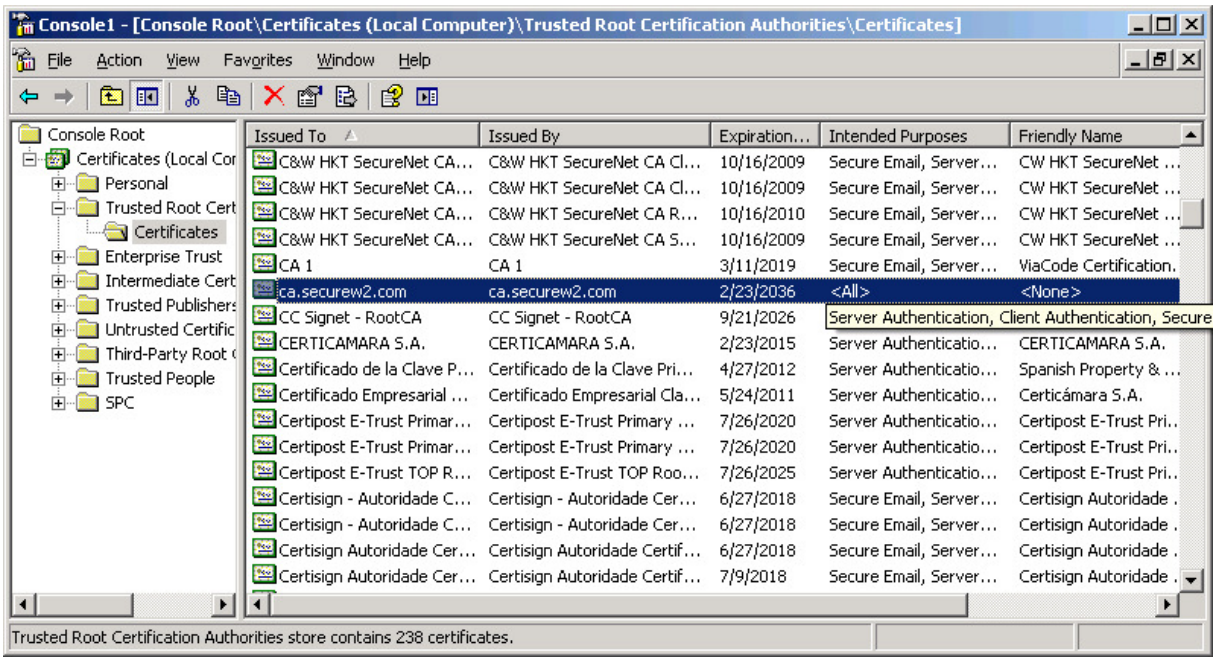
3.3.3.1 Retrieving/Converting Certificates

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:



To use the MMC do the following:

1. Click on the “Start” menu
2. Click on the “Run” option
3. 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.)
4. Select “File” in the top menu
5. 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.
6. 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.
7. Select the “Certificates” snap-in and click on the “Add” button.
8. When asked which certificates are to be managed select “Computer account”.
9. When asked for which computer the certificates are to be managed select “Local computer” and click on “Finish”.
10. Click on “Close” to return to the “Add/Remove snap-in” window that now shows the “Certificates” snap-in.
11. Click on “Ok” to return to the main MMC window.
12. 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”.
13. Look for your CA certificates and right click on the certificate and select “All Tasks” and then ”Export…”.
14. 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.

3.3.4 Service

In the Service section you can configure the behavior of the SecureW2 service.

Option	Description	Type	Default Value
DisableWirelessOnWired	Setting this option to true will instruct the SecureW2 service to disable the Wireless connectivity of the computer when a valid Ethernet connection is available.	Boolean	*false*

Example:

```
1 [SERVICE]
2 DisableWirelessOnWired = TRUE
```

NOTE: to use this option you must select the Service component during installation

3.4 LAN Configuration

3.4.1 [LAN]

Currently the configuration is applied to all wired adapters.

The following options MUST be defined:

Option	Description	Type
Profile	The SecureW2 TTLS Client Profile that is to be used for the LAN connection.	String

You MAY define the following optional options (Options that are not defined in the LAN section will be set to their Default value):

Option	Description	Type	Default Value
EAPType	This option determines which EAP type to use. Possible options are 21 = TTLS, 25 = PEAP.	Numeric	*21*
OneXEnforced	Force the adapter to use 802.1X	Boolean	*false*
OneXEnabled	Enable 802.1X for this adapter	Boolean	*true*
CacheUserData	Specifies whether the user credentials are cached for subsequent connections. When cacheUserData is TRUE, the credentials are cached.	Boolean	*true*
AuthMode	Specifies the type of credentials used for authentication, see 3.4.1.1 for more information.	String	*machineOrUser*

3.4.1.1 AuthMode

Value	Description
machineOrUser	Use machine or user credentials. When a user is logged on, the user's credentials are used for authentication. When no user is logged on, machine credentials are used.
machine	Use machine credentials only.
user	Use user credentials only.
guest	Use guest (empty) credentials only.

3.5 SSID Configuration

The following describes how to configure the different SSID ytou want to use and enable for SecureW2.

3.5.1 [SSID.n]

Each **[SSID.n]** section describes a different SSID configuration 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 options:

Option	Description	Type
SSID	Use this to set the name of the SSID.	String
Name	Use this to set the name of the SSID profile in Windows. This options allows one to define multiple profiles for the same SSID. **This only works in Vista, in Windows 2K and Windows XP the last defined profile is used**	String
Profile	The SecureW2 TTLS Client Profile that is to be used for this SSID	String

IMPORTANT: Do not use spaces in the profile string, this might cause problems when having two profile such as "Eduroam Wireless" and "Eduroam".

Example configuration:

```
[SSID.1]
Name = INSECURESSID
SecurityType = "WPA2-Personal"
EncryptionType = "AES"
NonBroadcast = "false"
SharedKey = "xxx"
UserPermission = "execute"

[SSID.2]
Name = SECURESSID
Profile = "SW2"
AuthenticationMode = "WPA2
EncryptionType = "TKIP"
ConnectionMode = "auto"
ConnectionType = "ESS"
NonBroadcast = "true"
AutoSwitch = "false"
UserPermission = "read"
```

Per SSID section you MAY define the following optional options (Options that are not defined in a SSID section will be set to their Default value):

Option	Description	Type	Default Value
SecurityType	See chapter 3.5.1.1 SecurityType.	String	*If not defined then the AuthenticationMode will be used.*
AuthenticationMode	Use this to set the type of authentication mode for this SSID.	String	*open*
EncryptionType	Use this to set the type of encryption for this SSID.	String	*WEP*
NonBroadcast	Use this to indicate if the SSID is broadcasted by the Access Point or not.	Boolean	*false*
ConnectionMode	Use this to if the SSID should be connected to automatically or by manually. Valid options are *auto* and *manual*	String	*auto*
ConnectionType	Use this to set the operating mode of the network. Valid options are *ESS* (infrastructure network) and *BSS* (ad-hoc network).	String	*ESS*
UseComputerCredentials	This controls the "Use computer credentials if available" option. Setting this to TRUE will tell Windows to use computer credentials, if available, to authenticate (during startup).	Boolean	*true*
UseGuestCredentials	This controls the "Use guest credentials if available" option. Setting this to TRUE will tell Windows to use guest credentials, if available, to authenticate.	Boolean	*False*
EAPType	This option determines which EAP type to use. Possible options are 21 = TTLS, 25 = PEAP.	Numeric	*21*
SharedKey	The option can be used to define the SharedKey used for the WEP and WPAPSK protocols.	String	*N/A*
UserPermission	See chapter: 3.5.1.3 UserPermission.	String	*write*
The following is only available in Windows Vista and higher			
AutoSwitch	This option determines the roaming behaviour of an auto-connected network when a	Boolean	*false*

	more preferred network is in range.		
CacheUserData	Specifies whether the user credentials are cached for subsequent connections. When cacheUserData is TRUE, the credentials are cached.	Boolean	*true*
AuthMode	Specifies the type of credentials used for authentication, see 3.4.1.1 for more information. **NOTE:** **use this instead of UseComputerCredentials or UseGuestCredentials**	String	*machineOrUser*

NOTE: The options described above are case sensitive. Best practice is to use the case shown in this document

The following options control the supplicant behavior:

Option	Description	Type	Default Value
HeldPeriod	The heldPeriod (OneX) element specifies the length of time, in seconds, in which a client will not re-attempt authentication after a failed authentication attempt.	String	1
AuthPeriod	The authPeriod (OneX) element specifies the maximum length of time, in seconds, in which a client waits for a response from the authenticator. If a response is not received within the specified period, the client assumes that there is no authenticator present on the network.	String	*18*
StartPeriod	The startPeriod (OneX) element specifies the length of time, in seconds, to wait before an EAPOL-Start is sent. An EAPOL-Start message is sent to start the 802.1X authentication process.	String	*5*
MaxStart	The maxStart (OneX) element specifies the maximum number of EAPOL-Start messages sent. After the maximum number of EAPOL-Start messages has been sent, the client assumes that there is	String	*3*

	no authenticator present on the network.		
MaxAuthFailures	The maxAuthFailures (OneX) element specifies the maximum number of authentication failures allowed for a set of credentials.	String	*1*
SupplicantMode	The supplicantMode (OneX) element specifies the method of transmission used for EAPOL-Start messages. See 3.5.1.2 for more info.	String	compliant

3.5.1.1 SecurityType

Use this option to set the Security used for this SSID. The following options are available:

Option
Open
Shared
WPA-Enterprise
WPA2-Enterprise
802.1X
WPA-Personal
WPA2-Personal

NOTE: The option, 'SecurityType' is available from version 3.5.3 and onwards.

3.5.1.2 SupplicantMode

Mode	Description
inhibitTransmission	EAPOL-Start messages are not transmitted. Valid for wired LAN profiles only.
includeLearning	The client determines when to send EAPOL-Start packets based on network capability. EAPOL-Start messages are only sent when required. Valid for wired LAN profiles only.
Compliant	EAPOL-Start messages are transmitted as specified by 802.1X. Valid for both wired and wireless LAN profiles.

3.5.1.3 UserPermission

This option controls whether a standard user (non Admin) can change the SSID configuration or connect/disconnect from/to the SSID. The following options are available:

Option	Description
read	The standard user can see the SSID configuration, but not change anything. The standard user is **NOT** able to manually connect or disconnect to/from this SSID.
execute	The standard user can see the SSID configuration, but not change anything. The standard user is able to manually connect or disconnect to/from this SSID.
write	The standard user can see the SSID configuration and change the configuration. The standard user is able to manually connect or disconnect to/from this SSID.

3.5.1.4 Single Sign On (SSO)

Per SSID section you MAY define the following optional options to enable Single Sign On (also known as PLAP):

Option	Description	Type	Default Value
UseSSO	Enable SSO for this SSID	Boolean	*FALSE*
SSOType	“preLogon” or “postLogon”	String	*preLogon*
SSOMaxDelay	Specifies, in seconds, the maximum delay before the single sign-on connection attempt fails	Numeric	*10*

```
1 [SSID.1]
2 Name = SSOSSID
3 Profile = "SSO"
4 UseSSO = TRUE
5 SSOType = "preLogon"
6 SSOMaxDelay = "10"
```

Single Sign On will only work if the computer has been joined to an Active Directory Domain. If everything has been configured correctly you will see the following text underneath the users login icon: “Windows will try to connect to <SSID name>".

3.6 Profile Configuration

The following describes how to configure the different SecureW2 Client Profiles you want to use.

3.6.1 [Profile.n]

Each **[Profile.n]** section describes a different Profile configuration where **n** is the number of the Profile section. This number must start at 1 and be incremented with each new Profile section.

Per Profile section you MUST define the following options:

Option	Description	Type
Name	Use this to set the name of the Profile.	String

Per Profile section you MAY define the following optional options (Options that are not defined in a Profile section will be set to their Default value):

Option	Description	Type	Default Value
General			
Description	The SecureW2 TTLS Client Profile that is to be used for this SSID.	String	empty
Localization			
AltUsernameString	Allows you to overide the Username label in the user interface	String	empty
AltPasswordStr	Allows you to overide the Password label in the user interface	String	empty
AltRePasswordStr	Allows you to overide the Second Password label in the user interface	String	empty
AltDomainStr	Allows you to overide the Domain label in the user interface	String	empty
AltCredsTitle	Allows you to overide the titel of the windows in the user interface	String	empty
AltProfileStr	Allows you to overide the Profile label in the user interface	String	empty
Connection			
UseAlternateOuterIdentity	This option instructs SecureW2 to use an alternate outer identity.	Boolean	true
UseAnonymousOuterIdentity	This option controls which alternate outer identity SecureW2 uses. true *indicates the outer identity will be anonymous@domain.* false *instructs SecureW2 to use the value defined in* AlternateOuterIdentity	Boolean	true
AlternateOuterIdentity	The value defined by this attribute is used as the outer identity. This option is only valid if UseAnonymousIdentity is **false**	String	*empty*

UseSessionResumption	Enabling this option instructs SecureW2 to use session resumption (quick connect)	Boolean	false
Certificates			
VerifyServerCertificate	Enabling this option instructs SecureW2 verify the TTLS server certificate	Boolean	true
TrustedRootCA.n	Each TrustedRootCA.n option defines a root certificate that will be trusted by SecureW2 where "n" is the number of the TrustedRootCA option. This number must start at 0 and be incremented with each new TrustedRootCA option. This option often coincides with the global certificate configuration. The value of this option 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. 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. Using openssl use the following command: *openssl sha1 < ttls.cer*. The hexadecimal string should not contain spaces.	String (Hex)	empty
VerifyServerName	Use this option to indicate you want to verify the ServerName. Use together with the option ServerName.	Boolean	false
ServerName	Use this option to define value that will be used to verify the common name in the certificate of the TTLS server.	String	empty
Authentication			
AuthenticationMethod	Use this option to define the inner authentication method used by SecureW2 to authenticate the user. Currently this can be two values: 'PAP' or 'EAP'.	String	PAP
InnerEAPType	If EAP has been selected as the AuthenticationMethod the value defined by this option is the EAP-Type that is to be used. The following EAP methods are available (depending on the EAP methods installed on the local computer): '4' (EAP-MD5) , '26' = EAP-MSCHAP v2	Numeric	0
InnerEAPConfigBlob	Use this to configure inner EAP methods. For example to enable EAP-MSCHAPV2 to use “Windows	String (Hex)	empy

	credentials to logon the user use the following hex string: “0100000002000000”		
User account			
PromptUserForCredentials	Set this option to instruct SecureW2 to prompt the user for credentials during authentication.	Boolean	true
UserName	If the option PromptUserForCredentials is set to *false* then setting this value to *PROMPTUSER* instructs the SecureW2 installer to prompt the user for credentials during installation.	String	empty
UserDomain	Set this option to the pre-configured domain name to use during installation.	String	empty
UseUserCredentialsForComputer	Enabling this option instructs SecureW2 to use the user credentials to logon the computer.	Boolean	false
Advanced			
UseAlternateComputerCredentials	Enabling this option instructs SecureW2 to use the following three credentials for a computer logon.	Boolean	false
ComputerName	The user name to use during a computer logon.	String	empty
ComputerPassword	The password to use during a computer logon.	String	empty
ComputerDomain	The domain to use during a computer logon.	String	empty
ServerCertificateOnLocalComputer	Enabling this option instructs SecureW2 to verify if the TTLS certificate is installed on the local computer.	Boolean	false
CheckForMicrosoftExtension	Enabling this option instructs SecureW2 to verify if the TTLS certificate contains the correct Microsoft Extended Key Usage.	Boolean	false
AllowNewConnections	Enabling this option instructs SecureW2 to allow users to setup new connections.	Boolean	false
UseMicrosoftCachedUserData	Enabling this option instructs SecureW2 to use the cached user data provided by Microsoft Windows.	Boolean	false
UseEmptyOuterIdentity	This option instructs SecureW2 to use an empty *outer anonymous identity that is compliant with RFC 4282. Requires* UseAlternateOuterIdentity *and* UseAnonymousOuterIdenity *to be*	Boolean	false

	enabled.		
RenewIPAddress	Enabling this option instructs SecureW2 renew the DHCP IP Address of the authenticating adapter.	Boolean	false
BypassBalloonNotification	Setting this option to true instructs SecureW2 to bypass the Microsoft Balloon notification and present the credentials dialog on screen directly. **NOTE: to use this option you must select the Tray component during installation**	Boolean	false
UseConfigurationLockdown	Setting this option prevent users (normal and admin) from changing the configuration after installation. **NOTE: only one profile is supported for configuration lockdown.**	Boolean	False
DisableSaveUserCredentials	Setting this to true disables the "Save credentials" option in the popup.	Boolean	false
AlwaysSaveUserCredentials	Setting this to true will always check the "Save credentials" option in the credentials popup.	Boolean	false
HideDomain	Setting this option will hide the “Domain”field in the credentials popup.	Boolean	false
ForceSecureW2UserInterface	Setting this option will instruct SecureW2 to always use the main Credentials interface for authentication.	Boolean	false

3.6.2 [Profile.n.GTC]

The **[Profile.n.GTC]** section allows one to configure the inner EAP method GTC where **n** is the number corresponding to the main **[Profile.n]**.

Option	Description	Type	Default Value
BypassBalloonNotification	Setting this option to true instructs SecureW2 to bypass the Microsoft Balloon notification and present the credentials dialog on screen directly	Boolean	False
UseCredentialsUserInterface	Setting this option instructs the GTC method to request the username and passcode up front.	Boolean	False
CheckForSoftwareToken	Setting this option instructs the GTC method to look for software tokens and if present add a drop down selection of the available softokens.	Boolean	False

4. SecureW2 SysCheck

SecureW2 SysCheck, available in 3.5.0 onwards, allows administrators to verify the system configuration of an End User before installation. SecureW2 SysCheck also offers the option of ongoing verification of the system configuration at administrator configurable intervals.

SecureW2 SysCheck consists of two components, Enterprise Client that gathers the information and SecureW2 SysCheck Server that receives and analyzes this information. SysCheck Server subsequently instructs Enterprise Client with custom responses to the End User depending on a set of administrator defined rules.

SecureW2 SysCheck runs in two modes, Offline and Online. Offline requires no internet connectivity and verifies if certain third party products are installed on the users system. Online verifies system information against a remote server and requires internet connectivity.

SecureW2 SysCheck Online gathers the following attributes per MAC Address:

- Operating System
  - OSArchitecture (32-bit/64-bit)
  - BuildNumber
  - ServicePackMajorVersion
  - ServicePackMinorVersion
- Driver
  - DriverProviderName
  - Manufacturer
  - DriverDate
  - DriverVersion
- SecureW2 Client Version

SecureW2 SysCheck supports the following custom responses, based on received SysCheck information:

- Show custom messages to the user on a per attribute basis
- Redirect user to a remediation website on a per attribute basis
- Abort the installation
- Combinations of the above

4.1 SysCheck Enterprise Client

4.1.1 SysCheck.Offline

SecureW2 SysCheck Offline is enabled in the SecureW2.inf file by adding the **[SYSCHECK.OFFLINE]** section. The following options are available:

Option	Description	Type	Default Value
Enabled	Enable SysCheck Offline	Boolean	FALSE
ProductName.n	Each ProductName.n option defines a substring that will be used to match against third party application installed on the system.	String	Required
WarningMessage	This option allows an Administrators to customize the message shown to the user when a Productname.n matches. The DEFAULT string used is:	String	Required

	“SecureW2 detected the following third party wireless applications:%sOn occasion these applications can cause a conflict, if you run into a problem please consider uninstalling them to resolve network connectivity issues” The “%s” will be replaced with the summary of the thirdparty applications that were found.		

```
[SYSCHECK.OFFLINE]
   1 Enabled = TRUE
   2 ProductName.1 = "DW WLAN"
   3 ProductName.2 = "Broadcom Management Programs"
   4 ProductName.3 = "Intel(R) PROSet/Wireless WiFi Software"
```

The above will look for the following 3 applications: “DW WLAN”, ”Broadcom Management Programs” and ”Intel® PROSet/Wireless WIFI Software” and show the following message to the user:

4.1.2 SysCheck.Online

SecureW2 SysCheck Online is enabled in the SecureW2.inf file by adding the **[SYSCHECK.ONLINE]** section. The following options are available:

Option	Description	Type	Default Value
Enabled	Enable SysCheck Online	Boolean	FALSE
Server	Specify the server domain name hosting the SysCheck script (i.e. server.organization.com).	String	Required
Service	Specify the script URL (i.e. /sysCheck/sysCheckService.php).	String	Required
Port	Specify the Server port number.	Numeric	443
UpdateTimeout	Specifies the interval of the ongoing SysCheck system verification. **NOTE: This does not apply to the check performed during installation**	Numeric	2592000 (30 days)

```
[SYSCHECK.ONLINE]
   1 Enabled = TRUE
   2 Server = www.securew2.com
   3 Port = 443
   4 Service = sysCheckService.php
```

The above example will send the gathered information to www.securew2.com/sysCheckService.php on port 443.

4.2 SysCheck Server

The SysCheck Server consists of a PHP capable web server running the SecureW2 SysCheck scripts and a database backend required to store the SysCheck configuration and gathered system attributes.

4.2.1 Requirements

- MySQL Database 5.x or higher
- PHP capable Web Server with HTTPS
- PHP 4.x or higher

4.2.2 Database Backend Setup

The following steps show how to setup the database backend for SecureW2 SysCheck:

1. Create a database to store the SysCheck information (http://dev.mysql.com/doc/refman/5.1/en/create-database.html);

2. Create a database user that has the following access to the database created in Step 1:
   a. SELECT, INSERT, UPDATE

3. Copy the MySQL file located in the SecureW2 SysCheck package, "SysCheck.sql", to a temporary location on your server. Edit the following line in the file:

   USE `SysCheck`;

   to reference the database you created in Step 1.

4. Run the following command on the server with the appropriate username/password information to load the database information (http://dev.mysql.com/doc/refman/5.1/en/mysql.html):

   ```
   mysql --user=username -password < SysCheck.sql
   ```

   This command will prompt for the password.

If the script is executed successfully it will create the following 4 tables that are used for SecureW2 SysCheck:

Table	Description
configuration	This table contains the configuration information used by SecureW2 SysCheck
reported_adapters	This table contains the reported adapter information
supported_adapters	This table contains the supported adapter information
supported_client	This table contains the support client information

4.2.2.1 Configuration

The configuration table contains the following items:

Item (Column)	Description
defaultContinueMode	Set this to the default ContinueMode that is used when returning information back to the client. See ContinueMode for more information.
defaultRemediationURL	Set this to the default URL that is used when returning information back to the client.
templateErrorClientVersion	This template is used to create the message sent back to the client when a client does not meet requirements.
templateErrorDriverVersion	This template is used to create the message sent back to the client when a Driver Version does not meet requirements.
templateErrorDriverDate	This template is used to create the message sent back to the client when a Driver Date does not meet requirements.

4.2.2.1.1 ContinueMode

The ContinueMode controls the way the Enterprise Client behaves when receiving a response from the SysCheck server.

ContinueMode	Description
0	DEFAULT, only report.
1	Warn the user with the appropriate message.
2	Direct the user to the remediation URL specified in RemediationURL. For more information see RemediationURL.
4	Abort the installation.

A combination of the above:

ContinueMode	Description
3	Warn the user and show remediation URL.
5	Warn the user and abort the installation.
6	Show remediation URL and abort the installation.
7	Warn the user, show remediation URL and abort the installation.

4.2.2.2 RemediationURL

If ContinueMode is configured to open a browser and redirect to an URL, the RemediationURL item is the value of the URL.

NOTE: The RemediationURL must start with “http://” or “https://”.

4.2.2.3 Templates

The templateXXX items in the configuration table are used to create the message that is to be displayed to the End User. The Administrator can customize these templates to meet their needs, such as Localized language messages.

The following keywords can be used in the templates by the Administrator that will be replaced by SecureW2 SysCheck:

- %REPORTEDVERSION%
- %SUPPORTEDVERSION%
- %REPORTEDDATE%
- %SUPPORTEDDATE%
- %REPORTEDDESCRIPTION%

For example in the following template the %REPORTEDVERSION% and %SUPPORTEDVERSION% is replaced with the information gathered from the Enterprise Client and what was configured in the database:

“Invalid Client Version: "%REPORTEDVERSION%" "%SUPPORTEDVERSION%" required)”

In the case the Enterprise Client reported version 3.4.6 and 3.5.0 is the supported version configured in the database the message to the user would look like:

Invalid Client Version: "3.4.6" ("3.5.0" required).

Another example of a template string is when an invalid Driver Version is detected:

SecureW2 detected an invalid Driver Version ("%REPORTEDVERSION%") for adapter "%REPORTEDDESCRIPTION%". Driver version "%SUPPORTEDVERSION%" is required.

In the case of driver 1.1 being reported and driver version 1.2 being supported. The following message will be displayed to the user:

SecureW2 detected an invalid Driver Version ("1.1") for adapter “Dell Wireless 1505 Draft 802.11n WLAN Mini-Card". Driver version "1.2" is required.

4.2.2.4 Reported_adapters

The reported_adapters table contains all the attributes gathered per adapter (MAC Address).

Item (Column)	Description
keyValue	This value contains the value that was used to identify the Client that reported the information. At this moment this is the MAC address of the reported adapter.
attributeName	This value contains the name of the reported attribute.
attributeValue	This value contains the value of the reported attribute.

4.2.2.5 Supported_adapters

The supported_adapters table is where the Administrator configures the rules determining which driver version and driver date is supported per OS. The following items are available.

Item (Column)	Description	Type	Default Value
Id	Id of supported adapter rule	Numeric	Generated by system
BuildNumber	This value specifies the BuildNumber of the OS this rule applies too. **NOTE:** **The value will be used as a substring match of the BuildNumber reported by the Client. So the value:** **"760"** **will match the following (Windows 7) builds:** **"7600" and** **"7601" (SP1)**	String	Required
ServicePackMajorVersion	This value specifies the BuildNumber of the OS this rule applies too.	String	Required
ServicePackMinorVersion	This value specifies the BuildNumber of the OS this rule applies too.	String	Required
DriverProviderName	This value specifies the DriverProviderName this rule applies too. **NOTE:** **The value will be used as a substring match as described in Buildnumber.**	String	Required
Manufacturer	This value specifies the Manufacturer value this rule applies too. **NOTE:** **The value will be used as a substring match as described in Buildnumber.**	String	Required
DriverDate	This value specifies the minimum DriverDate for this rule.	String	Required
DriverVersion	This value specifies the minimum DriverVersion for this rule.	Sting	Required
ContinueMode	This value overloads the default ContinueMode defined in the configuration table.	String	(NULL)
RemediationURL	This value overloads the default ContinueMode defined in the configuration table.	String	(NULL)
Label	This value is used to allow the rule to be applied to configuration Labels defined in the SecureW2.inf pre-configuration file	String	(NULL)
Description	This value specifies the adapter Description this rule applies too. **NOTE:** **The value will be used as a substring match as described in Buildnumber.**	String	(NULL)

NOTE: BuildNumber, ServicePackMajorNumber and ServicePackMinorVersion all support the wildcard values "*" and "". If a an item in the rule contains a wildcard the item of that rule will apply to all adapters that are being verified.

The following section shows some examples of rules that can be used.

4.2.2.5.1 Rule 1

Applies to all OS, All DriverProviderNames, but only Broadcom as Manufacturer and uses the default ContinueMode and RemediationURL defined in the configuration table.

Item (Column)	Value
BuildNumber	*
ServicePackMajorVersion	*
ServicePackMinorVersion	*
DriverProviderName	*
Manufacturer	Broadcom
DriverDate	20070920000000
DriverVersion	4.170.25.12
ContinueMode	(NULL)
RemediationURL	(NULL)

4.2.2.5.2 Rule 2

Applies to Windows 7, All Service Packs, All DriverProviderNames, but only Intel Corporation as Manufacturer and defines ContinueMode 3 and RemediationURL http://www.intel.com in the configuration table. The Client will respond by showing the user a message and then redirecting the browser of the user to the Intel website.

Item (Column)	Value
BuildNumber	760
ServicePackMajorVersion	*
ServicePackMinorVersion	*
DriverProviderName	*
Manufacturer	Intel Corporation
DriverDate	20070920000000
DriverVersion	4.170.25.12
ContinueMode	3
RemediationURL	http://www.intel.com/support/wireless/detect.htm

4.2.2.6 Supported_client

Option	Description	Type	Default Value
ProductName	This value specifies the Product name this rule applies too.	String	Required
ProductVersion	This value specifies the minimum Product version for this rule.	String	Required
ContinueMode	This value overloads the default ContinueMode defined in the configuration table.	String	(NULL)
RemediationURL	This value overloads the default ContinueMode defined in the configuration table.	String	(NULL)
Label	This value is used to allow the rule to be applied to configuration Labels defined in the SecureW2.inf pre-configuration file	String	(NULL)

4.2.3 SysCheck Script

SecureW2 provides the SecureW2 SysCheck Server functionality in two PHP scripts. These scripts will handle incoming requests and depending on the SecureW2 SysCheck configuration store information in the Database backend and respond to the client.

1. To install the PHP scripts copy the two files located in the SecureW2 SysCheck package:

   - sysCheckService.inc.php
   - sysCheckService.php

   to a location on your webserver accessible by a browser (i.e. /var/www/).

2. Edit the following lines in the “SysCheckService.php” file to reflect your database settings:

   ```
   // database information
   $sysCheckDBUser="syscheck";
   $sysCheckDBPassword="syscheck";
   $sysCheckDBName="syscheck";
   ```

3. After copying and editing the files change the owner and file permissions to reflect the web server security used on the system.

5.SecureW2.inf Examples

For examples and tutorials of how to create a SecureW2.inf please visit the Learn section of our website:

http://www.securew2.com/learn

6. SecureW2 installer errors

ERROR_ACCESS_DENIED (5)

The caller does not have sufficient permissions to create the SSID.

ERROR_NOT_READY (21)

This is most often the case of an outdated/invalid driver.

ERROR_CANTOPEN (1011)
ERROR_CANTREAD (1012)

This is most often an invalid certificate or the file could not be found.

ERROR_NO_MATCH (1169)

The interface does not support one or more of the capabilities specified in the profile. For example, if a profile specifies the use of WPA2 when the NIC only supports WPA, then this error code is returned.

ERROR_BAD_PROFILE (1206)

The profile is invalid, this is most often an incorrectly configured SecureW2.inf. For example you installed TTLS but configured the profile for PEAP.

ERROR_INVALID_STATE (5023)

If this error occurs this is most often due to third party WIFI applications running on the computer. The following applications are known to cause problems:

DW WLAN Card Utility
Broadcom Management Programs
Intel(R) PROSet/Wireless WiFi Software
Thinkvantage Wireless Connections

7.SecureW2 License registration

SecureW2 Enterprise Client customers who license on a per device basis are given access to download the software with an easy to use automatic software license registration system. To remove the administrative hassle of managing licenses SecureW2 has implemented a simple online license registration system that is very direct for the network administrator.

7.2 Software License key registration

Upon installation Enterprise client will attempt to register and download a valid license key based on the organization. Once the license key is downloaded by the software automatically from the securew2.com customer portal on the internet, the customer portal will update the number of license keys used and available to the organization. The registration is done online via access to securew2.com. If a licenses have been purchased, a 16-digit key is generated and downloaded to the device. The Client attempts to reach securew2.com for 30 days after installation of the software. During the 30 days the client will function fully without any restrictions. After 30 days if the registration is not complete the software disables itself. Re-installation of the software will start the license key registration process again for another 30 days.

7.3 Software License key deregistration.

Upon uninstallation Enterprise Client will contact the customer portal at securew2.com and uninstall the license key and make the key available for reuse to the organization. In the case of lost, stolen or data wiped devices the customer can send a report of such devices and SecureW2 will credit back licenses for reuse in the automatic license generation system.