Secure Connection Agent

Administrator's Guide

Latest update: 12 July 2024
Document reference number: 20240712_AG_SCA_1.0.6_EN_1.1

www.thegreenbow.com

Table of contents

Document revision history

Version	Date	Sections/pages concerned	Description of change	Author
1.0	2023-10-12	All	Initial release	FH, BB
1.1	2024-07-12	1	Specified client concerned, rephrased text for greater clarity	
		2.1	Rephrased text for greater clarity	FH, BB
		2.3, 2.4	Added procedure steps, inserted a screenshot	BB
		3.1, 3.2, 3.3, 4.1 & 4.3, 5.3.3	Corrected minor errors, inserted headings, revised text to improve readability	BB

1 Overview

The Secure Connection Agent (SCA) is an add-on for the Windows Enterprise VPN Client. It is part of the extended product offering and serves as a link between VPN Clients and the Connection Management Center (CMC).

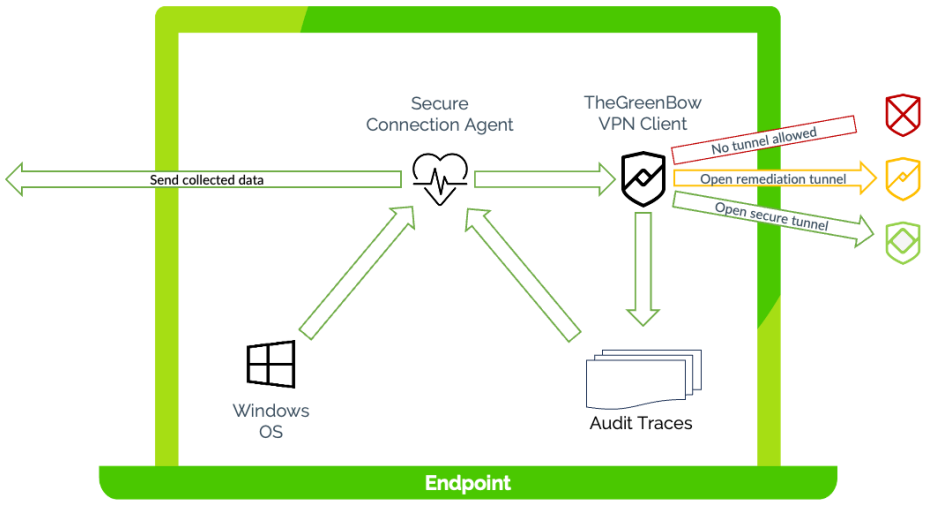
☞ Refer to the Windows Enterprise VPN Client's and the CMC's Administration Guides for more information about these products. You can find them on the Product documentation page on TheGreenBow's website: https://www.thegreenbow.com/en/support/product-documentation/.

The SCA provides the following two functions:

1. Endpoint compliance monitoring: the SCA checks whether the endpoint should be allowed to mount a tunnel in view of accessing the corporate network. The VPN Client will adapt its behavior according to the detected compliance level.
2. Forwarding of the VPN Client's audit traces to the Connection Management Center (CMC).

Function	Minimum version required
Endpoint compliance monitoring	Windows Enterprise VPN Client 7.4
Audit trace forwarding	TheGreenBow VPN Client 6.60

The following diagram shows how the SCA communicates with the VPN Client and the CMC, as well as with the Windows OS:



The SCA communicates with the Windows OS to determine the endpoint's conformity score. Depending on the score, the VPN Client will allow the endpoint to open a connection to the corporate network, to open a

remediation connection (e.g. to update the virus definitions), or it will not allow any tunnel to be opened.

The SCA also transfers to the CMC all audit traces the VPN Client generates.

2 Installing the agent

2.1 Introduction

The Secure Connection Agent can be installed using the MSI installer package `TheGreenBow_SCA.msi`, which TheGreenBow will provide. The software must be installed with administrator privileges.

The Secure Connection Agent must be installed after the VPN Client.

The installation can be customized using the command line. The only option currently available is the following:

MSI custom argument	Effect
`SERVICEMODE=0`	Unless `SERVICEMODE` is set to zero, the SCA will run automatically at Windows startup. Setting `SERVICEMODE` to zero disables the SCA service for customers who only want to use the compliance DLL (see chapter 5 Using the compliance DLL). Default value: `1`

2.2 Minimum requirements

The Secure Connection Agent is available for Windows 10 and 11, 64-bit.

The minimum system requirements are as follows:

- Processor: 1 GHz with x86-64 architecture
- RAM: 2 GB
- Available hard disk space: 10 MB

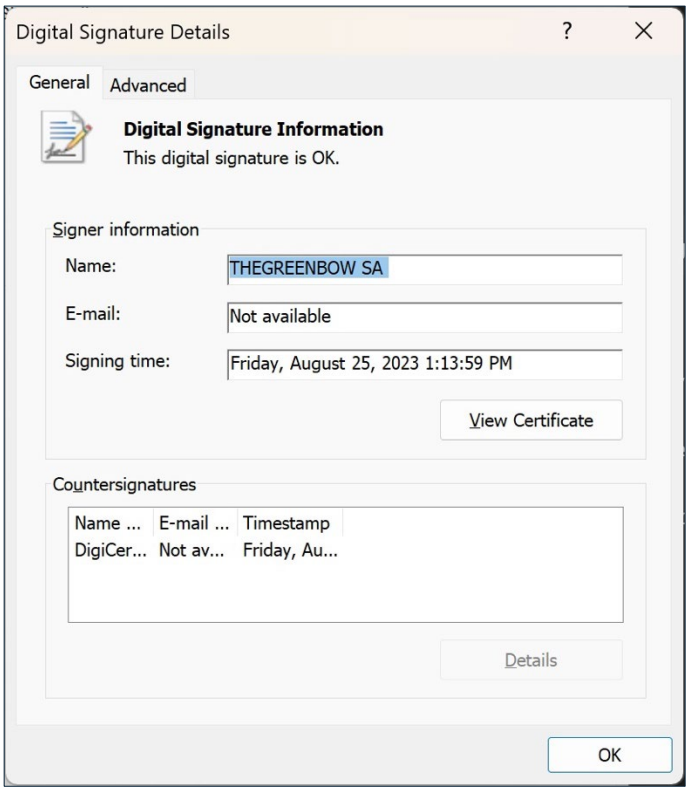
2.3 Digital signature and version

The installer software for the Secure Connection Agent (SCA) is signed with a certificate issued for THEGREENBOW SA. This allows the person performing the installation or the user to verify the integrity of the installation program.

To verify the software's authenticity, proceed as follows:

1. Right-click the MSI installer.

2. From the contextual menu that is displayed, select **Properties**.
3. In the windows that is displayed, select the **Digital Signatures** tab.
4. Select the signature in the signature list, then click **Details**. The digital signature's details are displayed:



Once the software is installed, you can check the SCA's version number by executing the following command in PowerShell:

```
Get-WmiObject -Class Win32_Product | WHERE Vendor -eq
"TheGreenBow"
```

2.4 Installation procedure and installer contents

To install the SCA, proceed as follows:

1. Open a command prompt with administrator privileges.
2. Navigate to the folder where you stored the installer or specify the path to the folder in front of the file name in the command below.
3. Run the following command:

```
msiexec /i TheGreenBow_SCA.msi SERVICEMODE=0
```

The confirmation dialog displayed at the end of the installation procedure may appear behind the progress window. If this occurs, click on the edge of the confirmation dialog to bring it to the foreground so that you can confirm the completion of the installation.

This command creates the directory `%programfiles%\TheGreenBow\TheGreenBow Secure Connection Agent\` with the following contents:

- `tgb_secure_connection_agent.exe`: standalone service
- `tgb_conformity.dll`: DLL used to access compliance information

To facilitate deployment, the MSI installer can be modified to include the CMC's address. To do so, use an MST transform file.

3 Audit trace forwarding

3.1 Introduction

Audit trace forwarding is used to collect the audit traces generated by the VPN Client (stored in the `LogFiles\System` sub-folder) and forward them to the Connection Management Center (CMC).

3.2 Configuring the agent

The Secure Connection Agent (SCA) needs to know the address and port of the server to which it is to forward the audit trace messages. They can both be configured in the self-documenting `settings.toml` file, which is a TOML 1.0.0 file. The file can be found in `%PROGRAMDATA%/TheGreenBow/TheGreenBow Secure Connection Agent/settings.toml`.

You must run the editor with administrator privileges to edit the TOML file.

The parameters that can currently be configured are the CMC syslog server address and the port to be used.

```
[syslog]
# Address of CMC syslog server
address = "cmc.domain.lan"
# Port of CMC syslog server
port = 514
```

The SCA process must be restarted for the changes made to the `settings.toml` file to take effect.

Make sure the firewall does not block the syslog port!

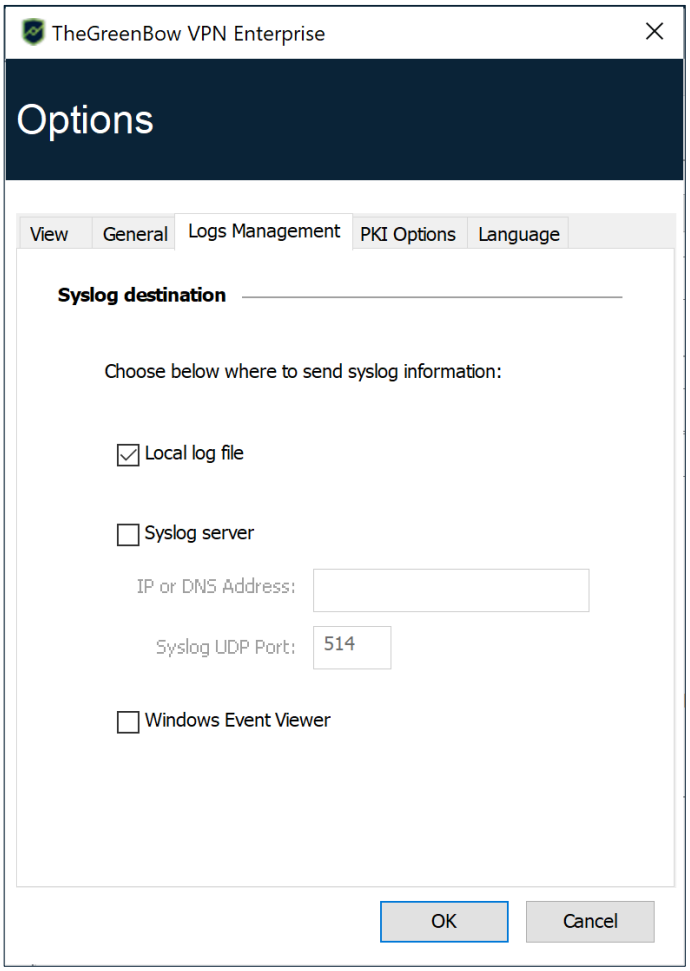
3.3 Configuring the VPN Client

Audit traces can only be forwarded if the VPN Client generates audit traces in the first place!

To enable audit traces, proceed as follows:

1. Access the Windows Enterprise VPN Client's **Configuration Panel**.

2. From the **Tools** menu, choose **Options...**
3. Select the **Logs Management** tab.
4. Check the **Local log file** box.
5. Click **OK**.



☞ Refer to the Windows Enterprise VPN Client's Administrator's Guide for a full description of the various types of logs available.

4 Endpoint compliance

4.1 Introduction

The endpoint compliance function checks the availability and status of the Windows firewall, the availability of an antivirus provider, and whether the virus and threat protection is up to date in Windows Security[1].

Currently there are three levels of compliance defined and the VPN Client will act differently according to each of these levels, as described in the truth table below.

Virus & threat protection		Firewall & network protection		Result
0	+	0	=	 Cannot open any tunnel
1	+	0	=	 Switch to a remediation area
0	+	1		
1	+	1	=	 Access sensitive network

A remediation VPN connection should be considered as a VPN tunnel with restricted access. It could for example allow a system administrator to take control over the PC from the corporate network.

After logging on to Windows, the Secure Connection Agent will use the last known compliance level until the Windows Security Center service has started.

4.2 Configuring the agent

This feature works out of the box and there's no need to configure anything on the Secure Connection Agent's side.

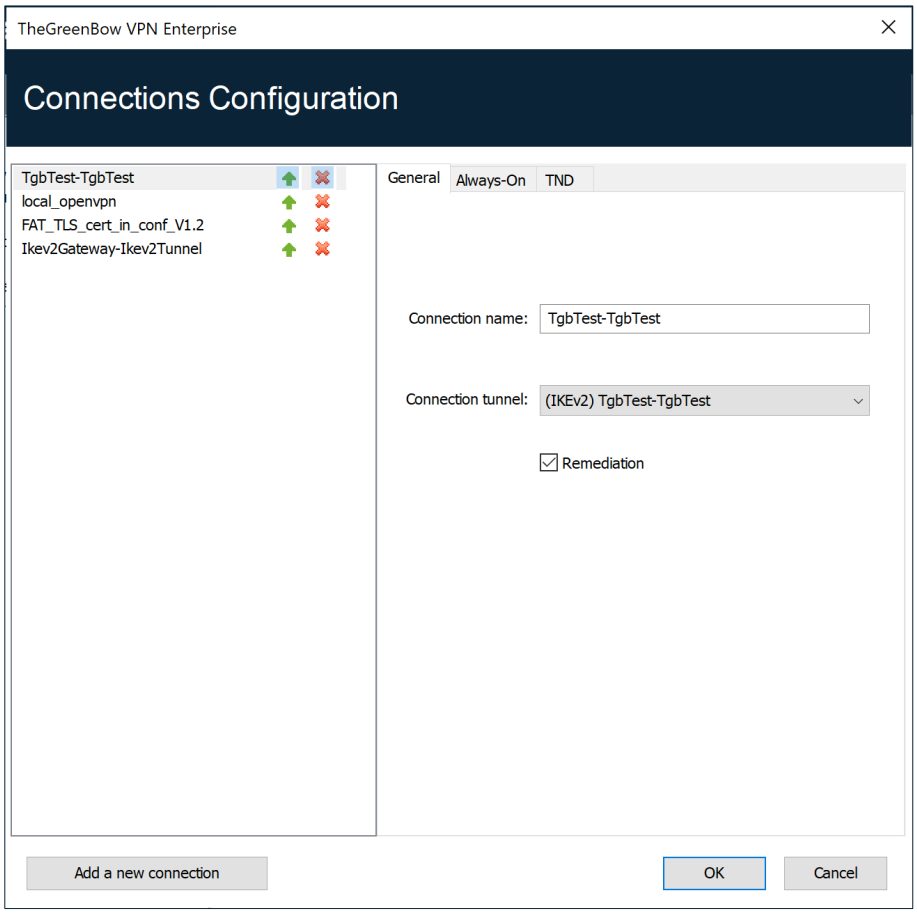
[1] Windows Security Center in Windows 10.

4.3 Configuring the VPN Client

When the Secure Connection Agent (SCA) detects near compliance, a remediation connection will be opened if such a connection has been configured.

To configure a remediation connection, proceed as follows:

1. Access the Windows Enterprise VPN Client's **Configuration Panel**.
2. From the **Tools** menu, choose **Connections Configuration** to open the **Connections Configuration** window.
3. On the **General** tab, check the **Remediation** box for the connection to be used as a remediation connection.



Only one connection should have the **Remediation** box checked. When the **Remediation** box is checked for several connections, there is no certainty as to which connection will be used.

With version 7.4 of the Windows Enterprise VPN Client, the Secure Connection Agent must have been enabled at least once. Otherwise, the checkbox will not be shown in the **Connections Configuration** window.

5 Using the compliance DLL

5.1 Introduction

The Secure Connection Agent (SCA) includes a compliance DLL that works in the background to send the collected data to the Connection Management Center (CMC). TheGreenBow provides an API for customers who want to access the SCA's compliance features through the DLL in view of integrating them with third-party products.

The following sections describe how to load the DLL and use the API to achieve this.

5.2 Loading the library

Once the SCA has been installed:

- The caller must check that the DLL `tgb_conformity.dll` is signed by `THEGREENBOW`
- If the signature is correct, use a standard `LoadLibrary` call to load the DLL (i.e. explicit linking of DLL)

```
LoadLibrary("C:\Program Files\TheGreenBow\TheGreenBow
Secure Connection Agent\tgb_conformity.dll")
```

5.3 API

5.3.1 Overview

The DLL has been compiled for an x86-64 architecture. The calling convention is `STDCALL`. The DLL's interface is defined in ISO C11 and specified by the following:

```
typedef enum
{
    TGB_CONFORMITY_GET_OK = 0,
    TGB_CONFORMITY_GET_UNKNOWN_ID = 1,
    TGB_CONFORMITY_GET_INVALID_ARGUMENTS = 2,
    TGB_CONFORMITY_GET_INTERNAL_ERROR = 3
} TGBGetConformityItemErrorCode;

typedef enum
{
    TGB_FIREWALL_CONFORMITY = 1,
    TGB_ANTIVIRUS_CONFORMITY = 2
} TGBGetConformityItemID;

typedef enum
{
    TGB_CONFORMITY_LEVEL1 = 1,
    TGB_CONFORMITY_LEVEL2 = 2,
    TGB_CONFORMITY_LEVEL3 = 3,
    TGB_CONFORMITY_LEVEL4 = 4

} TGBConformityLevel;

TGBGetConformityItemErrorCode TGBGetConformityItem(
    TGBGetConformityItemID ConformityItemID,
    TGBConformityLevel*    pConformityLevel,
    const char**           ppComment
);

void TGBFreeCommentString(const char* comment);
```

5.3.2 Error codes

The `TGBGetConformityItem` function returns one of the following error codes:

Error code	Meaning
`CONFORMITY_GET_OK`	OK
`CONFORMITY_GET_UNKNOWN_ID`	Error: unknown ID
`CONFORMITY_GET_INTERNAL_ERROR`	Error: cannot retrieve item
`TGB_CONFORMITY_GET_INVALID_ARGUMENTS`	Error: one or several input parameters were not valid

5.3.3 Description of parameters

The following parameters are used:

Name	Type	Description
`ConformityItemId`	Input parameter	The compliance item to be retrieved: • `TGB_FIREWALL_CONFORMITY`: retrieve compliance item for firewall status on endpoint • `TGB_ANTIVIRUS_CONFORMITY`: retrieve compliance item for antivirus status on endpoint
`pConformityLevel`	Output parameter	Conformity level of the requested item • `TGB_CONFORMITY_LEVEL1`: item is not installed • `TGB_CONFORMITY_LEVEL2`: item is installed, but not enabled • `TGB_CONFORMITY_LEVEL3`: item is installed and enabled, but up to date • `TGB_CONFORMITY_LEVEL4`: compliance level is satisfactory
`ppComment`	Output parameter	Displayed text (XXX stands for firewall or antivirus): • Conformity level is 1: "No XXX is installed on the endpoint." • Conformity level is 2: "XXX is installed on the endpoint, but it is not enabled." • Conformity level is 3: "XXX is enabled on the endpoint, but it isn't up to date." • Conformity level is 4: "XXX: YYY is enabled and up to date on the endpoint." Where possible, YYY identifies the name of the item (Windows Defender, McAfee, Norton, ESET, etc.).

All character strings are encoded in UTF-8 format.

⚠ When the `TGBGetConformityItem` function has been called and it has returned `CONFORMITY_GET_OK`, a subsequent call must be made to the `TGBFreeCommentString` function. Otherwise, the character string memory will not be properly freed.

5.4 Usage example

The following code sample shows how the DLL can be used. It does not include a verification of the DLL's authenticity.

```
#include <stdio.h>
#include <Windows.h>

typedef enum
{
    TGB_CONFORMITY_GET_OK = 0,
    TGB_CONFORMITY_GET_UNKNOWN_ID = 1,
    TGB_CONFORMITY_GET_INVALID_ARGUMENTS = 2,
    TGB_CONFORMITY_GET_INTERNAL_ERROR = 3
} TGBGetConformityItemErrorCode;

typedef enum {
    TGB_FIREWALL_CONFORMITY = 1,
    TGB_ANTIVIRUS_CONFORMITY = 2
} TGBGetConformityItemID;

typedef enum {
    TGB_CONFORMITY_LEVEL1 = 1,
    TGB_CONFORMITY_LEVEL2 = 2,
    TGB_CONFORMITY_LEVEL3 = 3,
    TGB_CONFORMITY_LEVEL4 = 4
} TGBConformityLevel;

typedef TGBGetConformityItemErrorCode(
    CALLBACK* TGBGetConformityItemFn)(
    TGBGetConformityItemID,
        TGBConformityLevel*,
        const char**);
typedef void(CALLBACK* TGBFreeCommentStringFn)(const char*);

static const LPCTSTR dll_path =
    L"C:\\Program Files\\TheGreenBow\\"
    L"TheGreenBow Secure Connection Agent\\tgb_conformity.dll";

int main(void)
{
```

```
printf("Starting\n");

HMODULE hdll = LoadLibrary(dll_path);
if (hdll == NULL)
{
    printf(
        "DLL could not be loaded. error %d\n",
        GetLastError());
    return 1;
}
printf("The DLL is loaded\n");

TGBGetConformityItemFn TGBGetConformityItem =
    (TGBGetConformityItemFn)GetProcAddress(
        hdll,
        "TGBGetConformityItem");

if (TGBGetConformityItem == NULL)
{
    printf(
        "Could not load dll could TGBGetConformityItem "
        "from DLL. error %d\n",
        GetLastError());
    return 2;
}

TGBFreeCommentStringFn TGBFreeCommentString =
    (TGBFreeCommentStringFn)GetProcAddress(
        hdll,
        "TGBFreeCommentString");
if (TGBFreeCommentString == NULL)
{
    printf(
        "Could not load dll could TGBFreeCommentString "
        "from DLL. error %d\n",
        GetLastError());
    return 3;
}
printf("Conformity functions loaded from DLL\n");

TGBConformityLevel level;
const char* comment;

TGBGetConformityItemErrorCode ec =
    TGBGetConformityItem(
        TGB_ANTIVIRUS_CONFORMITY, &level, &comment);
if (ec == TGB_CONFORMITY_GET_OK)
{
    printf("\tAntivirus compliance level: %d\n", level);
    printf("\tComment: %s\n", comment);
    TGBFreeCommentString(comment);
}
else
{
    printf("TGBGetConformityItem returned error %d\n", ec);
}
```

```
    printf("\n");

    ec = TGBGetConformityItem(
        TGB_FIREWALL_CONFORMITY, &level, &comment);
    if (ec == TGB_CONFORMITY_GET_OK)
    {
        printf("\tFirewall compliance level: %d\n", level);
        printf("\tComment: %s\n", comment);
        TGBFreeCommentString(comment);
    }
    else
    {
        printf("TGBGetConformityItem returned error %d\n", ec);
    }

if (hdll)
    {
        printf("Freeing DLL\n");
        FreeLibrary(hdll);
    }
    return 0;
}
```

The following command can be used in a PowerShell to ensure that UTF-8 strings are properly displayed:

```
[Console]::OutputEncoding = [Text.UTF8Encoding]::UTF8
```

6 Contact

6.1 Information

All the information on TheGreenBow products is available on our website: https://thegreenbow.com/.

6.2 Sales

Phone: +33.1.43.12.39.30

E-mail: sales@thegreenbow.com

6.3 Support

There are several pages related to the software's technical support on our website:

Online help

https://www.thegreenbow.com/en/support/online-support/

FAQ

https://www.thegreenbow.com/en/frequently-asked-questions/

Contact form

Technical support can be reached using the form on our website at the following address: https://www.thegreenbow.com/en/support/online-support/technical-support/.

Protect your connections
in any situation

28, rue Caumartin
75009 Paris – France

sales@thegreenbow.com

www.thegreenbow.com