TrueConf

TrueConf Border Controller

Administrator guide

Table of Contents

1. Description

The all-in-one TrueConf Enterprise solution includes the TrueConf Border Controller extension that provides external users (outside the corporate network environment) with secure access to video conferencing servers.

TrueConf Border Controller is a separate extension that acts as a border controller designed to be installed in the DMZ (demilitarized zone) of the corporate network and allowing only secure traffic from TrueConf client applications.

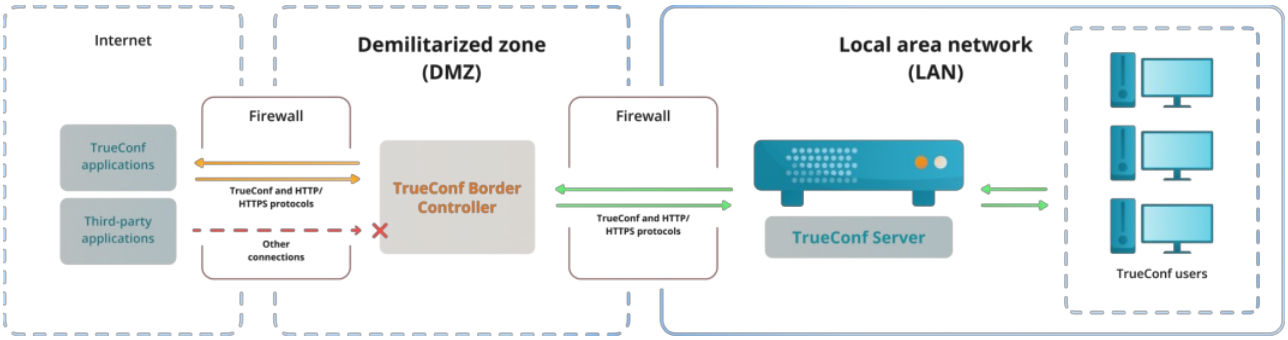
1.1. Parts of the solution

The extension consists of two components that validate traffic via TrueConf and HTTP/HTTPS protocols respectively.

> *i* We recommend using HTTPS on TrueConf Server since it improves the security of the web server resources and ensures the work of the scheduler, real-time meeting management tool, browser-based conference participation and access to a user's personal area.

Components of TrueConf Border Controller have to be configured separately, and are able to work independently of each other which means that it is possible to configure traffic transfer only via TrueConf protocol, but not via HTTPS.

How TrueConf Border Controller works:



1.2. How the solution works

1. The TrueConf Border Controller extension is installed in the DMZ.
2. The extension checks the protocols of the traffic that comes from the external network.
3. If the traffic does not come via TrueConf or HTTPS protocol, it will simply be rejected.
4. If the extension detects traffic from TrueConf client applications or via HTTPS, the connection will be accepted and a new connection will be created from TrueConf Border Controller to the selected TrueConf Server or TrueConf Enterprise instance. When the connection is established, the packets from the application will be sent via the new connection to the video conferencing server (only TrueConf and HTTPS traffic is allowed). This does not only ensure the transfer of media streams, but also ensures the work of the scheduler, access to the video conferencing server web pages, federation and other features.
5. If necessary, the traffic from TrueConf Border Controller to the video conferencing server can be encrypted with the help of multiple symmetric algorithms, including PSK (Pre-Shared Key) .
6. Apart from encryption, the extension does not perform any additional operations on traffic like analysis, saving, transfer to third-party services, and so forth.

So, the protection of a video conferencing server installed inside the corporate network is based on the following principles:

- TrueConf Border Controller does not create a new connection to TrueConf Server until it determines that the packets are coming from TrueConf client application or via the secure HTTPS protocol.
- No external traffic is directed to the video conferencing server by TrueConf Border Controller. This includes the traffic via SIP/H.323/RTP and others. For example, only TrueConf client applications will be able to connect to TrueConf Server from outside the network.
- The IP address of the video conferencing server inside the corporate network is hidden. The server only has to be connected to the DMZ, but it does not have to be connected to the Internet. Please note that server federation will be impossible if there is no connection with the Internet.
- Additionally, it is possible to encrypt the traffic sent via TrueConf protocol.

Every component of the extension is an executable file that does not require installation. It can be run from the console or added as a service on Windows or daemon on Linux.

2. System requirements

We recommend installing TrueConf Border Controller on a physical or virtual server in the DMZ; this server has to meet the following minimum requirements (based on the estimated 800 Mbit/s of transit traffic):

Parameter	Value
Operating system	Dedicated or virtual 64-bit operating system • Microsoft Windows Server 2008 R2/2012/2016/2019/2022 (including the Core edition) with latest updates installed • Debian 10 / 11 / 12 • CentOS Stream 9
CPU	Any CPU with at least four physical cores
RAM	4 GB
Available disk space	Disk space available for saving log files generated by the extension (if the extension is activated)

Contact our technical support to learn more about the exact requirements depending on the number of each TrueConf Border Controller component instances running in parallel on a single machine and the expected volume of traffic.

Next, we will show how to configure the launch of these components on Windows and Linux-based distros.

If you have any questions regarding the configuration of TrueConf Border Controller, please contact our technical support.

3. The component for the TrueConf protocol

It is provided as an installer for Windows and all supported Linux distros. The component settings are specified in the configuration file `tc_bc.cfg` which is created automatically when the component is installed. An example of the configuration file is presented after the list of parameters.

After the component is installed, the corresponding service will be automatically added on the OS:

- On Windows, it will be named **TrueConf Border Controller** and will have the id **tc_bc**, the path to the executable file will be `C:\Program Files\TrueConf\Border Controller\tc_bc.exe`.
- On Linux, it will be identified as **trueconf-bc**, the path to the executable file will be `/opt/trueconf/border-controller/bin/tc_bc`.

3.1. List of parameters

During component installation the configuration file, where parameters can be specified, will be created:

- on Windows: `C:\Program Files\TrueConf\Border Controller\etc\tc_bc.cfg`
- on Linux: `/opt/trueconf/border-controller/etc/tc_bc.cfg`

The component supports the following parameters (in brackets, you can find the aliases for some of these parameters).

3.1.1. General parameters

- `--Debug <level>` — the logging level from **0** (disabled) to **4**
- `--LogDirectory <path>` — the path for saving log files related to the extension
- `--LogToConsole` — log messages are displayed in the console instead of being written to log files
- `--Daemonize <path to the PID lock-file>` **(only for Linux)** — start as a daemon with the path for saving the PID file
- `--Service` **(only for Windows)** — start as a service
- `--R` — automatic service restart in case of an error

3.1.2. Routing parameters

- `-D <id>/<host>:<port>` (`--Destination <id>/<host>:<port>`) — the address or FQDN of TrueConf Server or TrueConf Enterprise where traffic will be redirected. Here:
  - `<id>` — (optional) unique string of the identifier used for combining options (when the same TrueConf Border Controller should follow several redirection rules, we **do not recommend this approach**)
  - `<host>` — IPv4, IPv6 or FQDN (IPv6 has to be specified in square brackets `[IPv6]`)
  - `<port>` — (optional) port, this parameter can be omitted if it is equal to the default value **4307**
- `-L <id>/<host>:<port>` (`--Listen <id>/<host>:<port>`) — the network interface for receiving the incoming traffic; the options of this parameter match the options for the `-D` parameter
- `-E <id>/<cipher>:<flags>:<key>` (`--Encryption <id>/<cipher>:<flags>:<key>`) — encryption of packets sent from TrueConf Border Controller to the video conferencing server. Here:
  - `<id>` — (optional) unique string of the identifier for combining options
  - `<cipher>` — selected encryption method, it can have such values as `None` (no encryption, default), `ChaCha20` , `AES-256-CTR` , `AES-256-OFB` , `AES-192-CTR` , `AES-192-OFB` , `AES-128-CTR` , `AES-128-OFB` , `xoshiro256++` , `xoshiro256**`

- `<key>` — encryption key (in the hexadecimal format) may be omitted if a randomly generated value is used (incompatible with PSK mode)
- `<flags>` — if this parameter is available and is equal to `PSK`, Pre-Shared Key encryption will be used. In this case some additional configuration will be needed on the side of the video conferencing server.

3.1.3. Command-line parameters for starting the component from the terminal (console)

You can launch the executable file of the component from the terminal with some parameters that cannot be used in the configuration file:

- `-h` (`--help`) — the display of the built-in help menu with the list of parameters and examples
- `-c <path>` (`--ConfigFile <path>`) — the path `<path>` to the configuration file
- `-v` (`--version`) — the component version.

For example, to open the help section on Linux, use:

```sh
sudo /opt/trueconf/border-controller/bin/tc_bc -h
```

3.1.4. An example of a configuration file

```
LogDirectory=/opt/trueconf/border-controller/var/log
Listen=10.140.10.123
Destination=10.110.10.10
Encryption=ChaCha20
```

3.2. Launching the component

When the configuration file is ready, you can launch the component.

3.2.1. For Windows

To control the services on Windows, one can use either GUI or the command line (terminal).

To quickly open the services management window, start the command line (terminal) or PowerShell and run the command `services.msc`. In the opened window, you can select the service **TrueConf Border Controller** from the list and start it. In addition, you can choose if this service should be launched automatically when the OS is started.

To fully control services from the terminal, you can use the tool `sc.exe`. All the commands should be run on behalf of the OS administrator. For example, to start the service, execute this command:

```sh
sc start tc_bc
```

To add the service to the automatic startup, run:

```sh
sc config tc_bc start=auto
```

3.2.2. For Linux

To manage the services (called *daemons* within the context of Linux), one should use the `systemctl` tool.

To start the daemon **trueconf-bc**, run:

```sh
sudo systemctl start trueconf-bc
```

To make sure that the **trueconf-bc** daemon is launched automatically when the OS is started, run:

```sh
sudo systemctl enable trueconf-bc
```


4. HTTPS component

It is provided as an installer for Windows and all supported Linux distros. The component settings are specified in the configuration file `webproxy.toml` as it is shown below.

After the component is installed, the corresponding service will be automatically added on the OS:

- On Windows, it will be named **TrueConf Border Controller htpps** and will have the id **tc_bchttps**, the path to the executable file will be `C:\Program Files\TrueConf\Border Controller\tc_bchttps.exe`.
- On Linux, it will be identified as **trueconf-bchttps**, the path to the executable file will be `/opt/trueconf/border-controller/bin/tc_bchttps`.

The launch of this component is configured similarly to the component for handling the TrueConf traffic. However, there are certain differences:

- You need to configure the certificate in advance.
- The operation parameters are defined in the configuration file **webproxy.toml**.

4.1. Configuration of certificates

1. If a self-signed certificate is configured on TrueConf Server, download it via the link **Download ca.crt** in the **Self-signed certificate** section and add it to the trusted root certificates on the machine with TrueConf Border Controller. Check the documentation for your OS to learn how it can be done.

For example, **on Debian**:

- Copy the certificate file to the certificate storage in the directory `usr/local/share/ca-certificates/`:

```sh
sudo cp ca.crt /usr/local/share/ca-certificates/
```

- Update the certificate storage with this command:

```sh
sudo update-ca-certificates -v
```

* If there is an error message indicating that the command was not found, install its package from the repository:

  ```sh
  sudo apt install -y ca-certificates
  ```

- To check if your OS trusts the certificate, run this command:

```sh
openssl verify /usr/local/share/ca-certificates/ca.crt
```

2. After copying certificate files on Linux, make sure that these files are owned by `trueconf` (otherwise, the TrueConf Border Controller service will not start correctly). To check the status, run this command:

```sh
ls -l /usr/local/share/ca-certificates/ca.crt
```

The terminal should display `trueconf trueconf` in columns 2 and 3. If this is not the case, execute this command:

```sh
sudo chown trueconf:trueconf /usr/local/share/ca-certificates/ca.crt
```

3. In the TrueConf Server control panel, go to the **Web →Settings** section and specify the address of the

machine with TrueConf Border Controller in the **External address of TrueConf Server web** field.

4. Create a certificate for the machine with TrueConf Border Controller. If you do not have a commercial certificate, you can create a self-signed certificate as it is described in our knowledge base.

5. Copy the certificate and key obtained at step 3 to the directory `<path_to_border_controller>\etc\crt\` where `<path_to_border_controller>` is the path to the executable file of the component on your OS.

6. Rename the certificate and key files as `<guid>.crt` and `<guid>.key` where `<guid>` is a 128-bit GUID identifier which will be the same for both files. It can be generated with the help of the online service UUID Generator .

4.2. Configuration file settings

The configuration file `webproxy.toml` will be created during component installation:

- on Windows: `C:\Program Files\TrueConf\Border Controller\etc\webproxy.toml`
- on Linux: `/opt/trueconf/border-controller/etc/webproxy.toml`

By default, the configuration file contains the following lines:

```
[certificate]
cert_extension = '.crt'
key_extension = '.key'

[dir]
executable_relative = false
installation = '/opt/trueconf/border-controller'

[file]
configname = 'webproxy'

[interfaces]
[interfaces.list]
[interfaces.list.0]
Address = '[::]:80'
EnableTLS = false
ReadTimeout = 0
TLSConfigID = ''
TargetID = ''

[proxy]
trust_client_headers = false

[targets]
[targets.list]

[tls]
[tls.list]
```

Specify the following values to configure the component for the HTTPS protocol:

- in the `[dir]` section:
  - `installation` — the path to the executable file of the component
- in the section `[interfaces.list.0]`:
  - `Address` — HTTPS port if it is different from the standard **443**

- `TLSConfigID` — the name of the certificate and key files received at step 5
- `TargetID` — GUID for identifying a block of HTTPS settings from the `[targets]` section

- in the section `[interfaces.list.1]`:
  - `Address` — the port for accessing the control panel via HTTP if the port is different from the standard **80** port
  - `TargetID` — GUID for identifying a block of HTTP settings from the `[targets]` section
- for each `[targets.list.<guid>]` blocks in the `[targets]` section:
  - generate unique GUIDs and add them instead of `<guid>`
  - `address` — IP address or FQDN of TrueConf Server and the port for the transfer of traffic from the component
  - `is_secure` — the value is equal to `true` if an HTTPS port was specified for the `address` parameter of the current `[targets.list.<guid>]` block ; otherwise it is equal to `false`
- in the `[tls]` section:
  - for the `[tls.list.<guid>]` block name, replace `<guid>` with the `TLSConfigID` value (it is also the name of the certificate file from step 5)
  - `CertificateID` and `ID` — value of `TLSConfigID` .

7. Save the file `webproxy.toml` and run the component.

4.3. Launching the component on Windows

Similarly to the component for the TrueConf protocol, the service on Windows can be started from the `services.msc` tool or from the terminal with the help of the `sc.exe` utility. For example:

```sh
sc start tc_bchttps
```

The component can be added to the automatic startup in a similar way, for example:

```sh
sc config tc_bchttps start=auto
```

4.4. Launching the component on Linux

To control the component, use the utility `systemctl` as it was described for **trueconf-bc**. For example, to start the component, run this command:

```sh
sudo systemctl start trueconf-bchttps
```

