

E M S C B – Milestone No. II
Turaya.VPN – Secure Virtual Private Network

REQUIREMENTS & ARCHITECTURE

based on
European Multilaterally Secure Computing Base (EMSCB)



Abstract: Turaya-VPN is based upon the EMSCB security platform and describes the possibility to establish a transparent and secure Virtual Private Network. The secret used for setup and communication is protected by the EMSCB security platform.

Version July 5, 2007

Contents

| | | |
|----------|-------------------------------------|----------|
| 1 | Requirement Specification | 2 |
| 1.1 | Introduction | 2 |
| 1.2 | Security Environment | 2 |
| 1.3 | Use Case Model | 2 |
| | /UC 10/ Create Connection | 3 |
| | /UC 20/ Receive Data | 4 |
| | /UC 30/ Delete Connection | 4 |
| 2 | Architecture | 6 |

1 Requirement Specification

1.1 Introduction

Turaya-VPN shall establish a Virtual Private Network (VPN) by using the EMSCB security platform and therefore facilitate protected communication with servers (web servers, email servers, news servers etc.) in a protected intranet.

Turaya-VPN ensures that a user within an unsecure network gets secure access to a protected area. The communication between the user and the secure area is protected by the EMSCB security platform.

The secret required for establishing the connection is managed by EMSCB. The user operating system has no access to the information.

1.2 Security Environment

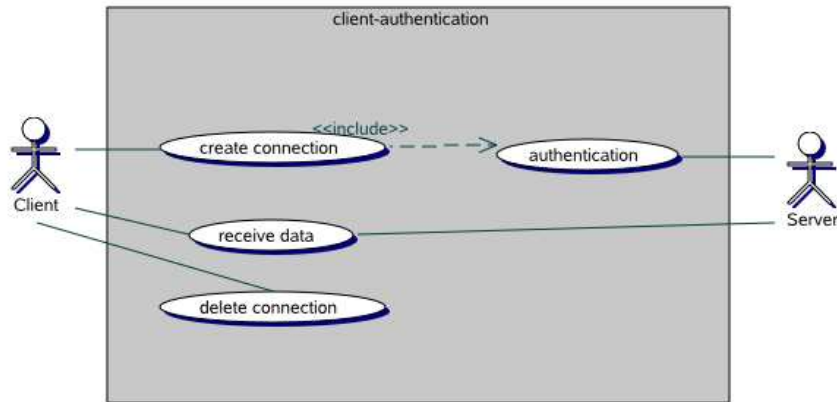
The EMSCB security platform ensures that the secret required for establishing the connection is kept in safe custody. The user operating system gets no access to the secret. This ensures that compromising the user operating system does not endanger the security of the whole VPN, because the secret is not within reach of the user operating system.

1.3 Use Case Model

The use case “Client Authentication” describes how a client process (e.g. a browser) from the user system accesses a server (e.g. a web server) located in a protected intranet. To obtain access to the server’s service it is necessary that the SVPN system establishes a protected channel to the corresponding intranet gateway server on request of the user system. Subsequently, the client process is able to exchange confidential data with the server.

At this, the user system has to authenticate itself at the server. The authentication is carried out by a secret protected by the SVPN system. Every user system request is proceeded through the EMSCB security platform to the SVPN system, gets encrypted there and is transmitted to the gateway of the protected intranet. Additionally to the encryption, the integrity of the data transmitted from the gateway of the protected intranet is also ensured.

Consequently, two communication channels with different security properties exist. The communication channel between the user system and the SVPN system is managed and protected by the EMSCB security platform. The communication channel between the SVPN system and the gateway of the protected intranet obtains its security by an encrypted connection. The secret required for establishing the connection is protected by the SVPN system and not accessible by the user system. The encryption is transparent for the user system.



/UC 10/ Create Connection

Description A client process running on the user system wants to establish a connection to a server in a protected intranet.

Actors Server, client

Rationale The server demands an authentication (pre-shared key, certificate, ...) to approve a connection establishment. While establishing the connection, a session key, which is used to encrypt the subsequent communication, is transmitted securely.

Includes Authentication

Preconditions The client requests data from a server.

Postconditions Once a connection is established, the client is able to access all data provided for it.

Normal Flow

1. The client sends a request for connection establishment to the server.
2. The server demands the client's authentication.
3. The client authentication is successful.
4. The server approves the connection establishment of the client.

Alternative Flow

1. The client sends a request for connection establishment to the server.
2. The server demands the client's authentication.
3. The client authentication fails.
4. The connection establishment is aborted.

/UC 20/ Receive Data

Description A client of the user system requests data from the server.

Actors Server, client

Rationale The client (e.g. browser) obtains data to be displayed from the server. The server requires an authentication from the client, because the server possesses sensitive data which are to be requested by specific clients only. The authentication is carried out by means of a secret bound to the TPM of the respective client system.

Includes u20

Preconditions A connection to the server has been established. The client possesses a TPM, to which a valid secret, which is accepted by the server, is bound to.

Postconditions The data is transmitted completely. The connection remains established.

Normal Flow

1. The client requests data from the server.
2. The server requires an authentication.
3. The client's authentication at the server is successful.
4. The data exchange between client and server begins.

Alternative Flow

1. The client requests data from the server.
2. The server requires an authentication.
3. The client's authentication at the server fails.
4. The procedure "receive data" is aborted.

/UC 30/ Delete Connection

Description The data exchange between client and server is finished. The connection between client and server can be terminated.

Actors Client, server

Rationale After transmission, the connection has to be terminated.

Includes

Preconditions There exists an active connection between client and server. All data have been transferred between client and server.

Postconditions The connection has been terminated and the secret is reset where appropriate.

Normal Flow

1. All data have been transmitted from the client to the server.
2. The client sends an acknowledgement that all data have been received and that the connection can be terminated.

3. The server approves the connection termination and terminates the connection.
4. The connection is terminated.

Alternative Flow

1. All data have been transmitted from the client to the server.
2. The client sends an acknowledgement that all data have been received and that the connection can be terminated.
3. The server updates the secret stored inside the client by using the still existing secure connection.
4. The server approves the connection termination and terminates the connection.
5. The connection is terminated.

2 Architecture

In the context of the EMSCB environment, a certificate-based VPN shall be implemented. The following requirements to the communication channel are made:

- Integrity of the transmitted data
- Confidentiality of the transmitted data

These requirements implicate that all systems concerned have to attest to each other. The attestation is due to the properties of a system. The properties of a system change as components are added, removed or modified. The host system which is operated by the user of Turaya-VPN is based on the EMSCB security platform. All security aspects are also regarded within Turaya-VPN and are part of the security of the complete system.

In the following we will distinguish between the user system (e.g. Linux) and the Turaya-VPN system. Both work on a host system and are managed by the EMSCB security platform. The user system is the user operating system the user interacts with. Every input (keyboard, mouse, etc.) is processed by the user system. The Turaya-VPN system is executed in parallel through the EMSCB security platform on the same host system. Every network request of the user system is forwarded to the Turaya-VPN system by EMSCB. There, the requests are classified and processed according to their security property. This can mean that the Turaya-VPN system establishes a secure connection to a protected intranet gateway and the data transmitted are encrypted.

