Gateway 6.17.3 Security Guide

Saved as PDF June 17, 2026

Axway Doc Portal

Contents

Gateway Security guide 1

Axway Gateway (formerly XFB Gateway) is an Axway product in the Managed File Transfer category, that serves as a communication gateway. It provides a secure front end for handling data exchange between partners that are connected via public networks such as the Internet.

Security

This guide provides instructions and recommendations to help you strengthen the security of Axway Gateway. Security descriptions include:

- How the product was developed in a secure way
- A list of main security features
- Secure configuration parameters, including the Secure by Default configuration
- Identity and Access Management in this product
- Best practices to use this product in a secure way

This document is targeted to the following audiences:

- Security teams in charge of auditing the security of the product
- Global network engineers
- Product administrators

Get more help

Tip: online documentation lists

To find available documents for most Axway products and versions:

1. Go to https://docs.axway.com/bundle.
2. In the left pane Filters list, select your product or product version.

Note Customers with active support contracts can log in to access restricted content.

Full documentation set

Here's the full list of the Gateway documentation set (including this guide):

- Axway Gateway *User guide*
  The main guide for Axway Gateway. As the users of this software mostly have an administrator profile, this guide serves as an Administration guide
- Axway Gateway *Release Notes*
- Axway Gateway *Installation Guide*
  Provides information about installation and prerequisites

- Axway Gateway *Configuration Guide (UNIX)*
  Provides information specific to different variants of the UNIX Operating System
- Axway Gateway *Upgrade guide*
  This guide helps to perform the migration between towards Gateway 6.17.3
- Axway Gateway *Security guide* (requires login) (this guide)
  This guide provides instructions and recommendations to help you strengthen the security of Axway Gateway. To protect your data from possible hackers, access to this guide requires an Axway login.
- Axway Gateway *Interoperability Guide*
  Provides information relevant to interoperability with other products, such as Composer,Gateway Navigator, Integrator

Documentation of other related products

- Axway Secure Relay Master Agent Administrator Guide
- Axway Secure Relay Router Agent Administrator Guide

Gateway documentation set

For more information about supported platforms:

- Axway Supported Platforms

Training services

Axway offers training across the globe, including on-site instructor-led classes and self-paced online learning. For details, go to: https://www.axway.com/en/services/training-certification

Support services

The Axway Global Support team provides worldwide 24 x 7 support for customers with active support agreements.

Email support@axway.com or visit Axway Support at https://support.axway.com.

Document version: 19 June 2023

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Introduction 2

This guide provides instructions and recommendations to help you strengthen the security of Axway Gateway. Security descriptions include:

- How the product was developed in a secure way
- A list of main security features
- Secure configuration parameters, including the Secure by Default configuration
- Identity and Access Management in this product
- Best practices to use this product in a secure way

This document is targeted to the following audiences:

- Security teams in charge of auditing the security of the product
- Global network engineers
- Product administrators

Links to documentation set for Axway Gateway 6.17.3:

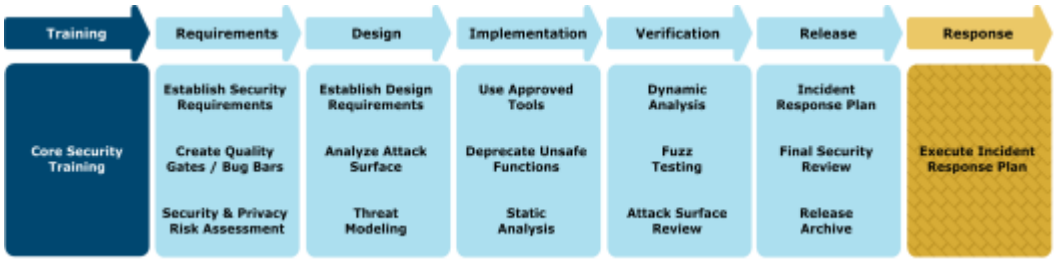
- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Secure Development Lifecycle

3

Axway implements a Secure Development Lifecycle (SSDLC) in the development of its main products. This SSDLC is similar to the one defined by Microsoft. A dedicated team, the SSG (Software Security Group), in association with the product development teams, is in charge of managing this SSDLC.

The secure development lifecycle consists of standardized process flows during the development lifecycle. The SSDLC process is a spiral development model. Axway works within this model and with an Agile development model to deliver the required security artifacts for the SSDLC. The following depicts the simplified Microsoft SSDLC model.



Axway implements many SSDLC process and control points. Some important steps along the path to secure product delivery include the creation of both security and design requirements as well as the associated design-time threat models. These steps ensure that products meet initial specifications and that the posited security designs pass all the threat model criteria established both within the security community and with the Axway PSG.

At Axway, there are a broad suite of tools associated with the implementation and verification phases of the SSDLC. Both static and dynamic analysis tools are run to identify potential code weaknesses and to discover security issues potentially exposed at runtime. Suites of attack surface tools and penetration testing tools are run to ensure the deployed products on the target platforms meet gated criteria in the SSDLC and criteria provided by the internal PSG. Axway provides tool suite information and our customized usage profiles upon customer request. Well-known security industry-standard tools are used as well as many other enhanced test scenarios required by our most security-conscious customers.

The new product introduction (NPI) process at Axway requires a final security review with associated development and testing artifacts. This NPI process supports the release of new products or major product revisions. The NPI also ensures the SSDLC is started early in development to optimize the delivery of secured products for you, our customer.

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Certifications and compliance

4

Compliance

FIPS 140-2

The current version of this product (Gateway 6.17.3) is FIPS 140-2 level 1 compliant. This compliance is valid on the following platforms:

- Linux-x86-32
- Linux-x86-64
- Sun-sparc-32
- Aix-power-32
- Aix-power-64
- Hp-IA-64

About FIPS

Federal Information Processing Standards (FIPS) are standards and guidelines that the National Institute of Standards and Technology (NIST) issues for use in United States federal government computer systems.

The Federal Information Processing Standard 140–2 (FIPS 140–2) defines security requirements for cryptographic modules that are used within a security system protecting sensitive but unclassified information, in computer and telecommunication systems. FIPS 140-2 requires organizations that do business with a US Government agency or department that requires the exchange of sensitive information, to ensure that they meet the FIPS 140-2 security standards.

The standard provides four increasing, qualitative levels of security intended to cover a wide range of potential applications and environments. The security requirements cover areas related to the secure design and implementation of a cryptographic module. These areas include:

- cryptographic module specification,
- cryptographic module ports and interfaces,
- roles, services, and authentication,
- finite state model;
- physical security;
- operational environment;
- cryptographic key management;
- electromagnetic interference/electromagnetic compatibility (EMI/EMC);
- self-tests;

- design assurance;
- mitigation of other attacks.

For more information, read SECURITY REQUIREMENTS FOR CRYPTOGRAPHIC MODULES (a publication of Information Technology Laboratory - National Institute of Standards and Technology)

Gateway FIPS compliance

Gateway falls under the “Cryptographic key management” area of the FIPS 140-2 standard. As such, when in FIPS mode, Gateway applies restrictions for random number and key generation, key establishment, key distribution, key entry/output, key storage, and key zeroization.

Gateway Server uses the OpenSSL library for encryption. When FIPS mode is enabled, the OpenSSL library is used in FIPS mode.

The Secure Relay Master Agent module included in Gateway Server uses the Entrust FIPS-compliant cryptography provider.

When FIPS mode is enabled, any encryption operation performed within Gateway Server (of either transferred or persisted data) is compliant with the FIPS 140-2 standard.

To ensure this compliance, some restrictions apply in the way Gateway security-related objects are configured, as well as in the execution of the data transfers.

For more information about Gateway and FIPS, refer to the *Gateway User Guide*.

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Security features 5

This product contains the following security features:

- Secure connections
- Protocol level security
- Transport level security
- Host system security
- Integration in secured environments
- DMZ deployment
- Password management
- Certificate Management
- Identity and Access Management

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Secure connections

Secure connections

The following secure connections are available:

- Connections between the **UI** (Gateway Navigator) and the **server** can be secured, but it is not by default. This is accomplished using SSL/TLS. The choice is made on server side and the client must connect in a secure way if the server is so configured.
- Connections to **other Axway products** can be SSL\TLS secured:
    - Connection to **Passport AM** can be secured using mutual authenticated TLS. The Passport server CA is delivered in Gateway installation. The certificate used to secure these connections is created and stored at Gateway startup (once) by sending to PassPort a CSR with the certificate generated by the product. When the generated certificate expires, it is renewed. Connection is not secured by default.
    - Connection to **Passport PM** and **PS** can be secured using server side authenticated TLS. Connection is not secured by default.
    - Connection between **SecureRelay MA** and SecureRelay **RA** is always secured
    - Connection to **Swift infrastructure** can be secured from the Swift API (RA) configuration
- Connections with **partners** can be secured using SSL\TLS over the different clear protocols (HTTP, FTP,

EDIINT, OFTP, SMTP/POP3, PeSIT etc) or using secure transport protocols such as SFTP. Except SFTP, connections with partners are not secured by default.

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Protocol level security

With Gateway, you can analyze the identity of Transfer partners during protocol connection to determine whether or not you want to authorize them to connect and exchange files. You have the option of using internal or external processes for this identity control.

Possible connection control methods that can be used in Gateway are:

- CGate – internal processing; can be configured through user exit or directly in Gateway
- SGate – internal processing configured only through user exit
- Passport PM – external processing

Internal processing: Using the CGate/SGate object

You can use the CGate/SGate object to link client access rights to connection authorization. You can define:

- File visibility for groups or sub-groups of users
- Access to portions of the VFD (Virtual File Directory) tree
- File transfer rights for individual users
- Filtering based on the port number and IP address of the caller

Note that, after passing the CGate/SGate control, the user can be authenticated a second time at remote site level (only available for physical remote sites)

CGate objects can be forced (selected) from user exit, using an external user access control database.

SGate objects are always forced (selected) from user exits.

External processing: Using Passport PM

Connection control is made using partner information stored in Passport database.

OFTP protocol

Additionally, the OFTP v2 protocol offers a protocol specific authentication using CMS signature/encryption

For more information refer to Security features.

HTTP protocol

HTTP server identification methods *CGI session id* and *CGI user password* are insecure by design and should not be used.

Gateway offers several other identification methods, which should be used instead:

- Web
- Cookie session id
- Cookie user password

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Transport level security

At the transport level, Gateway provides support for:

Protocol and network translation

SSL (Secure Socket Layer) / TLS (Transport Layer Security) on incoming and outgoing connections

SSL/TLS protocol

Most protocols supported in Gateway can be secured using SSL/TLS, but connections are not secured by default. Protocols that can be secured are:

- HTTP → HTTPS
- FTP → FTPS (implicit or explicit)
- EDIINT protocols: AS1, AS2, AS3
- RosettaNet
- OFTP
- PeSIT D and PeSIT E
- PEL
- SMTP
- POP3

To add SSL/TLS for incoming connections, a server SSL/TLS profile object must be created and attached to the network profile corresponding to the protocol listen point. In the security profile object you can specify

- The client authentication type
- The version of TLS to use
- The accepted cipher suites – Gateway currently supports only RSA and ECDSA based cipher suites, but the list can be extended in combination with Secure Relay and security termination

Note: several cipher suites are marked as “weak”, and cannot be used unless the configuration parameter [tls]enable_weak_ciphers is set to yes. These cipher suites could be removed in a future version. We recommend against using weak cipher suites.
For more information regarding the accepted cipher suites, refer to *TLS cipher suites*

- Accepted authorities
- Certificate for server side authentication (local PKI or PassportPS)
- Different options for TLS protocol: session cache, key renegotiation, close notify exchange etc

To add SSL/TLS for outgoing connections, a client SSL/TLS profile object must be created and attached to the remote site used for connecting to partners

SSH/SFTP protocol

Authentication: The claimed identity of the client user is checked via a private/public key pair or user/password.

Supported key algorithms: DSA, RSA, ECDSA (NISTP256, NISTP384, NISTP521), x.509v3 certificate (RSA keys only)

Supported key exchange algorithms: Diffie-Hellman key exchange with SHA-1 and Oakley Group 2 (diffie-hellman-group1-sha1), Diffie-Hellman Group and Key Exchange with SHA-1 (diffie-hellman-group-exchange-sha1), Diffie-Hellman Group and Key Exchange with SHA-256 (diffie-hellman-group-exchange-sha256), Diffie-Hellman key exchange with SHA-1 as HASH and Oakley Group 14 (diffie-hellman-group14-sha1), and ephemeral-ephemeral elliptic curve Diffie-Hellman on nistp256 with SHA-256 (ecdh-sha2-nistp256), nistp384 with SHA-384 (ecdh-sha2-nistp384) and nistp521 with SHA-512 (ecdh-sha2-nistp521). **Note**: Gateway does not support Kerberos or GSSAPI authentication in either SFTP server or client mode.

Supported encryption algorithms: AES128_CBC, 3DES_CBC, AES256_CBC, AES128_CTR, AES256_CTR

Supported MAC algorithms: SHA1, MD5, SHA1_96, MD5_96, SHA256

Security features:

- Integrity: Adds a message abstract (also referred to as a hash code or message digest) to the data to ensure that it arrives at its destination intact. In SSH the hash code, referred to as MAC (Message Authentication Code), is added to each message sent using the negotiated MAC algorithm. Refer to Certificates and keys.
- Confidentiality (also referred to as Privacy): Prevents third parties from accessing data that is not intended for them. Typically, data is protected using either symmetrical or asymmetrical encryption techniques. Refer to Axway Gateway security features.
- Authentication: The claimed identity of the client user is checked via a private/public key pair or user/password.
- For more details see User guide > *SFTP*

For detailed information please refer to Security features

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Host system security

Host system security

Gateway provides complete protection for the host system via:

- **Virtual File Directory** - the VFD has two operating modes that you can run separately or simultaneously:
    - **VFD** (Virtual File Directory) mode: The VFD is a logical representation of directories and files (VFD Files). The logical files in the VFD represent a duplication of a given Transfer Request extract in the Gateway Mailbox. From the Gateway client side (FTP, SFTP or HTTP clients), you can view the logical representation of these files, but cannot directly access them.
    - **RFD** (Real File Directory) mode: You can use a mount mechanism to link the logical file system (VFD) to the physical file system (RFD) on the Gateway host machine. The files seen from the client side are then real files that are located on the Gateway host machine. These files are optionally associated with a Gateway Mailbox Transfer.

  Files in RFDs must be protected by the host administrator using proper access rights at the operating system level.

  Partner access rights to VFD and RFD files and directories are configured in Gateway via CGate/CGateGroup and SGate/SGateGroup objects.

- **Database encryption**

  You can use AES (Advanced Encryption Standard) encryption mechanisms to store Gateway administration objects and/or the Mailbox file securely on the host machine. Only users with specific rights can read, modify, or delete files. This encryption mechanism prevents external attacks. No one can access data that could enable them to bypass the Gateway authentication process.

- **Temporary file encryption**

  To route files, Gateway must make a local copy of the files before routing them. Gateway uses temporary file encryption to protect the files it temporarily stores from malicious attacks. This is particularly important when you deploy Gateway in a DMZ. In a DMZ deployment, you should take the additional precaution of encrypting the Gateway Mailbox.

  The key used for temporary file encryption is an AES key generated on the spot for each file received. The generated key is attached to the transfer entry in the mailbox and it is recovered when we need to recover the file. The mailbox can be also AES128 encrypted (but it is not by default) by activating the mailbox database encryption option.

  When you route files using Gateway and temporary file encryption, you must link the input transfer to the output transfer using the combination of a Decision Rule (to identify the incoming transfer) and an applied Model (to specify the outgoing transfer characteristics).

  This enables Gateway to identify the security key necessary to read the deposited ciphered file in order to process it for the subsequent transfer.

- **User Access Control of Gateway objects**

User Access Control applies to all methods of accessing Gateway objects: via the Gateway GUI, command lines, APIs or the Axway MFT Navigator.

The User and Profile objects in Gateway enable you to define user identification and access rights. Typically only Administrators have access to these objects. The User object defines properties for each user and the Profile object defines the type of operations a set of users can perform on each Gateway object.

For detailed implementation information please refer to *Security features*

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Integration in secured environments

Integration in secured environments

Gateway is fully compatible with SOCKS4, FTP and HTTP proxies.

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

DMZ deployment

You can further enhance the isolation and security of your enterprise network by deploying Gateway in a DMZ architecture.

Secure Relay is a secure, multi-protocol, front-end product that prevents external partners from directly accessing Gateway for file exchange. Secure Relay processes data streaming of external connections from the outside world to Gateway without any data storage (even temporary) in the DMZ. Secure Relay is compatible with any Gateway supported protocol that uses a TCP/IP network.

When you use Gateway with Secure Relay, you derive the following specific benefits:

- All of your data is stored in your local network, protected by your firewall
- Connections from outside the protected area are refused
- Configuration data, including private server keys, are never stored in the DMZ
- No decryption occurs within the DMZ
- You can adapt the connection parameters of your Gateway to conform to the policies you establish in the DMZ

Distribution of functions

Native Gateway functions work together with the specific Secure Relay functions to provide DMZ security.

In a Secure Relay / DMZ configuration:

- Gateway:
    - Manages streaming connections with the Secure Relay Router Agent situated in the DMZ
    - Manages user access to file resources (workspaces, downloads and uploads)
    - Handles the file transfer protocol dialog with partners
    - Manages file transfer scheduling and monitoring
    - Can terminate transport security communication (either SSL or SSH for SFTP) to ensure a transparent link between partner identities (X.509 certificate) and file operations
    - Generates links with internal IT system applications by either routing files to other destinations or by direct integration
    - Handles physical file management operations
- Secure Relay:
    - Manages connections with the outside world for file exchange, depending on the Gateway directives
    - Handles data streaming of flows to and from the Internet
    - Can terminate transport security communication (either SSL or SSH for SFTP) to ensure a transparent link between partner identities (X.509 certificate) and file operations

Secure Relay architecture

Axway Supported Platforms Guide is divided into two basic components parts:

- Secure Relay Master Agent
  The Secure Relay Master Agent is a component that resides with and is fully integrated in the Gateway product. All of the processes that support Secure Relay functions are automatically included in Gateway on installation.
- Secure Relay Router Agent
  The Secure Relay Router Agent is a stand-alone component. You install the Secure Relay Router Agent on a machine located in the DMZ, independently of Gateway.

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Password management

The following password features are available:

Passwords stored in CGate/SGate objects can be stored hashed in the database. To activate password hashing, configuration parameters [monitor]disable_password_regex='yes' and [monitor]cgate_hashed_password='yes'

should be used.

Passwords to access certificates and keys when importing them in Gateway's local PKI are used only at import time and they are not stored.

All objects databases can be encrypted using AES128 (see database encryption).

All passwords used to access other products are stored encrypted on the disk using PKCS5 standard. These passwords are:

- Passwords to client certificates used for securing connections between products (PassPort AM, Secure Relay, Navigator to Gateway server)
- Passwords for authenticating Gateway as client in other products (PassPort PM, PassPort PS)
- Password for authenticating remote command line clients to Gateway server

For each of these passwords, three files are stored on disk and path to these files is configured in Gateway:

- A salt file
- A derived key file
- An encrypted password file

Access to these files should be restricted at operating system level only to those users allowed to manage Gateway passwords.

When access management is enabled, the same mechanism is used to encrypt commands (command line utility, exits, scripts). The administration user, decryption key file pathname and encrypted password file pathname are stored in the p_cs_username, p_cs_dk_file and p_cs_encpass_file environment variables, respectively; access to the decryption key file and encrypted password file should be restricted at operating system level.

The encrypted password files are created with a Gateway tool named `pelencpass`. For more information about creating encrypted password files, refer to Password Management .

For partner management, PassPort PM can be used in combination with Gateway. In this case, partner passwords and server side passwords (passwords to check) will be stored in the PassPort database. Passwords will be provided to Gateway at request together with the rest of the partner information (connection to PassPort PM can be TLS secured).

For access management, PassPort AM can be used. In PassPort AM we can define password policies to meet the company's requirements for security. User passwords are stored in the PassPort AM database. At login time, user login and password are sent to PassPort AM for authentication. It is recommended that the connection to PassPort AM be TLS secured.

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Certificate Management

Certificate Management

Certificate management can either be performed internally in the product or using PassPort PS.

PassPort PS

The following features are available:

- The product can be connected to PassPort PS to fully manage certificates. When using PassPort, a full validation of the certificates is done (CRL or OSCP server, full chain validation).
- Local private keys are stored in the database AES 256 encrypted. The key used to perform this encryption uses a key derived from the password provided at installation time. This password can be changed at any time using the administration UI.
- The local private keys can also be stored in an HSM (nCipher nShield, WebSentry).
- CSR can be imported or generated to create a certificate in association with an external CA.

See the PassPort documentation for a complete list of features in this product.

When not using PassPort

The following features are available / not available:

- The revocation list are not checked
  Reason: that feature is not available
- Certificate path are checked.
- Keys and certificates are stored in a local file database. The database can be encrypted using AES128 by activating object database encryption. The key to encrypt is generated based on a passphrase given at encryption time. The passphrase can be changed at any time from Gateway configuration tools (command line).
- Certificates with RSA keys up to 8192 bits length can be imported in Gateway (PEM and DER formats are supported).
- RSA keys up to 4096 bits length can be imported in Gateway (PEM and DER formats are supported)
- DSS(DSA) keys up to 1024 bits length can be imported in Gateway (PEM/DER)
- X.509 Certificates can be imported in Gateway (PEM, DER, P12 – single cert or chain). X.509 certificates with RSA or ECDSA keys are supported.
- At import time, each key or certificate are given an internal name (an alias). When referring to keys or certificates in Gateway store, the Secs alias object type must be used.

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Identity and Access Management

Identity and Access Management

Identity and Access Management can either be performed internally in the product or using PassPort AM.

PassPort AM

The following PassPort AM features are available:

- An RBAC (Role Base Access Control) is used.
- The users and associated passwords can be stored in either a PassPort database, or retrieved from an LDAP directory or from Active Directory. Also, they can be retrieved from any other third-party product, but some code must be written to access these users.
- An SSO option is available. PassPort includes a reverse proxy and a token manager to provide an SSO option only for Axway products.

See the PassPort documentation for a complete list of features in this product.

When not using PassPort

The following features are available:

- An RBAC (Role Base Access Control) is used. Roles are defined through “Profile” objects and stored in Gateway object database.
- Each defined user have a profile assigned which controls actions that the user is allowed to execute on the different Gateway resources according to its role.

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

IP filtering

This section describes the IP filtering feature.

About IP filtering

It is possible to filter incoming connections in the DMZ with the help of an IP address list, which can either be a:

- Blacklist – allows everyone access, except for the partners contained in the list
- Whitelist – authorize access only to the members in the list

The address that will be checked against the IP list is the remote socket address as seen on the Router Agent side. If an intermediary system which will change the partner's address (for example, a proxy) is present, IP filtering in Secure Relay should not be used.

The IP list consists of a semi-colon separated group of IP addresses (for example, 10.133.56.14; 10.*.57.1?; support.axway.com; *.axway.net; fe80::8923:fec5:91ae:2b3e). The allowed IP address types are:

- IPv4 address in the well-known decimal dotted representation (RFC 820) with the possibility to specify wildcards. Two characters are allowed for wildcards: the star character ('*') will replace an entire group (for example, 10.15.1.* is equivalent to the range 10.15.1.1-10.15.1.255), whilst the question mark character ('?') will replace a single character (for example, 10.15.1.1?2 stands for the set 10.15.1.102, 10.15.1.112 to 10.15.1.192)
- IPv6 address in the known hexadecimal long and short notations (RFC 5952) with the possibility of using wildcards as above. Ranges are accepted for both IPv4 and IPv6.
- Hostnames (for example, machine-a2432)
- Fully qualified domain names, in short FQDNs, (for example, support.axway.com), with the possibility to specify the star character ('*') to replace one group (for example, *.axway.com)

Dynamic whitelist

Gateway can instruct Secure Relay Router Agent to create a list of IP addresses from accepted connections to listen points with Security Termination and Protocol Login. This feature is called the *dynamic whitelist*, in contrast to the *static whitelist* which requires user interaction.

- the dynamic whitelist can only be built in blacklist operation mode. IP addresses are added as soon as the TLS/SSH handshake is performed and authentication is successful. As such, they are added to the list of valid partners.
- Filtering using this list will work only in whitelist mode, in parallel with the existing static list for all listen points that have IP filtering switched on.
- Not all SAPs (Service Access Points) can contribute to the creation of dynamic whitelist, but they can benefit from its filtering capabilities by simply having IP filtering enabled.

Currently, the only method to remove IP addresses from the dynamic whitelist is to restart the Router Agent.

Lists and agents

By default, any IP address list (either whitelist or blacklist or even dynamic whitelist) is dedicated to each Master Agent, therefore:

- if multiple Gateways are connected to the same Router Agent, each Master will have its own list on that Router Agent.
- if one Gateway is connected to multiple Router Agents, then:
    - the *whitelist* or *blacklist* will be *identical* on each Router
    - the *dynamic whitelist*, will *differ* depending on what valid connections each Router Agent receives. This is because it depends on what valid connections each Router Agent receives.

Performance considerations

The recommended IP address types are IPv4 or IPv6, ranges included, because they have the best filtering performance. Hostnames and FQDNs without wildcards are almost as fast, because the name resolution is performed once, at initialization.

On the other hand, wildcards for IPv4 and IPv6 and FQDNs are checked with Java regular expressions with lesser performance. Performance is even less for FQDNs because a reverse name resolution is required for each connection.

Related topics

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Audit

Gateway stores the actions performed by users relating to the Gateway configuration in a dedicated audit file, to provide traceability of changes done to the product's configuration. This concerns changes both to global and to object configuration.

This functionality cannot be disabled, and the audit file can only be written by the audit process inside Gateway.

The content can be exported in a readable text format using the command line or UI. When Gateway is stopped, the audit can still be consulted using a dedicated command line tool.

You can archive the current audit file. Gateway then creates an archive file from the current audit.dat file by adding the archiving timestamp. To automatize the process of audit file archiving, you can create a scheduler to call a script calling the archive command.

The following operations are audited:

- creating, updating or deleting Gateway objects (Sites, Trading Partners, CGates/SGates, CGateGroups/ SGateGroups, Applications, Diffusion Lists, Decision Rules, Rule Tables, Models, Purge Models, Property Lists, Proxy, Users, Profiles).
- creating or updating Transfers
- starting/stopping a Site
- creating, updating or deleting a Gateway Security Object (TLS Security Profiles, SSH Security Profiles, Network Profiles, Certificates, Keys)
- creating, updating or deleting VFD objects
- control operations on log, statistics, connection statistics, connections and connected users (for more details, refer to Gateway User Guide / User Interfaces / Online commands / *Catalog of online commands/ pelctl*)
- configuration changes (not in peluconf standalone mode)
- purge operations

For further information refer to Gateway User Guide > Managing Audit files

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

payload_integrity

The purpose of *payload integrity* is to detect any tampering with files stored by Gateway.

You can enable the payload integrity check to assure that payload haven't changed between the moment is has been received by Gateway and further actions: routing to Integrator or routing to a third party. If activated, Gateway generates the signature for the incoming transfer's payload using Gateway private key and is validating the signature for the routed transfers. The signature for the routed transfers can be generated either by Gateway or by Integrator. The protocols for which the signature is computed in Gateway, for incoming transfers are: SWIFTNet, PeSIT and JMS.

Data integrity operations rely on digital signatures, so its usage assumes the following:

- Gateway is configured with pair of keys to build/verify a signature
- A public key performs signature verification on requests submitted from other sources
- Local operations, including requests submitted directly in Gateway (via command line or GUI), are performed only using the keys from Gateway.
- Requests submitted by Integrator are performed by Integrator's public key.

Payload integrity coverage

- For requests submitted by Integrator, when the option is enabled, data integrity is always enforced. Any submission without signature information fails.
- Currently, Gateway supports only RSA keys with the SHA256 hash algorithm in all payload integrity related operations (sign and verify).
- Requests submitted locally in Gateway are **not** covered by payload integrity
- For transfers initiated via command line tools or Navigator, it is possible to set up mechanisms to implement payload integrity.

Outbound transfers

For outbound transfers, data integrity is achieved by attaching a signature when a transfer request is submitted to Gateway.

- If the transfer is initiated by a local operator, the attached signature is generated using Gateway's private key.
- If the submitted by Integrator, the signature is generated with the Integrator's private key.

Once the transfer is actually started, the verification is performed with the pair of that key.

For further information refer to Payload Security.

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Security features in depth

Axway Gateway: Overview

Gateway includes a set of security features whose principal purpose is to protect the data stored and transferred over your network. This topic introduces basic security concepts and describes security implementation in Gateway.

Security concepts

File transfer security relies on three basic concepts:

- *Privacy*
- *Integrity*
- *Authentication*

Although these three concepts are independent, they must be combined to achieve a high level of security.

Privacy

Privacy (also referred to as *confidentiality*) involves protecting access to data. Privacy prevents third parties from accessing data that is not intended for them. Typically, data is made confidential through *encryption*.

Two cryptographic techniques are often used and combined:

- **Symmetrical encryption**: uses a unique key to encrypt and decrypt a message. Both connected parties share a key that they use to encrypt and decrypt data. The task of privately choosing the key before encrypting data is more problematic, since the connected parties share a single key that they cannot easily exchange.
- **Asymmetrical encryption**: uses a pair of keys, where each key can decrypt messages encrypted using the other key. This technique solves the problem of exchanging the key between connected parties. When two parties want to communicate using this encryption mechanism, each party publishes one of its keys and keeps the other one secret.

SSL or TLS protocols use a symmetric encryption mechanism to encrypt data (DES, triple DES, RC4, and AES). This mechanism generates a key for each session and exchanges this key during the handshake phase. They are transmitted using an asymmetrical encryption mechanism, which is slower but more secure (RSA) than a symmetric encryption mechanism.

Integrity

Integrity consists in ensuring that the received data has not been modified. Typically, integrity verification requires adding an abstract data to the data that you send. On receiving a message, the receiver makes an abstract of the message and compares it with the abstract sent with the message. If the abstracts match, message integrity is assumed to be valid. Otherwise, it is not.

The message abstract is referred to as the *message digest* or the *hash code*. The procedure used to create the hash code is referred to as either the *digest function* or the *hash function*.

When you use SSL/TLS or SSH, a hash code, referred to as **MAC** (Message Authentication Code), is added to

each message sent. The standard hash functions are **SHA-1** or **MD5**.

Authentication

Authentication is the process of verifying a claimed identity. The most widely used authentication mechanisms are:

- Password request
- Proof request

Password requests are associated with user login requests. When the user sends a login name, the password request is displayed.

Proof requests are associated with asymmetrical encryption mechanisms. The user sends the public key (or any information relevant enough to get the public key, such as a certificate), and then receives a random encrypted message. To decrypt the message, the user must own the unique private key associated with the public key. This means that if the user can decrypt the message, the user identity is authenticated.

When using TLS and SSL protocols, the client encrypts the symmetrical key with the public key for the server. If the server can decrypt the key, the client is as claimed. Otherwise, dialog is not possible and the connection is closed. When requested, the server uses a signature process to authenticate the client.

Authentication signatures

Message signing is an authentication technique that incorporates integrity methods. It consists in sending an encrypted hash code with the message.

When a signed message is received, both the sender identity and data integrity are verified by:

- Decrypting the signature
- Comparing the result with the calculated hash code

If the hash code is correct, data integrity is valid, and the sender is considered to be authenticated because the sender had the valid key to encrypt the digest.

Gateway includes functions that ensure security at each level of the file transfer process:

- *Protocol connection level security*
- *Transport level security*
- *Host system security*
- *Integration in secured environments*
- *DMZ deployment*

Protocol connection level security

With Gateway, you can analyze the identity of Transfer partners during protocol connection to determine whether or not you want to authorize them to connect and exchange files. You have the option of using internal or external processes for this identity control:

Internal processing: Using the CGate object

By default, Gateway uses the CGate object to process connection identification at the protocol connection level.

You can use the CGate object to link client access rights to connection authorization. You can define:

- File visibility for groups or sub-groups of users
- Access to portions of the VFD (Virtual File Directory) tree
- File transfer rights for individual users
- Filtering based on the port number and IP address of the caller

CGate connection mechanism

When a remote user attempts to connect to Gateway via a CGate, the following identification process occurs:

1. Gateway identifies a CGateGroup that corresponds to the connected user and then scans CGate objects within that group. Since CGates can contain wildcards (* and ?), this selection is based on a best matching process. Gateway selects the CGate object whose parameters most closely correspond to those taken from the network.
2. After selecting a CGate object, Gateway checks the protocol-dependent parameters, security parameters, and so on. If these checks fail, Gateway rejects the connection.
3. Gateway retrieves and merges the CGateGroup hierarchy for the selected CGate, to build a set of the following output parameters:
    - Protocol-dependent parameters for use while the connection is active
    - Connected user access rights to the VFD

External processing: Using the Protocol Connection Exit

Gateway provides the user exit ExitRcProtoConn that enables you to configure remote identification from a database or an LDAP directory. This exit overrides the default CGate identification mechanism described above. To activate Protocol Connection Exit scheduling as a replacement for the default identification process, set the Connection control method parameter in the configuration menu.

Transport level security

At the transport level, Gateway provides support for:

- Protocol and network translation
- SSL (Secure Socket Layer) / TLS (Transport Layer Security) on incoming and outgoing connections
- SSH protocol
- Network Profile objects that determine which Security Profile to use
- Security Profile objects to define security characteristics
- Certificates and keys to authenticate partners involved in the exchange

Protocol and network translation

Gateway is both a protocol-level and a network-level gateway. You can use it, for example, to route an incoming FTP Transfer over TCP/IP as an outgoing PeSIT transfer over a TCP/IP network . The external network and the

enterprise network can be completely different, which makes attacks against internal networks more difficult.

TLS/SSL security protocol support

SSL (Security Socket Layer) is a creation of Netscape. The Internet Engineering Task Force (IETF) used Version 3.0 of SSL to define the TLS protocol (Transport Layer Security). These very similar protocols operate between the network protocol layer (for example, TCP/IP) and application protocol layer (for example, FTP).

TLS functions independently of the file transfer protocol used. With TLS you can use any protocol except SFTP. This type of connection enables initiator and responder partners to communicate in a way that provides:

- Confidentiality – to prevent eavesdropping
- Simple (server-to-client) or mutual authentication - to prevent modification or manipulation of information
- Integrity – to prevent message forgery

Since TLS security may involve mutual authentication of connection partners, Gateway includes an internal stand-alone X.509 certificate manager.

Gateway may be included in a more global certificate manager, commonly known as a PKI (Public Key Infrastructure). PKI integration requires specific development through security server exit customization.

Remote access from Axway MFT Navigator

The Axway MFT Navigator, enables you to perform actions on Gateway from a remote machine using a Web browser. These commands are transmitted to Gateway using a proprietary protocol (also called APICS) that runs over a TCP/IP network.

This type of TCP/IP connection is vulnerable to attacks that may endanger the whole information system involving Gateway. You can therefore secure the connection using TLS.

SSH security protocol support

SSH (Secure Shell) is a protocol that provides secure remote login and other secure network services over an insecure network. SSH guarantees:

- Authentication (via certificates or keys)
- Integrity
- Confidentiality

The SSH File Transfer Protocol (SFTP) enables you to exchange files between two remote machines. SFTP is an application-level protocol that is used as a generic protocol for communication between users and proxies or gateways to other Internet systems. It relies on the SSH protocol that establishes a user login and secure connection.

SSH and SFTP are separate and technically different, but are exclusively used together in Gateway. Gateway supports SFTP in client and server modes.

SFTP/SSH links require the use of a DSS, ECDSA or RSA key (or certificate) and the configuration of the following Gateway objects:

- Remote Site

- Network Profile
- SSH Profile

Network Profile

Gateway uses Network Profile objects to determine whether an incoming connection must be secured. Network Profiles contain matching rules that Gateway applies to the incoming connection parameters (origin address, SAP…). From the Network Profile that best matches the incoming connection, Gateway determines which Security Profile to use. (The Security Profile contains all required security elements.)

The Network Profile object contains the following parameters:

- Network type (TCP)
- Called address
- Calling address
- Security option (TLS, SSH or none)
- Security Profile to use if the Security option is set to SSH or TLS

When Gateway receives an incoming connection, it selects a Network Profile and a Remote Site that match the received network parameters (called address and calling address).

Security Profile

Gateway uses the Security Profile object to establish security for both incoming and outgoing connections. It is a configuration set that determines which security protocol to use, which encryption method, and so on.

Gateway provides two types of Security Profile:

- TLS Profile
- SSH Profile

In most cases, when you use TLS/SSL or SSH protocols, the connection is secured as soon as you establish a network connection. When an incoming connection occurs, Gateway uses the Network Profile to determine whether to use TLS/SSL or SSH.

SSH Security Profile

To establish an SSH connection, a Security Profile is mandatory. You can define several Security Profiles in the database, which can be of the type *Client* or *Server*.

Security Profiles contain all the parameters required to establish the SSH transport layer, such as:

- Key exchange algorithms
- Encryption
- MAC (Hash)
- Server key
- Client authentication method

Gateway selects the Security Profile in the following way:

In client mode, the outgoing transfer is destined to an SFTP Site. The Remote Site contains a parameter that indicates which Security Profile to use. The Security Profile is of the type CLIENT and is enabled.

In server mode, an incoming connection arrives at your Gateway server. This connection is characterized by its calling address and its called address. Gateway then searches for a Network Profile with a called address and calling address that match those of the connection. The Network Profile that Gateway selects must have the security option set to SSH and must contain a valid Security Profile name. This Security Profile must be of the type SERVER and must be enabled. If Gateway does not find an appropriate profile the connection is rejected.

TLS Security Profile

You can configure cryptographic and authentication parameters via the TLS Profile object. Gateway selects the Profile to use for a TLS session depending on the way the connection between the partners is established (incoming or outgoing connection).

An incoming connection in TLS has a minimum of two levels of verification. The first is identification of a corresponding Network Profile object and the second is the identification and checking of the Remote Site object and its parameters.

For an outgoing connection, the Remote Site parameters determine whether or not to use a TLS Security Profile.

Certificates and keys

SSH requires client and server authentication. Client and server can be authenticated by public key or by certificate. With either method, server authentication is a two-step process beginning with server identification via a public key and followed by specification of the preferred authentication method.

In TLS/SSL, the public key is integrated into the certificate.

Certificates

A certificate is an electronic document that:

- Identifies an individual, a server, a company or any other entity
- Associates that identity with a public key

When you communicate with any remote party, you determine the identity of that correspondent, before dealing with data integrity and confidentiality. Forbidding access to data and verifying the integrity of the received data is meaningless if you are not sure of the identity of the other party. Certificates help prevent the use of fake public keys for impersonation. Only the public key certified by the certificate works with the corresponding private key that belongs to the entity identified by the certificate.

You can import the following types of certificate into Gateway:

- Root (top-level CA certificate)
- CA (Certification Authority)
- User
- Other

Keys

SSH uses DSS, ECDSA and RSA keys (or RSA keys in certificates) to authenticate partners. Gateway supports DSS, ECDSA and RSA keys and certificates with RSA keys.

The private key is a local key that is assigned to Gateway (or an application using Gateway). It has a public part and a private part. You can configure Gateway to import keys automatically. Alternatively, you can import keys manually.

Host system security

Gateway provides complete protection for the host system via:

- Virtual File Directory
- Database encryption
- Temporary file encryption
- User Access Control of Gateway objects

Virtual File Directory (VFD)

Internally, Gateway represents each Transfer as a transfer record registered in the Mailbox. However, transfer records are not organized hierarchically. To provide data presentation and organization, Gateway supplies a virtual device used in association with the Gateway Mailbox, called the Virtual File Directory (VFD). The VFD provides a logical and internal representation of files and directories. Remote clients have only a logical view of files. Clients cannot directly access the associated physical files (stored on the Gateway host machine).

The VFD is available to connected HTTP, FTP and SFTP clients. Using the VFD feature, Gateway behaves like any standard FTP SFTP or HTTP server.

The VFD offers two running modes that you can operate separately or simultaneously:

- Logical mode: The VFD is a logical representation of directories and files (also known as VFD Items). These logical files represent a duplication of a given Transfer Request extract in the Gateway Mailbox. From the Gateway client side (FTP, SFTP or HTTP clients), you can view the logical representation of these files, but cannot directly access them.
- RFD (Real File Directory) mode: The RFD is supported by a mount mechanism that links the logical file system to the physical file system of the Gateway host machine. The files seen from the client side are real files that are located on the Gateway host machine. Optionally, you can associate these files with a Gateway Mailbox transfer record.

The content of a real directory (or a mounted directory) is an exact image of the content of the corresponding directory on the Gateway host machine File System. When you make a file available for clients, you effectively make a physical copy of the file in the corresponding directory.

Database encryption

You can use AES (Advanced Encryption Standard) encryption mechanisms to store Gateway administration objects and/or the Mailbox file securely on the host machine. Only users with specific rights can read, modify, or delete files. This encryption mechanism prevents external attacks. No one can access data that could enable them to bypass the Gateway authentication process.

Gateway integrates an internal PKI (Public Key Infrastructure) to encrypt the internal security database that stores security objects and all certificates and private keys.

Additionally, an external PKI is accessible via user exits:

- To access an external LDAP certificate directory
- To perform online certificate validation: OCSP

Temporary file encryption

A network machine located in a DMZ can be accessed by machines in unsecured areas. It may therefore be subject to malicious attacks.

To route files, Gateway must make a local copy of the files before routing them. Gateway uses *temporary file encryption* to protect the files it temporarily stores from malicious attacks. This is particularly important when you deploy Gateway in a DMZ. In a DMZ deployment, you should take the additional precaution of encrypting the Gateway Mailbox.

When you route files using Gateway and temporary file encryption, you must link the input transfer to the output transfer using the combination of a Decision Rule (to identify the incoming transfer) and an applied Model (to specify the outgoing transfer characteristics).

This enables Gateway to identify the security key necessary to read the deposited ciphered file in order to process it for the subsequent transfer.

Alternately, you can use the Axway Secure Relay module for your DMZ deployment. With this solution, Gateway and all Gateway files are located within the protected zone.

User Access Control of Gateway objects

User Access Control applies to all methods of accessing Gateway objects: via the Gateway GUI, command lines, APIs or the Axway MFT Navigator.

The *User* and *Profile* objects in Gateway enable you to define user identification and access rights. Typically only Administrators have access to these objects. The User object defines properties for each user and the Profile object defines the type of operations a set of users can perform on each Gateway object.

In the Profile object you specify the operations (such as Read, Write or Delete) allowed on the different types of Gateway object (Sites, Models, Transfer Requests, and so on).

For example, you could specify that all Users associated with a given Profile are authorized to access Site objects in Read and Write mode only.

Alternatively you can use Customizable user exits or PassPort AM to manage user access control.

Integration in secured environments

Gateway is fully compatible with SOCKS4, FTP and HTTP proxies.

DMZ deployment

You can further enhance the isolation and security of your enterprise network by deploying Gateway in a DMZ architecture.

Next topic

Related topics

About CGates and CGateGroups

Internal user exits

Managing TLS

SSH protocol

Network Profiles

Managing SSH Security Profiles

Managing TLS Security profiles

Certificates and keys

About Virtual File Directories

About Access Control Management

About Proxy objects

Overview: DMZ deployment

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

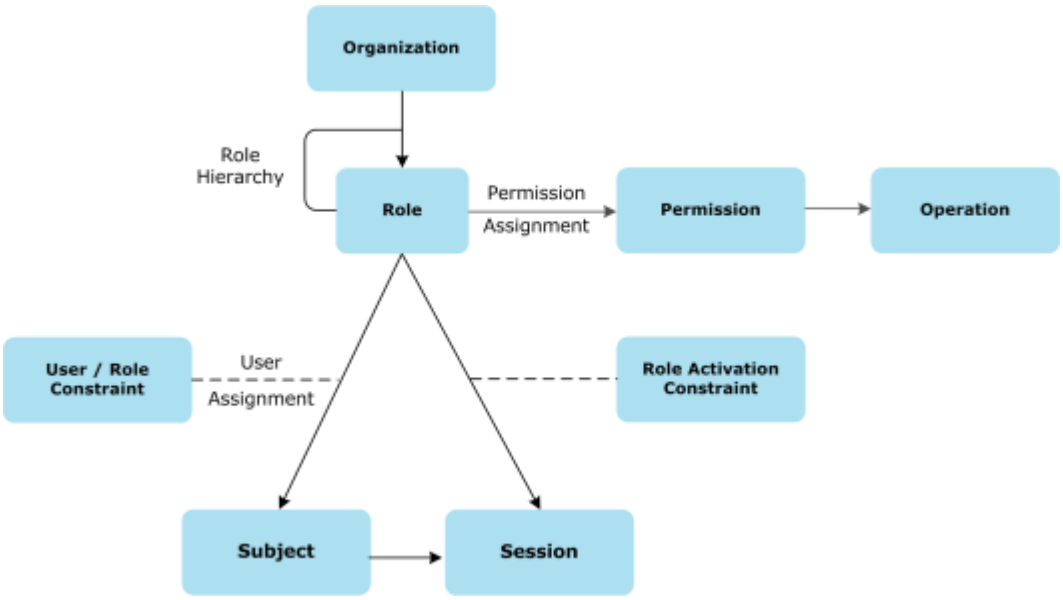
Identity and access management

6

The RBAC model

This product implements an RBAC model to manage identity and access control. This model is based on the concept that there are actually few roles within an organization, and instead of assigning access to each individual resource, the permission can be assigned to roles, and then those roles assigned to users. This process makes it much easier to manage permissions. It is not necessary to review the complete list of users each time a new resource is added, but only to review the list of roles. In the same way it is not necessary to review the complete list of resources each time a new user is added, but only to review the list of roles.

The following diagram illustrates an RBAC model.



The following describes how the RBAC model works:

- A role is a list of permissions (or privileges).
- A permission is the right to execute one or multiple actions on one resource.
- When using a permission, it becomes an operation.
- Roles can be grouped to create hierarchical roles.
- Roles belong to an organization.
- When assigning a role to a subject (or user), it is possible to create a User/Role constraint to limit the scope of the role for this subject.
- To define these constraints, attributes can be associated with Resources.

- When the subject starts using a role, a session is created. When creating the session, it is possible to limit the scope of the role for this particular session through a role activation constraint.

For example:

- A resource is a file.
- Actions on this resource are: Create, Read, Update, or Delete.
- A permission (privilege) on this resource is Read File.
- An attribute of this resource is Directory Name.
- The User/Role constraint is If directory name=XYZ.
- The role RRR is created including this permission.
- The constraint is added when assigning this role to a subject (user).

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Available resources, actions and privileges

The following is a list of available resources in Gateway. Included for each object are associated actions, attributes, and a description.

Generic resources

Resource	Description	Resource actions
Sites Objects	In Gateway, to define the parameters of a file transfer, you require a description of the partner for the file exchange. You specify the characteristics of partner sites in Site objects. **Note**: Read Configuration rights are needed to correctly use the write/ update access rights for the Site Objects.	• Read • Write • Update • Delete • Admin
Applications Objects	When you define a File Transfer Request you can provide certain file system attributes via the Application object	• Read • Write • Update • Delete • Admin

Resource	Description	Resource actions
Transfers Objects	Gateway provides a set of objects that enable you to create and define the characteristics of a Transfer.	• Read • Write • Update • Delete • Admin
Models Objects	Gateway Model objects enable you to group sets of attributes for the creation of Transfer Requests, Decision Rules, Axway Messaging connector objects	• Read • Write • Update • Delete • Admin
Diffusion lists Objects	Gateway enables you to create lists of computers that you want to use as destinations for a given type of file.	• Read • Write • Update • Delete • Admin
Users Objects	You define each Gateway User by specifying a unique name, an access password and the name of the Profile object with which the User is associated. All Users, apart from the Administrator, must be associated with a specific Profile. All Users associated with the same Profile have identical rights.	• Read • Write • Update • Delete • Admin
Profiles Objects	The Profile object defines the type of access rights that a user, or a set of users, has to specific objects or files.	• Read • Write • Update • Delete • Admin
Log Objects	The Log file contains a trace of all significant events relating to transfer monitor activities and to the transfers performed.	• Read

Resource	Description	Resource actions
		• Write • Update • Delete • Admin
CGates Objects	CGate and CGateGroup objects are identification filters used internally in server mode at the protocol connection level to control the connections of remote partners. **Note**: Read Configuration rights are needed to correctly use the write/ update access rights for the CGates Objects .	• Read • Write • Update • Delete • Admin
VFD Objects	The Virtual File Directory (VFD) is a virtual device that enables you to provide an administrator-controlled view of data organization to specific users via a file directory tree. The VFD comprises directories and files.	• Read • Write • Update • Delete • Admin
Rule Tables Objects	A Rule Table object is an ordered list that contains one or more Decision Rules.	• Read • Write • Update • Delete • Admin
Decision Rules Objects	In a Decision Rule object, you define one or more events that you want to trigger a specific action.	• Read • Write • Update • Delete • Admin
Proxies Objects	In Gateway, you use the Proxy object exclusively in client mode for contacting a proxy server. The protocol settings of a Proxy object define the TCP address and access port for the connection server and the set of connection parameters (for example, the user name and corresponding password).	• Read • Write

Resource	Description	Resource actions
		• Update • Delete • Admin
Trading Partners Objects	Provide the information to Gateway necessary for compression, signature, encryption and encapsulation operations you create a Trading Partner object. **Note**: Read Configuration rights are needed to correctly use the write/ update access rights for the Trading Partners Objects.	• Read • Write • Update • Delete • Admin
Transfer Security Objects	Gateway integrates a set of objects that you can use to manage your transfer security requirements: Certificates and keys, Network Profiles, TLS Security Profiles, SSH Security Profiles. **Note**: Read Configuration rights are needed to correctly use the write/ update access rights for the Transfer Security Objects.	• Read • Write • Update • Delete • Admin
Connections Objects	A Connection object contains information describing a protocol session.	• Read • Write • Update • Delete • Admin
Connected Users Objects	A Connected User is a virtual object that is identified with the Site used for these connections. All information that single connections carry is accumulated at the Connected User level.	• Read • Write • Update • Delete • Admin
Mailbox	Each time Gateway processes a Transfer Request it creates a record of the Request and stores it in the Mailbox.	• View • Manage

Resource	Description	Resource actions
Security Certificate object	enables you to import a certificate from a file into the local database	• Import • Display • Modify • Export • Delete
Key Object	Piece of information used for cryptographic operations	• Import • Display • Rename • Delete Enable/ Disable
User Exits	The exits are triggered by different types of Gateway processing event. They enable you to integrate your own custom processing routines with Gateway standard processing activity.	• Create • Delete • View • Modify
Security Profile Object	Gateway uses the Security Profile object to establish security for both incoming and outgoing connections.	• Create • Delete • View • Modify
Network Profile	Network Profiles associate an incoming network connection request with a Security Profile.	• Create • Delete • View • Modify • Filter
Statistics file	Formatted text file that contains records of terminated transfers.	• Archive • Suspend • Resume

Resource	Description	Resource actions
Audit Objects	The Audit file contains a trace of actions performed by users on the Gateway configuration (global and objects configuration).	• Read • Admin
Configuration	Gateway configuration objects **Note**: Read Configuration rights are needed to correctly use the write/update access rights for the Transfer Security Objects, Sites Objects, CGates Objects and Trading Partners Objects.	• Read • Admin

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Predefined default roles

Predefined default roles

The following is a list of predefined roles in Gateway.

- Admin
    - Enables access to all the resources. Should be assigned to a system administrator which should have access to the system's resources and architecture.
    - Can create other user profiles in accordance with specific requirements.
    - Should be employed using all security best practices to include but not limited to: strong passwords, not used for current application activity but only for product management, along with audit trails and proper ACL employment.
    - Using the Admin *default* role should be restricted as much as possible. In many cases, administrative tasks can be delegated to user roles with specific access rights to individual Gateway resources.
- Access all transfers
- Anonymous
    - This user can connect with no credentials and has access to all resources.
    - It is meant to be an initial setup user and it should not be allowed to function after initial setup as it presents a great security risk in a production environment.
- Customizable User

    - General user which should be custom tailored to accommodate the need to perform certain actions by certain users. These can be clustered in common user profiles but for audit purposes a finer grained access control should be employed through an individual basis.

Through Gateway's user profile management, a fine grained access control can be implemented and thus numerous types of users and user-profiles can be created and managed, given a variety of access privileges to the application's resources.

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Configuration 7

This section contains information on:

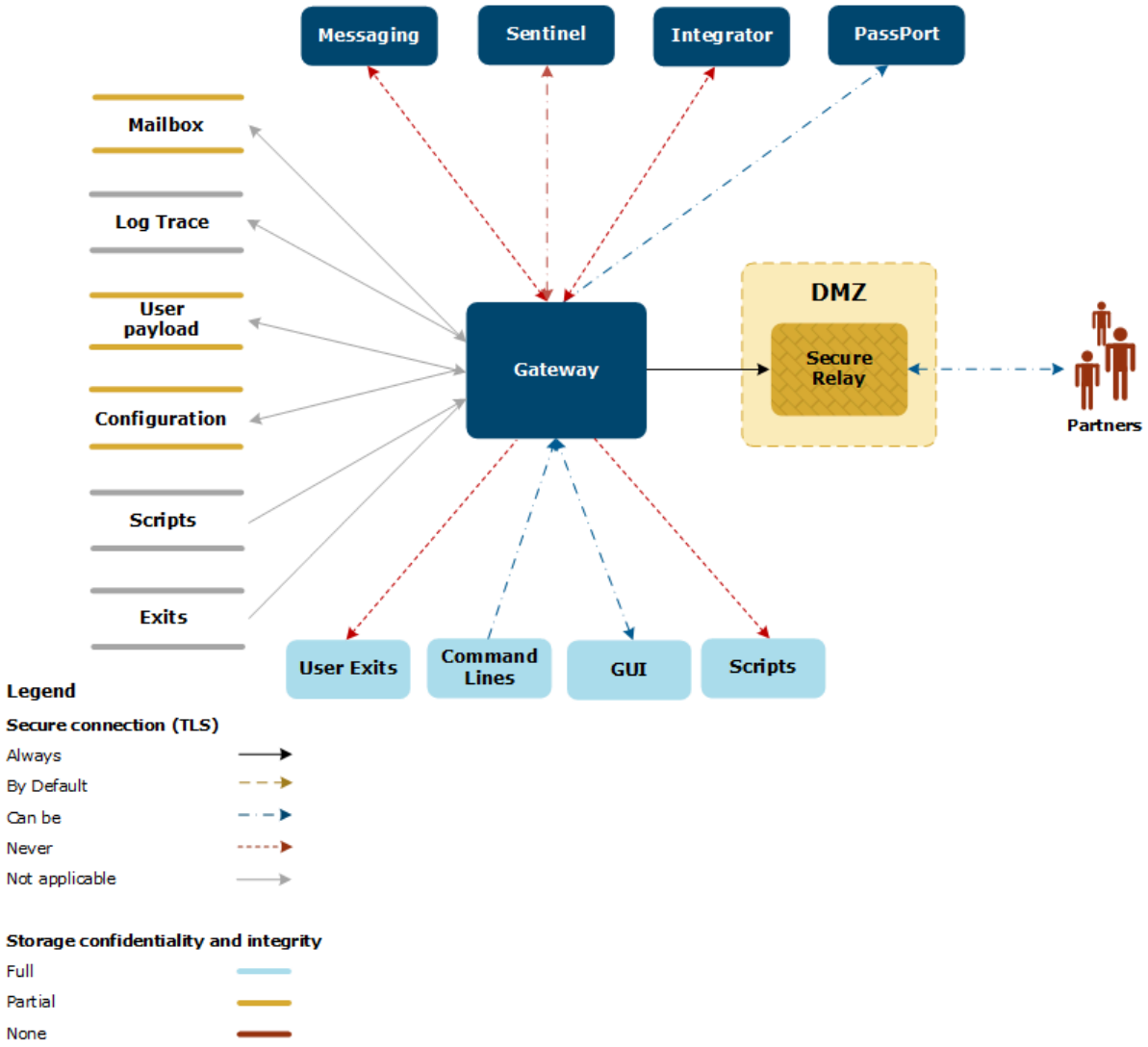
- *Security architecture*
- *Security configuration*
- *Product certificates*

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Security architecture

Security architecture



Gateway Security Architecture diagram

Gateway security diagram - legend

Security configuration

Creating an SSL/TLS connection

- Server role
  For SSL/TLS securing the server role on one of the supported TCP protocols, the user must:
  - Create a SERVER type SSL/TLS security profile object with details about the TLS version to use, cipher suites and algorithms supported, client authentication policy, certificate to use for server authentication, trust policy, trusted CAs, extensions used, control and data connection security policies (especially for FTP and AS3)

        - Attach the SERVER SSL/TLS security profile created previously at the network profile object attached to the listen port for that protocol and set the network security type to TLS
- Client role
  For SSL/TLS securing the client role on one of the supported protocols over TCP, the user must:
    - Create a CLIENT type SSL/TLS security profile
    - In the remote site used for connecting to a partner, set the network security to TLS and attach the CLIENT SSL/TLS profile as the profile used for outgoing connections
    - Optionally, if the remote site is used also for handling incoming connection, a SERVER SSL/TLS profile can be attached for the incoming connections

Creating an SFTP connection

Server side

For a server side SFTP (SSH) connection, proceed as follows:

- Create a SERVER type SSH security profile where to specify the algorithms (HMAC, public key, encryption etc) supported, the key to use, the trust policy
- On the network profile for the SFTP listen port, to set the network security type to SSH and attach the SERVER SSH profile

Client side

For a client side SFTP (SSH) connection, proceed as follows:

- Create a CLIENT type SSH security profile
- In the remote site used for connecting to a partner, attach the CLIENT SSH profile. If no profile is attached to the remote site, Gateway will look by default to a SSH client profile named SFTP_OUT

Importing certificates and keys

- From command line
    - For importing keys, use `secadm import_key` command
    - RSA keys up to 4096 bits can be imported (DER or PEM format)
    - DSA keys up to 1024 bits can be imported (DER or PEM format)
    - ECDSA nistp256, nistp384 and nistp521 keys can be imported (DER or PEM format)
    - For importing certificates use secadm import_cert command
    - Certificates in PKCS12, DER, PEM(BASE64) or PKCS7 formats are supported
    - Private keys can be in the same container or in a separate file; formats DER, PEM (BASE64), PKCS8 and CR_PKCS8 are supported
    - RSA and ECDSA key based certificates are supported

- From GUI
    - For importing keys, right click on Security Management->Transfer Security Management->Key and select Import
    - For importing certificates, right click on Security Management->Transfer Security Management->Certificate and select Import
- Auto import keys and certificates
    - During the SSH and SSL/TLS handshake, partner keys and certificates can be auto imported in Gateway store if the security profile is configured for auto-import
    - Keys and certificates are imported in disabled state and the first handshake attempt fails
    - Enabling the imported keys and certificates requires user action and should be done only after partner validation

Associating keys and certificates to partners

- Keys
    - Keys alone are used only in the SFTP protocol
    - If the trust state of a key (the fact that it is imported and enabled) is not secure enough with respect to the company policies, the key alias can be attached to a remote site handling the incoming connection as a remote site authentication parameter. In this way, the key is associated to the partner corresponding to that remote site
- Certificates
  Certificates can be used:
    - For TLS/SSL authentication
    - If the trust state of a certificate (imported, enabled and having a trusted certification path) is not enough, the certificate alias can be attached to a remote site handling the incoming connection as a remote certificate filter. In this way the certificate is associated to the partner corresponding to that remote site
    - For CMS authentication (OFTP v2)
    - The certificate used for the CMS authentication in OFTPv2 case is selected in the OFTP tab of the remote site corresponding to the partner
    - For payload signature and encryption
    - The certificate/certificates used for payload signature and encryption (EDI protocols – AS1, AS2, AS3, SMIME mail, OFTPv2 and RosettaNet over mail and HTTP) are selected by their alias in the trading partners used in the flow

Securing the GUI connection to Gateway server

For securing the Navigator connection to Gateway server, do the following:

1. From the server configuration (in Gateway Navigator), select **Connectivity**->**Navigator server** and check the **Use TLS** option. Indicate which user certificate to use for the server. The password for accessing the p12 certificate must be stored using the PKCS5 standard (see the chapter about Password management)

2. **Or**, from the command line set [cstcp]TlsActive='1' , [cstcp]TlsServerCertificateFile and [cstcp]TlsPasswordFile parameters..
3. When connecting from Navigator and setting up the server connection, check also the **TLS** option and add the appropriate certificate for validating the server certificate (the server CA).

Securing the connection to Passport

Connections to Passport (AM, PM and PS) are not secured by default. Security can be activated at installation time or later and security parameters must be provided.

- Passport AM and PM
    - For Passport version 4.4 and higher, a mutual authenticated TLS connection can be used

      At installation time (or later if necessary) a path to a client certificate used for mutual authentication and the password to access this certificate are provided. The client certificate does not exist at this moment.

      At Gateway first start, the product self-registers to Passport. A key-pair is generated and the public key is sent to Passport in a Certificate Signing Request. Passport signs the certificate and returns it to Gateway. Then, using the private part of the key and the provided password, the certificate is packaged into a P12 format certificate. This resulting certificate will be stored in the configured location and will be used for mutual authentication.
    - For Passport versions below 4.4, only server side authenticated TLS connection can be used. In this case, the user must authenticate to Passport using login and password.

      For server side authentication, only the Passport SSL CA certificate, provided with Gateway installation, a user login and a password are necessary to connect to Passport in a secure manner.
- Passport PS

  For securing the Passport PS connection, only server side authenticated TLS connection is available. Connection is not secured by default.

  For connecting to Passport PS in a secure way, the Passport SSL CA certificate (provided in Gateway installation), a user login and a password are necessary.

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Product certificates

Gateway product is delivered with the CA certificate needed to connect to Passport product (Passport SSL CA certificate). This certificate is delivered in extras\PassPort\ directory of the installation and corresponding configuration parameters for accessing the CA certificate are defined to this path by default.

Gateway does not deliver sample certificates.

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Security best practices 8

Self-signed certificates

Using self-signed certificates may also be a security risk for many reasons, including:

- Anyone can generate his own certificate and you need a very secure process to receive/send these certificates to make sure they are coming from the right partner. When using CA-signed certificates, you can rely on the CA.
- If self-signed certificates are not securely stored, anyone change them. CA-signed certificates also must be securely stored, but no one can change them.
- There is no way to revoke self-signed certificates.

Privileged access user list

Maintain a list of users with privileged access. At a minimum, maintain a list of administrators and a list of users with access to multiple functions.

Internet access limitation

As much as possible, limit the number of internet access points. Do not open useless Internet connections and limit interconnections with external networks as much as possible. This reduces the risk of external attacks and makes it easier to audit the product.

Correct upgrade procedure

In the event of a possible vulnerability discovered in the product, you must be able to apply the patch or new Service Pack as soon as possible. Make sure you have the correct procedure to complete the upgrade. Always use the latest version of the product, if possible, as it contains fixes to known vulnerabilities.

Generic or anonymous users

The term “generic users” means that the password is shared among multiple specific users. This makes it easier for an attacker to retrieve this password. In addition, the procedure to change shared passwords can be complicated and risky. In case of an incident, these generic or anonymous users make it impossible to determine who completed the erroneous action.

Password policy

Password policy refers to size and complexity of the password, as well as to all the rules to manage this password. The size and complexity is important. The policy should define that the length be a minimum of 8 characters, contain a mix of alphabetic characters with numbers and special characters, and be case-sensitive.

Your password policy:

- Must force passwords to be change periodically.
- Should prohibit reuse of a password before n number of different passwords and within a certain period of time.
- Must define how the password is created differently from the old one (such as: at least a certain number of different characters).
- Must limit the number of failed attempts.

Default authentication account

The product is delivered with the following default users:

- User: admin
- Password: can be chosen at install time. If none provided default password is admin

Your first task after login with this user must be to create your own administrator account and to delete the default one, or at least to change this administrator password.

Remote connections

You should limit your remote connections in the following ways:

- If no one needs to access the product remotely, make sure that the UI ports are closed in your firewall.
- If someone needs to connect remotely only occasionally, set a procedure to open the port only on demand.
- If there are regular remote users, make sure that the connections are as secure as possible, such as using HSTS or VPN.

Sensitive files and databases

You must protect your sensitive files and databases. The configuration file contains information that can be useful to a hacker. Even if part of this information is encrypted, access to this file must be restricted by your local access management. Only the product and one or two administrators should have access to this file in any mode (read, update, and delete).

The files containing the password used to generate encryption keys must be protected as well. This file is encrypted, but access to this file should only be granted to the product.

The file path to the passphrase used for encrypting the mailbox and database objects is configured through [monitor]ciphering_passfile parameter.

Location of files

- The configuration files are located in $Gateway_Home/run_time/etc.
- The name and location of the encryption key generating files are established when creating them using the pelencpass command.

List of sensitive files

The following files should have a high level of protection in Gateway:

- the **mailbox** and the **object database files** in `run_time/data` directory of the installation:
    - mailbox: `xfer.dat` and `xferb.dat`
    - object databases: admin.dat, agenda.dat, cert.dat, cgate.dat, cgategroup.dat, dlist.dat, key.dat, model.dat, nprof.dat, profile.dat, proxy.dat, rules.dat, sgate.dat, sgateb.dat, sgategroup.dat, sgategroupb.dat, sprof.dat, sshprof.dat, trade.dat, user.dat
    - The mailbox and object database files can be encrypted for higher level of protection, but they are not by default.
- **certificate files**
    - certificate for connecting to Passport - [passport]client_cert_file in configuration
    - certificate for connecting to XSR RA - [xsr]master_cert_file in configuration
    - certificate for connecting to Gateway server using the Navigator - [cstcp]TlsServerCertificateFile in configuration
- **password files**
  password files for XSR client certificate:
    - [xsr]master_cert_password_dkfile
    - [xsr]master_cert_password_saltfile
    - [xsr]master_cert_password_datafile

  password files for Passport client certificate:
    - [passport]client_cert_password_salt_file
    - [passport]client_cert_password_dk_file
    - [passport]client_cert_password_file

  password files for Passport login
    - [passport]password_salt_file
    - [passport]password_dk_file
    - [passport]password_file

  password files for XSR PKI login
    - [xsr]pki_password_dkfile
    - [xsr]pki_password_saltfile
    - [xsr]pki_password_datafile

  shared secret files for Passport
    - [xsr]pki_password_dkfile
    - [xsr]pki_password_saltfile
    - [xsr]pki_password_datafile

  password files for Swift HA database login
    - [swnet]swnet_ack_userexit_db_salt_file
    - [swnet]swnet_ack_userexit_db_dk_file
    - [swnet]swnet_ack_userexit_db_password_file
- **configuration files**

- run_time/etc/conffile – static configuration
- run_time/etc/pelsetup.ini – static configuration
- run_time/etc/dconfsave.ini – dynamic configuration
- run_time/etc/pelsetup.def – default values

- **temporary files**
  Default location for temporary files is in `run_time/tmp`. The temporary files can be AES128 encrypted for a higher level of protection, but they are not by default. Use `[monitor]received_file_ciphered configuration` parameter to activate the received file encryption.

Recommendations for files

When the product package is downloaded on the Axway Support web site, it is highly recommended:

- to compute the hash of the downloaded file (a tool is required to do that)
- to compare it with the hash that is displayed in the page you reach when clicking on the name of the package.
  Both SHA 256 and MD5 hashes are displayed but it is safer to use SHA 256.

It is also recommended to protect files integrity after the product has been installed, using any file integrity monitoring tool.

File to monitor		Affected by...			
Directory to monitor	files	Transfer mangmt	Partner mangmt	Event mangmt	Securi
$Gateway_Home/bin	*.dll,*.exe	Yes	No	No	
capint	*.dll,*.LIB	Yes	No	No	
UserGuide* & sub-directories	*	Yes	No	No	
$Gateway_Home/extras	*.dll,*.LIB	Yes	No	No	
PassPort	*	Yes	No	No	
Perl & sub-directories	*	Yes	No	No	
Sentinel & sub-directories	*	Yes	No	No	
swnet & sub-directories	*	Yes	No	No	
$Gateway_Home/inc	*	Yes	No	No	
$Gateway_Home/jre*	*	Yes	No	No	
$Gateway_Home/licences	*	Yes	No	No	
$Gateway_Home/META-INF	*	Yes	No	No	
$Gateway_Homen/nlt & sub-directories	*	Yes	No	No	
$Gateway_Home/obj	*	Yes	No	No	
exitclnt	*	Yes	Yes	No	
exitextc	*	Yes	Yes	No	
exitrte	*	Yes	Yes	No	
exituser	*	Yes	Yes	No	
migexit	*	Yes	Yes	No	
$Gateway_Home/run_time/				No	
data	admin.dat, trade.dat, cgate*.dat,	Yes	Yes	No	

File to monitor			Affected by...		
		sgate*.dat, dlist.dat, proxy.dat			
		cert.dat, nprof.dat, key.dat, sprof.dat, sshprof.dat	Yes	Yes	No
		model.dat			No
		rules.dat, agenda.dat			No
	etc	*.bat, *.ini,	Yes	Yes	Yes
	scripts & subdir	*	Yes	Yes	Yes
$Gateway_Home/tools & sub-directories		*.dll, *.exe, *.jar,	Yes	No	No

Notes:

- Files only modified on specific actions may also be modified during upgrade
- All others file are modified at runtime and cannot be controlled by the monitoring tool.

Identified threats and possible mitigations

During threat modeling process, several possible threats or weaknesses were identified:

- **User exit and event scripts customization**:
  - Gateway provides different user exits as entry points for custom code that modifies the Gateway behavior in different points of the flow of data. Please see the *User Guide* > *External User exit* section for details about different user exits available.
  - Also, in Gateway, you can define routing and scheduling rules that will handle different events. One of the routing/scheduling methods is through launching a script (e.g. perl, shell script, batch file).
  - Both user exits and event scripts represent custom code that runs in Gateway context, meaning it will have the same privileges a Gateway process has.
  - Please be aware that running custom code in Gateway context implies some potential risks if security best practices are not followed when writing custom user exits and event scripts.
  - Risks include but are not limited to:
    - compromise overall system functioning
    - compromise different Gateway functionalities
    - execute unwanted random/harmful code/commands
    - compromise confidential data
  - As Axway has little to no control over the way user exits are implemented, it is the implementor's responsibility to follow security procedures and best practices when writing custom code that will run in Gateway context.
  - Possible mitigations and best-practices:
    - always review and test a piece of code before running it in production
    - use static and dynamic analysis tools (where available) for instrumenting, profiling and testing custom user exit and event script code
    - never trust input data of the user exit or event script. Some of the values that a user exit or event script takes as input may come from untrusted sources (i.e. over

the internet), even if it transits Gateway first. Gateway itself cannot encode/sanitize user input before passing it to the exit procedure, as Gateway doesn't know in advance the context where that data will be used and how to encode it/sanitize it. It is the custom user exit or event script implementors responsibility to validate input according to the data usage context.

- pay attention to buffer allocations and capacities. Always deallocate heap memory allocated within an exit. Always check sizes before reading/writing
- avoid (where possible) launching new processes from the user exit. The child process would run with Gateway privileges but will not be under Gateway's control. If you do launch child processes from the user exit, the same security concernsand verifications should be applied to the child process as for the user exit code itself.
- avoid storing or transmitting sensitive data in clear text

- **User / password can be sent plain text trough the URL**
  - Possible mitigations and best practices: for HTTP connections, use TLS security over TCP (HTTPS) with mutual authentication. Avoid using Web basic authentication, where credentials are sent through the URL. Choose more secure authentication methods, like session cookie ID.
- **Private keys are not encrypted by default**
  - Possible mitigations and best practices: database object files, including keys and certificates database files are not encrypted by default. It is recommended to AES128 encrypt the object database files using [monitor]ciphering_option configuration parameter.
- **No mechanism that forces the change of the default password of the default admin user**
  - Best practice: if not set at installation time, change the default password of the predefined admin user immediately after the installation.
- **Configuration & the security configuration can be changed by command line that does not request any authentication**
  - Best practice: it is recommended that access to configuration and database files be strictly controlled. See also the previous paragraph regarding the sensitive files.
- **User passwords are not hashed and Salted in the system**
  - Possible mitigations and best practices: CGate and SGate objects can keep the user passwords hashed, but they are not configured to do so by default. It is recommended to activate this option through [monitor]disable_password_regex='yes' and [monitor]cgate_hashed_password='yes' configuration parameters.
  - For the other passwords it is recommended to encrypt the object database files. It is recommended to AES128 encrypt the object database files using [monitor]ciphering_option configuration parameter.

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Security administration 9

Axway Gateway: Managing Security

For information about Security Management in Gateway, refer to:

Section	Describes...
Password management	• How to create a Secure Password Storage using pelencpass
Access Control Management	• How to control access to Gateway objects • How to create and manage Users and User Profiles
Transfer Security Management	• How to manage your file transfer security requirements • The set of Gateway objects you use to manage file transfer security
DMZs and Firewalls	• The deployment options that enable you to link Gateway to the Internet through a Demilitarized Zone (DMZ) – for example, Secure Relay • The management of TCP ports for firewalls
Payload integrity	• Ensure payload integrity by attaching signatures to all received files, then using that additional information when the file is routed to a third party

For an overview of security in Gateway, refer to Overview > Security features.

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Password Management

Secure Password Storage

Password-based authentication usually requires a you to enter a password in order to gain access to a protected resource. The password can be said to be “stored” in the user’s memory; the system controlling the access to the resource must not store the password itself, but only enough information to validate the password - with no way of deriving the password data back from the stored information. A salted hash can be used to transform the password to a value which can be stored safely, while not allowing the reverse operation.

However, password-based authentication can also take place between software components communicating with each other. In this case, the password must be stored by the component authenticating (playing the role of the “user”), being possible to retrieve it when needed. A secure, encrypted storage is an absolute requirement for such sensitive data.

Gateway features a secure storage mechanism for passwords and other sensitive data (like encryption keys). The encryption is performed with a strong symmetric algorithm (AES 256 CBC), using a PBES2 encryption scheme.

Creating an encrypted password storage file set is done using the pelencpass command. See Creating a Secure Password Storage using pelencpass).

Starting from a user input string (for generating the derived key) and the password, 3 files are created:

- A salt file
- A derived key file, containing the key used for securing the password
- An encrypted password file, containing the password encrypted with the derived key

The pelencpass command also allows reusing a previously generated derived key file.

Using the Secure Password

Separate configuration entries for the 3 output files - the salt, derived key and encrypted password files - replaces the single entry for the password. The secure mechanism is implemented for PassPort AM/PassPort PM, SecureRelay and the SWIFTNet High Availability Database connection. Refer to the specific documentation sections for details regarding the new configuration parameters.

The encryption keys must be stored in a secure location, separate from the data they are protecting. Use operating system security features like local server access, Access Control Lists (ACLs), strong passwords and session timeouts to control access to the files containing encryption keys.

Creating a Secure Password Storage using pelencpass

Syntax	pelencpass encrypt_pass {-encrypt_input_string(-encis)} {-password_to_encrypt(-encpwd)} {-salt_file(-saltf)} {-derived_key_file(-dkf)} {-encrypted_data_file(-encf)} [-generate_key(-genkey) { (generate) | reuse }] [-output_format(-fmt) { (binary) | base64 }]
Description	Use this command to generate the encryption key.
Parameters	• password_to_encrypt(-encpwd): Enter the password to be encrypted. • encrypt_input_string(-encis): Enter a user input string to generate the derived key (required when -generate_key is *generate*). • salt_file(-saltf): The full path to the file to store the salt. • derived_key_file(-dkf): The full path to the file to store the derived key (when -generate_key is *generate*), or that contains a previously generated derived key (when -generate_key is *reuse*). • encrypted_data_file(-encf): The full path to the file to store the encrypted password. • generate_key(-genkey): Specify if a new key should be generated (*generate*, default) or if a previously generated key should be used (*reuse*). • output_format(-fmt): Specify the output format.

Once a derived key file is created, it can be reused by subsequent pelencpass encrypt_pass commands, by specifying the -generate_key reuse parameter. If -generate_key is set to *generate* (or not specified), a new derived key file will be created. Note that when creating a new key, any existing content of the derived key file will be overwritten.

The -output_format parameter specifies the output format as *binary* (default) or *base64*. Gateway configuration requires using the binary format encrypted files. The *base64* option converts the encrypted password to a human-readable BASE64 representation, the same as used in Gateway objects export files. For more information, see Secure Password Export.

Login for operations

The password used for authorizing operations on Gateway through commands (command line utility, exists, scripts) when access management is enabled is stored in encrypted form.

To use the online commands with user access security, Gateway uses the following environment variables to define the user login and user password:

User name:

- p_cs_username

Password:

- p_cs_dk_file: path to the key file used for password encryption (produced by pelencpass tool)
- p_cs_encpass_file: path to the encrypted password file (produced by pelencpass tool)

Password files

The encrypted password files are created with a Gateway tool named pelencpass. For information about how to do this, see Creating a Secure Password Storage using pelencpass above.

Secure Password Export

Some Gateway objects can contain passwords which would be revealed, if they were exported in a clear text form. Those objects include Remote Sites, Connection Gates (CGates) and Super CGates, Trading Partners, SSH Security Profiles and TLS Security Profiles.

Gateway offers a mandatory mechanism for encrypting such sensitive information in the export files. This is done by specifying an additional parameter (-encryption_key_file) in the corresponding export command (pelbase, secbase). The password is first encrypted using the specified key, then BASE64-encoded. If the password field is not set, an empty value is used instead.

The encryption key file is an AES 256 symmetric key generated by the same mechanism as the derived key file generated using pelencpass encrypt_pass; however, since only the encryption key file is needed, the syntax is simpler.

Syntax	pelencpass generate_key {-encrypt_input_string(-encis)} {-derived_key_file(-dkf)} [-salt_file(-saltf)]
Description	Use this command to generate an AES 256 encryption key.
Parameters	• encrypt_input_string(-encis): Enter a user input string to generate the derived key. • derived_key_file(-dkf): The full path to file that contains the derived key. • salt_file(-saltf): The full path to file that contains the salt.

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Access Control Management

About Access Control Management

You can manage user Access Control in Gateway using any of the following methods:

Method	Description
Gateway	Users are managed by *User* and *Profile* objects in Gateway. Users must supply their authentication (user ID and password) in their login request. Access control is based on object type. A user with authorized access to a type of object has the same access to all instances of that object type.
Customizable user exits	
PassPort AM	Users are managed by Axway PassPort AM. Access control is managed centrally for all products of the Axway Platform.

These User Access Control methods are mutually exclusive. When one method is selected, you cannot use another one at the same time.

User Access Control of Gateway objects

User Access Control applies to Gateway objects, whatever the interface used: the Gateway GUI, command lines, APIs or the Axway MFT Navigator.

Users and Profile objects

The *User* and *Profile* objects in Gateway enable you to define user identification and access rights. Typically only Administrators have access to these objects.

Users

The *User* object defines properties for each user:

- Name and password
- Group for the user
- Profile for the user (all users must be associated with a user Profile)

Profiles

The *Profile* object defines the type of operations a set of users can perform on each Gateway object. All users associated with the same user Profile have identical access rights. You can define authorization for the following types of operation:

- Read: retrieve or list objects
- Write: create new instance of an object
- Update: modify attributes of an object
- Delete: remove objects
- Admin: suspend, remove or cancel Transfer Requests, enable or disable Sites, suspend, restart or archive the log file

The Profile object specifies the authorized operations on the following objects:

- Applications
- CGates
- Decision Rules
- Diffusion Lists
- Log
- Models
- Profiles
- Proxies
- Sites
- Transfers
- Users
- VFD

For example, you could specify that all Users associated with a given Profile are authorized to access Site objects in Read and Write mode only.

Note: A user only has access to the elements defined in his profile.

Related topics

Working with Users

Working with User Profiles

About Users and Profiles

Access Control using exits

About PassPort AM

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Users and Profiles

Axway Gateway: Managing Security

About Users and Profiles

Users and Profiles

Controlled elements

Access rights

Access to Transfer Requests

This topic describes the *User* and *Profile* objects that Gateway uses to manage Access Control.

Alternatively you can use:

- *PassPort AM* to manage User Access Control for all products of the Axway Platform.
- *Customizable user exits* to manage Access Control on Gateway

Users and Profiles

You define each Gateway **User** by specifying a unique name, an access password and the name of the Profile object with which the User is associated. All Users, apart from the Administrator, must be associated with a specific Profile. All Users associated with the same Profile have identical rights.

Users only have access to their own User records (records bearing their names), unless they are administrators.

For security reasons, you should restrict access to User and Profile objects to administrators only.

The **Profile** object defines the type of access rights that a user, or a set of users, has to specific objects or files. For example, a user may have *read access* for Site objects and *write access* for Application and Model objects.

Controlled elements

The objects and files that are subject to access control are:

Element
Sites
Applications
Transfer Requests
Models
Diffusion Lists
Users
Profiles
Log file
CGates and CGateGroups
Configuration
Audit
Virtual File Directory (VFD) *Used by administrators only for modifications*
Rule Tables
Decision Rules

Element
Proxies
Trading Partners
Transfer security

Access rights

The types of access that you can assign are:

Access	Enables the user to...
Read	Retrieve or list the object
Write	Create a new instance of the object
Update	Modify object attributes
Delete	Remove the object
Admin	Perform all operations: • Suspend, resume or cancel a Transfer • Enable or disable a Site • Suspend, restart or archive the log file

Since, logically, any update or delete action is always preceded by a read access, users with update or delete access rights must also have read access to an object or file.

Access to Transfer Requests

When users create Transfer Requests, they also need access to information on Site and Application objects. Therefore, a user with write (create) access to Transfer Request objects, must, as a minimum, have *read* access to both Site and Application objects.

Typically, users only have access to Transfer Requests that they created. To grant user access to all Transfer Requests, you must assign the *access all transfers* privilege.

Related topics

About Access Control Management

Working with Users

Working with User Profiles

Working with Users (command line)

Working with Profiles (command line)

Access Control using exits

About PassPort AM

Working with PassPort AM

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Working with Users

Axway Gateway: Managing Security

Creating a new User object

Displaying all User objects

Displaying a list of selected User objects

Viewing User object attributes

Modifying a User object

Deleting a User object

Creating a new User object

To create a new User object:

1. In the left pane of the GUI main window, click ⊞ to expand the nodes:

Security Management > Access Control Management

2. Right-click the **User** sub-node, and then select **New** from the context menu.

Gateway displays the *New User* window.

3. In the *New User* window, complete the fields described in the following table:

Field	Meaning
Name (*mandatory*)	Enter a unique name that identifies the user to Gateway. For platforms that support multi-users, the user name should be the same as that used in the operating system. Maximum: 31 characters.
Group	*Optional* Enter the Group name for this User. If you do not specify a Group, the default Group GDEFAULT is used.
Comments	Free-text field.

Field	Meaning
	Maximum: 80 alphanumeric characters.
Access control:	
Set password	Opens a dialog box to enter a user password. If you do not set a user password, a new password will be required at login time. Maximum: 15 characters.
User profile	Enter the name of the Profile object that defines the access rights (authorized actions) for a user or a set of users. This record is stored in the Profile database. You must specify an access Profile for all users other than administrators. Alternatively, click the down arrow to display a list of available Profiles, and then select a Profile from the list.
Expiration date	*Optional* Select an expiration date for user access.
Maximum retries	*Optional* Enter an integer that represents the upper limit for consecutive unsuccessful login attempts. Beyond this limit, the user account is disabled, and administrator intervention is required to re-enable it. A value of zero stands for no limit checking.
Disabled	Check this option to disable the user account.
Additional rights:	
Administrator	Check this option to assign administrator access to the User. Administrators have access to all operations. If you assign Administrator access, the associated Profile definition is ignored.
Access all transfers	Each Transfer Request is associated with its creator, also known as owner. Access to Transfer Requests is limited to: • Transfer Request creator • Administrators • Users with *Access all transfers* privilege Check this option to assign access to all Transfers. The actions the User can perform on Transfer Requests are still limited to those defined in the associated Profile object.
Allow old versions	Allow login from this user when connecting with an older API version. Useful especially for XIB users, where the API update is not in sync with the Gateway release.

4. Click **OK** to save the new user definition and close the window.

Note:

Define overall User Access Control in the Gateway configuration menu (Gateway > Connection control).

If you want to use script or batch functions, you must define environment variables for the following:

- an administration account login (p_cs_username)
- an administration decryption key file (p_cs_dk_file)
- an administration encrypted password key file (p_cs_encpass_file)

(Storing the cleartext password in the environment variable p_cs_password is no longer supported.)

For more information regarding password encryption, read Password management.

Displaying all User objects

To display a list of all User objects stored in the database:

1. In the left pane of the GUI main window, click ⊞ to expand the nodes:

Security Management > Access Control Management

2. Click the **User** sub-node.

Gateway displays a list of all existing User objects in the right (display) pane.

Displaying a list of selected User objects

You can reduce the number of User objects listed in the display pane by applying selection criteria to the display.

To display a selected list of User objects:

1. In the left pane of the GUI main window, click ⊞ to expand the nodes:

Security Management > Access Control Management

2. Click the **User** sub-node.

Gateway displays a list of all existing User objects in the right pane.

3. Right-click the **User** sub-node, then select **Select...** from the context menu.

Gateway displays the *Select User* window.

4. In the *Select User* window, use the data fields to enter criteria for User display selection. You can use one or more of the following User attributes as selection criteria:
    - Maximum number of lists (default = 50)
    - User Name
    - Group
    - Comments
    - User profile
    - Disable
    - Administrator
    - Access all transfers
5. Click **OK** to accept the display selection criteria.

Gateway regenerates the right window User list display, taking into account your selection criteria.

Viewing User object attributes

To view the attributes of a specific User object:

1. In the left pane of the GUI main window, click ⊞ to expand the nodes:

Security Management > Access Control Management

2. Click the **User** sub-node.

Gateway displays a list of all existing User objects in the right (display) pane.

3. In the right pane, right-click the name of the User object you want to view. Select **View...** from the context menu.

Gateway displays the *User (read only)* window. This window provides you with a view of all attributes for the selected User object.

4. When you have finished viewing the information, click **Close** to quit the window.

Modifying a User object

To modify the attributes of an existing User object:

1. In the left pane of the GUI main window, click ⊞ to expand the nodes:

Security Management > Access Control Management

2. Click the **User** sub-node.

Gateway displays a list of all existing User objects in the right (display) pane.

3. In the display frame (right frame), right-click the User object that you want to modify. Then in the context menu click **Modify...**

Gateway opens the *User* editing window. This window comprises two tabs.

4. Modify the value fields of the tabs according to your requirements.
5. Click **OK** to confirm changes and complete the modification procedure.

Deleting a User object

To delete an existing User object:

1. In the left pane of the GUI main window, click ⊞ to expand the **Partner Management** node.
2. Click the **User** sub-node.

Gateway displays a list of all existing User objects in the right (display) pane.

3. In the right pane, right-click the name of the User object you want to delete. Select **Delete...** from the context menu.

Gateway displays a confirmation dialog box.

4. In the confirmation dialog box, click **Yes** to confirm the delete operation.
   Alternatively, click **No** or **Cancel** to abandon the operation.

Gateway deletes the User object from the database and removes the entry from the display window.

Related topics

About Access Control Management

Configuration: Gateway parameters

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Working with User Profiles

Axway Gateway: Managing Security

Creating a new User Profile object

Displaying all User Profile objects

Displaying a list of selected User Profile objects

Viewing User Profile object attributes

Modifying a User Profile object

Deleting a User Profile object

Creating a new User Profile object

To create a new User Profile object:

1. In the left pane of the GUI main window, click ⊞ to expand the nodes:
   Security Management > Access Control Management
2. Right-click the **Profile** sub-node, and then select **New** from the context menu.
   Gateway displays the *New Profile* window.
3. In the *New Profile* window:
   - Enter a unique name for the Profile in the **Name** field. You can enter up to 31 alphanumeric characters.
   - Optionally, enter a free-text description in the **Comments** field. You can enter up to 80 alphanumeric characters.
   - Complete the object/access matrix. To grant complete access to all objects, click the **Select All** button. Alternatively, check individual boxes.
4. Click **OK** to save the definition and close the window.

Displaying all User Profile objects

To display a list of all User Profile objects stored in the database:

1. In the left pane of the GUI main window, click ⊞ to expand the nodes:
   Security Management > Access Control Management
2. Click the **Profile** sub-node.
   Gateway displays a list of all existing User Profile objects in the right (display) pane.

Displaying a list of selected User Profile objects

You can reduce the number of User Profile objects listed in the display pane by applying selection criteria to the display.

To display a selected list of User Profile objects:

1. In the left pane of the GUI main window, click ⊞ to expand the nodes:
   Security Management > Access Control Management
2. Click the **Profile** sub-node.
   Gateway displays a list of all existing User Profile objects in the right pane.
3. Right-click the **Profile** sub-node, then select **Select...** from the context menu.
   Gateway displays the *Select Profile* window.
4. In the *Select Profile* window, use the data fields to enter criteria for User Profile display selection. You can use one or more of the following User Profile attributes as selection criteria:
   - Maximum number of lists (default = 50)
   - Profile user name
   - Comments
   - Active
5. Click **OK** to accept the display selection criteria.
   Gateway regenerates the right window User Profile list display, taking into account your selection criteria.

Viewing User Profile object attributes

To view the attributes of a specific User Profile object:

1. In the left pane of the GUI main window, click ⊞ to expand the nodes:
   Security Management > Access Control Management
2. Click the **Profile** sub-node.
   Gateway displays a list of all existing User Profile objects in the right (display) pane.
3. In the right pane, right-click the name of the User Profile object you want to view. Select **View...** from the context menu.
   Gateway displays the *User Profile (read only)* window. This window provides you with a view of all attributes for the selected User Profile object.
4. When you have finished viewing the information, click **Close** to quit the window.

Modifying a User Profile object

To modify the attributes of an existing User Profile object:

1. In the left pane of the GUI main window, click ⊞ to expand the nodes:
   Security Management > Access Control Management
2. Click the **Profile** sub-node.
   Gateway displays a list of all existing User Profile objects in the right (display) pane.
3. In the display frame (right frame), right-click the User Profile object that you want to modify. Then in the context menu click **Modify...**
   Gateway opens the *User Profile* editing window.
4. Modify the Profile rights according to your requirements.
5. Click **OK** to confirm changes and complete the modification procedure.

Deleting a User Profile object

To delete an existing User Profile object:

1. In the left pane of the GUI main window, click ⊞ to expand the **Partner Management** node.
2. Click the **Profile** sub-node.
   Gateway displays a list of all existing User Profile objects in the right (display) pane.
3. In the right pane, right-click the name of the User Profile object you want to delete. Select **Delete...** from the context menu.
   Gateway displays a confirmation dialog box.
4. In the confirmation dialog box, click **Yes** to confirm the delete operation.
   Alternatively, click **No** or **Cancel** to abandon the operation.
   Gateway deletes the User Profile object from the database and removes the entry from the display window.

Related topics

About Access Control Management

Configuration: Gateway parameters

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Command line operations

Axway Gateway: Managing Security

peladm create_user

peladm delete_user

peladm update_user

peldsp display_user

peldsp select_user

Creating a User: peladm create_user

Syntax	**peladm create_user (**-username**) [***-optional parameters, see below***]**
Description	*This command is reserved for administrators only.* Use this command to create a new user. All parameters apart from username are optional. The Profile that you specify must be defined in the Profile database.
Parameters	*Mandatory* -username (-un): Login name for this user
	-profile (-pr): Access profile name for this user
	-group (-gr): Group name for this user
	-password (-pswd): Password for this user
	-comments (-cts): Enter a free-text description of up to 80 alphanumeric characters
	-expired (-exd): Enter an expiration date in the format *YYYYDDMM HHMMSS*
	-maxretry (-mxr): Enter a number between 0 and 999 for the maximum number of connection retries allowed (default = 0)
	-administrator (-admin): Define the user as an administrator or not. Enter one of the values: • **N**: non-administrator users • **Y**: administrator users (default)
	-allxfer (-ax): Defines the user with/without access to all transfers. Enter one of the values: • **N**: non-administrator users (default) • **Y**: administrator users
	-disabled (-dd): Enables/disables User. Enter one of the values: • **N**: non-administrator users (default) • **Y**: administrator users
	-allow_old_ver (-aov) : Allow login from this user when connecting with an older API version. Useful especially for XIB users, where the API update is not in sync with Gateway release. Enter one of the values: • **N**: don't allow • **Y**: allow
Example	Enter the command:

Syntax	**peladm create_user (**-username**) [***-optional parameters, see below***]**
	peladm create_user -username USERONE -password passone -comments "Gateway administrator" -group admin -administrator Y Gateway creates a new User with the name *USERONE*, the password *passone*, who belongs to the Group *admin* and has administrator rights.

Note:

Overall user access to Gateway is controlled by the **User access control** parameter in the **Configuration** menu. If you enable this option, you must edit the userprof.bat script (Windows) or the osprofile (UNIX) file, or modify exitlgin.c, to set the following environment variables:

- administration user: p_cs_username
- decryption key file pathname: p_cs_dk_file
- encrypted password file pathname: p_cs_encpass_file

These environment variables must already be defined in an administration account.

(Storing the cleartext password in the environment variable p_cs_password is no longer supported.)

For more information regarding password encryption, read Password management.

Modifying a User: peladm update_user

Syntax	**peladm update_user (**-username**) [***-optional parameters, see below***]**
Description	*This command is reserved for administrators only.* Use this command to modify a User definition. For non-administrator users, the only modifiable parameter is password, which enables users to set their own passwords as needed. All parameters apart from username are optional. If not specified, a parameter keeps its previous value. The Profile that you specify must be defined in the Profile database.
Parameters	*Mandatory* -username (-un): Login name for this user.
	-profile (-pr): Access profile name for this user.
	-group (-gr): Group name for this user.
	-password (-pswd): Password for this user.
	-comments (-cts): Enter a free-text description of up to 80 alphanumeric characters.

Syntax	**peladm update_user (**-username**) [***-optional parameters, see below***]**
	-expired (-exd): Enter an expiration date in the format *YYYYDDMM HHMMSS*.
	-maxretry (-mxr): Enter a number between 0 and 999 for the maximum number of connection retries allowed (default = 0).
	-administrator (-admin): Define the user as an administrator or not. Enter one of the values: • **N**: non-administrator users • **Y**: administrator users (default)
	-allxfer (-ax): Defines the user with/without access to all transfers. Enter one of the values: • **N**: non-administrator users (default) • **Y**: administrator users
	-disabled (-dd): Enables/disables User. Enter one of the values: • **N**: non-administrator users (default) • **Y**: administrator users
	-allow_old_ver (-aov) : Allow login from this user when connecting with an older API version. Useful especially for XIB users, where the API update is not in sync with Gateway release. Enter one of the values: • **N**: don't allow • **Y**: allow
Example	Enter the command: peladm update_user -username USERONE -password PASS Gateway sets the password of the User *USERONE* to the value *PASS*.

Deleting a User: peladm delete_user

Syntax	**peladm delete_user [**-username**]**
Description	Use this command to delete a User definition. Repeat the command for each User that you want to delete.
Parameters	-username (-un): Name of the user you want to delete
Example	Enter the command: peladm delete_user -username USERONE Gateway deletes the User *USERONE*.

Selecting Users: peldsp select_user

Syntax	**peldsp select_user (-**username**) [***-optional selection parameters, see below***]**
Description	Use this command to display a subset of Users that matches a defined selection criteria. To list all Users, enter the command without any selection parameters.
Parameters	-username (-un): Login name for this user.
	-profile (-pr): Access profile name for this user.
	-group (-gr): Group name for this user.
	-comments (-cts): Enter a free-text description of up to 80 alphanumeric characters.
	-administrator (-admin): Define the user as an administrator or not. Enter one of the values: • N: non-administrator users • Y: administrator users
	-allxfer (-ax): Defines the user with/without access to all transfers. Enter one of the values: • N: non-administrator users • Y: administrator users
	-disabled (-dd): Enables/disables User. Enter one of the values: • N: non-administrator users • Y: administrator users
Example	Enter the command: peldsp select_user -group MIS Gateway selects all Users who belong to the group MIS for display.

Displaying a User: peldsp display_user

Syntax	**peldsp display_user (-**username**)**
Description	Use this command to display the full definition of a User that you identify by name. Each parameter is displayed on a separate line, in the format: name_of_field="value"
Parameters	-username (-un): Name of the user for whom you want to display the definition.
Example	Enter the command: peldsp display_user -username USERA

Syntax	**peldsp display_user (-**username**)**
	Gateway displays all attributes of user record USERA: u_username='USERA' u_group=admin u_profile='admin' u_password='pass1' u_comments='Gateway administrator' u_maxretry=0 u_expired='' u_admin='Y' u_allxfer='Y' u_disabled='N'

Related topics

Managing profiles (command line)

About Access Control Management

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Axway Gateway: Managing Security

peladm create_profile

peladm update_profile

peladm delete_profile

peldsp select_profile

peldsp display_profile

Creating a Profile: peladm create_profile

Syntax	**peladm create_profile (-profile)** [optional parameters, see below]
Description	*This command reserved for administrators only.* Use this command to create a new access Profile. Each access parameter can contain any combination of the letters **R**, **W**, **U**, **D** and **A** as described below. If an access type letter is specified, that type of access is granted. Otherwise it is denied. All parameters apart from profile are optional.
Parameters	*Mandatory*

Syntax	**peladm create_profile (-profile)** [optional parameters, see below]
	-profile (-pr): Enter an access profile name of up to 16 alphanumeric characters.
	-comments (-cts): Enter a free-text description of up to 80 alphanumeric characters.
	-site_acc (-site): Specify access rights to Site objects. Enter one or more of the following values: • R = read • W = write • U = update • D = delete • A = administer
	-appli_acc (-appli): Specify access rights to Application objects. Enter one or more of the following values: • R = read • W = write • U = update • D = delete • A = administer
	-xfer_acc (-xfer): Specify access rights to Transfer Requests. Enter one or more of the following values: • R = read • W = write • U = update • D = delete • A = administer
	-log_acc (-log):
	-list_acc (-list): Enter a number between 0 and 999 for the maximum number of connection retries allowed (default = 0).
	-model_acc (-model): Specify access rights to Model objects. Enter one or more of the following values: • R = read • W = write • U = update • D = delete

Syntax	**peladm create_profile (-profile)** [optional parameters, see below]

- A = administer

-cgate_acc (-cgate): Specify access rights to CGate objects.
Enter one or more of the following values:

- R = read
- W = write
- U = update
- D = delete
- A = administer

-vfd_acc (-vfd): Specify access rights to Virtual File Directories objects.
Enter one or more of the following values:

- R = read
- W = write
- U = update
- D = delete
- A = administer

-decrule_acc (-decisionrule): Specify access rights to Decision Rule objects.
Enter one or more of the following values:

- R = read
- W = write
- U = update
- D = delete
- A = administer

-ruletab_acc (-ruletable): Specify access rights to Rule Table objects.
Enter one or more of the following values:

- R = read
- W = write
- U = update
- D = delete
- A = administer

-user_acc (-user): Specify access rights to User objects.
Enter one or more of the following values:

Syntax	**peladm create_profile (-profile)** [optional parameters, see below]

- R = read
- W = write
- U = update
- D = delete
- A = administer

-prof_acc (-prof): Specify access rights to Profile objects.
Enter one or more of the following values:

- R = read
- W = write
- U = update
- D = delete
- A = administer

-proxy_acc (-proxy): Specify access rights to Proxy objects.
Enter one or more of the following values:

- R = read
- W = write
- U = update
- D = delete
- A = administer

-secs_acc (-secs): Specify access rights to Security objects.
Enter one or more of the following values:

- R = read
- W = write
- U = update
- D = delete
- A = administer

-trade_acc (-trade): Specify access rights to Trading Partner objects.
Enter one or more of the following values:

- R = read
- W = write
- U = update

Syntax	**peladm create_profile (-profile)** [optional parameters, see below]

- D = delete
- A = administer

`-audit_acc (-audit)`: Specify access rights to Audit objects. Enter one or more of the following values:

- **R** = read
- **A** = administer

`-config_acc (-config)`: Specify access rights to Configuration objects. Enter one or more of the following values:

- **R** = read
- **U** = update

Example

Enter the command:

peladm create_profile -profile finance
-comments "financial dept"
-site 'R'
-appli 'R'
-xfer 'RWU'
-log 'R'
-list 'R
-model 'R'

Gateway creates a Profile called *finance*. All users associated with this Profile have *read* access to Sites, Applications, Transfers, Log, Diffusion Lists and Models. In addition, they have *write* and *update* access to Transfer Requests.

Modifying a Profile: peladm update_profile

Syntax	**peladm update_profile (-profile)** [optional parameters, see below]
Description	*This command is reserved for administrators only.* Use this command to modify an existing access Profile definition. All parameters apart from profile are optional. If not specified, a parameter retains its previous value. Each access parameter can contain any combination of the letters **R**, **W**, **U**, **D** and **A** as described below. If an access type letter is specified, that type of access is granted. Otherwise it is denied.
Parameters	*Mandatory* -profile (-pr): Enter the name of an existing profile.
	-comments (-cts): Enter a free-text description of up to 80 alphanumeric characters.

Syntax	**peladm update_profile (-profile)** [optional parameters, see below]
	-site_acc (-site): Specify access rights to Site objects. Enter one or more of the following values: • R = read • W = write • U = update • D = delete • A = administer
	-appli_acc (-appli): Specify access rights to Application objects. Enter one or more of the following values: • R = read • W = write • U = update • D = delete • A = administer
	-xfer_acc (-xfer): Specify access rights to Transfer Requests. Enter one or more of the following values: • R = read • W = write • U = update • D = delete • A = administer
	-log_acc (-log):
	-list_acc (-list): Enter a number between 0 and 999 for the maximum number of connection retries allowed (default = 0).
	-model_acc (-model): Specify access rights to Model objects. Enter one or more of the following values: • R = read • W = write • U = update • D = delete • A = administer
	-cgate_acc (-cgate): Specify access rights to CGate objects. Enter one or more of the following values:

Syntax	**peladm update_profile (-profile)** [optional parameters, see below]

- R = read
- W = write
- U = update
- D = delete
- A = administer

-vfd_acc (-vfd): Specify access rights to Virtual File Directories objects.
Enter one or more of the following values:

- R = read
- W = write
- U = update
- D = delete
- A = administer

-decrule_acc (-decisionrule): Specify access rights to Decision Rule objects.
Enter one or more of the following values:

- R = read
- W = write
- U = update
- D = delete
- A = administer

-ruletab_acc (-ruletable): Specify access rights to Rule Table objects.
Enter one or more of the following values:

- R = read
- W = write
- U = update
- D = delete
- A = administer

-user_acc (-user): Specify access rights to User objects.
Enter one or more of the following values:

- R = read
- W = write
- U = update

Syntax	**peladm update_profile (-profile)** [optional parameters, see below]

- D = delete
- A = administer

-prof_acc (-prof): Specify access rights to Profile objects.
Enter one or more of the following values:

- R = read
- W = write
- U = update
- D = delete
- A = administer

-proxy_acc (-proxy): Specify access rights to Proxy objects.
Enter one or more of the following values:

- R = read
- W = write
- U = update
- D = delete
- A = administer

-secs_acc (-secs): Specify access rights to Security objects.
Enter one or more of the following values:

- R = read
- W = write
- U = update
- D = delete
- A = administer

-trade_acc (-trade): Specify access rights to Trading Partner objects.
Enter one or more of the following values:

- R = read
- W = write
- U = update
- D = delete
- A = administer

-audit_acc (-audit): Specify access rights to Audit objects. Enter one or more

Syntax	**peladm update_profile (-profile)** [optional parameters, see below]
	of the following values: • **R** = read • **A** = administer
	-config_acc (-config): Specify access rights to Configuration objects. Enter one or more of the following values: • **R** = read • **U** = update
Example	Enter the command: peladm update_profile -profile finance -user 'RU' Gateway updates the Profile *finance* to assign *read* and *update* rights to all its users.

Deleting a Profile: peladm delete_profile

Syntax	**peladm delete_profile [-profile]**
Description	Use this command to delete a Profile definition. Repeat the command for each User that you want to delete.
Parameters	-profile (-pr): Name of the Profile you want to delete.
Example	Enter the command: peladm delete_profile -profile FINANCE Gateway deletes the Profile *FINANCE.*

Selecting Profiles: peldsp select_profile

Syntax	**peldsp select_profile (-profile)** [optional selection parameters, see below]
Description	Use this command to display all profiles, or subset of profiles that matches a defined selection criteria. To list all profiles, type the command without any selection parameters.
Parameters	-profile (-pr): Login name for this user.
	-comments (-cts): Uses comments as selection criteria.
	-active (-a): Use active or inactive state as selection criteria. Enter one of the values: • N: non-administrator users • Y: administrator users

Syntax	**peldsp select_profile (-profile)** [optional selection parameters, see below]
Example	Enter the command: **peldsp select_profile** Gateway displays all existing profiles.

Displaying a Profile: peldsp display_profile

Syntax	**peldsp display_profile (-profile)**
Description	Use this command to display the full definition of a Profile that you identify by name. The Profile parameters are displayed by return. Each parameter is displayed on a separate line, in the following format: name_of_field="value"
Parameters	-profile (-pr): Name of the profile for which you want to display the definition.
Example	Enter the command: **peldsp display_profile -profile ADMIN** Gateway returns: _profile='admin' p_comments='Gateway administrator' p_count='2' p_site_acc='RWUDA' p_appli_acc='RWUD' p_xfer_acc='RWUDA' p_log_acc='RA' p_list_acc='RWUD' p_model_acc='RWUD' p_cgate_acc='RWUDA' p_vfd_acc='R' p_decrule_acc='RWUDA' p_ruletab_acc='RWUDA' p_user_acc='RWUD' p_prof_acc='R' p_proxy_acc='RWUDA' p_conn_acc='RA' p_cous_acc='RA' p_trade_acc='RWUD' p_secs_acc='RWUD' p_audit_acc='RA' p_config_acc='RU'

Related topics

About Access Control Management

Configuration: Gateway parameters

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

PassPort AM

About PassPort AM

Axway PassPort AM centralizes the management of users for all products of the Axway Platform platform. If you have installed PassPort AM, you can use it to define all your users and profiles. Gateway can then connect to PassPort AM to check user permissions.

PassPort AM provides control of:

- Authentication
  The authentication functionality handles the identification of a user via login and password.
- Access control management
  The access control management functionality restricts access to Gateway resources.

In PassPort AM, users are given role-based access and are assigned roles that define the privileges they have.

The use of PassPort AM is not mandatory. Like many other Axway products, Gateway also has its own method of managing users, based on users and profiles. When PassPort AM is selected, you cannot use Gateway's own user management (via the User and Profile objects). The two methods are exclusive.

To use this feature, select PassPort AM Access Control Management when installing or configuring Gateway.

There are some differences between using Gateway and PassPort AM for managing user access. For example, in Gateway, the Admin access right on a Transfer object enables the user to perform all Pause, Resume and Cancel actions. In PassPort AM however, separate actions exist: Pause, Resume and Cancel.

Component Security Descriptor (CSD) file

For each Axway Platform component, PassPort AM uses an XML-format Component Security Descriptor (CSD) file. Each CSD file contains access definitions relating to a specific Axway Platform component.

The CSD file for Gateway contains information about:

- Component identification
- Resources (Gateway objects - for example SITE)
- Possible actions related to each resource (for example, view or create)

The CSD file for Gateway is provided with PassPort AM. After installing Gateway and PassPort AM, the CSD file must be imported.

Resources defined in the Gateway CSD file

Resource
SITE
TRANSFER
APPLICATION
TRADING PARTNER
CGATE
CGATEGROUP
LOGICAL SITE
DIFFUSION LIST
PROXY
MODEL
VFD
DECISION RULE
RULE TABLE
CONNECTION
CONNECTED USER
LOG
CERTIFICATE
SPROF
SSHPROF
KEY
NETPROF

Resource
CONFIGURATION
AUDIT
LOGIN

Actions defined in the Gateway CSD file

Action	Comment
VIEW	
CREATE	
EDIT	
DELETE	
ACTIVATE	For Sites
DEACTIVATE	For Sites
MOUNT	For VFDs
UNMOUNT	For VFDs
PAUSE	For Transfers & Logs
RESUME	For Transfers & Logs
CANCEL	For Transfers
DISCONNECT	For Connections and Connected Users
RESET	For Connected Users
ARCHIVE	For Logs
LOGIN_WITH_OLDER_API	For LOGIN resource - allow login to Gateway server using older client version. Used for the LOGIN_XIB predefined privilege.

Note: The actions available depend on the resource.

Related topics

[About Access Control Management]

[Working with PassPort AM]

For more information

For information about PassPort AM, refer to the Axway PassPort AM documentation.

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Working with PassPort AM

Axway Gateway: Managing Security

Selecting PassPort AM Access Control Management

You can select PassPort AM as the method to use for Access Control Management when installing or when configuring Gateway.

When PassPort AM is selected, you cannot use Gateway's own user management (via the User and Profile objects). The two methods are exclusive.

Modifying the Access Control Management method

If the Access Control Management method is currently *None*, you can modify it via the Gateway configuration menu.

After setting the Access Control Management method to a secure user access (using Passport AM or Gateway), you can no longer change the option unless you have administrator rights in Gateway or the privilege to update the configuration resource in Passport AM.

Before using PassPort AM with Gateway

Before using PassPort AM Access Control Management in Gateway:

1. Install PassPort AM.

2. Start PassPort AM.
3. Import the CSD file into the PassPort AM repository.
4. Install Gateway.
5. Update the PassPort AM repository with the installed Gateway instance name.

For information about PassPort AM, refer to the Axway PassPort AM documentation.

For information about installing Gateway, refer to the *Gateway Installation Guide*.

Configuring Gateway for PassPort AM

In the Gateway configuration menu:

1. Select Gateway > Connection control.
2. In **User access control**, select *By PassPort AM*.
3. Select Connectivity > Axway Connectivity > PassPort.
4. Enter the following information to configure the PassPort AM connection:
    - The PassPort version (v4)
    - Select the **Use access management** option
    - The PassPort server host name or IP address
    - The PassPort HTTP port
    - The Component Id
    - The login user name
      Note: you need to set the component password by running the `pelencpass` command locally on the Gateway machine.
    - You can also choose to use **TLS**

Note: If you log in with the username admin and password admin into Passport AM, you can leave the password field blank.

The lifetime of the cache is available for user configuration through the general configuration parameter [am] cache_duration. It accepts integer values, expressed in minutes, range from *0* to *120*, and it defaults to *1*.

Calling the PassPort AM server

Gateway calls the PassPort AM server for:

- Authentication
- Access management

A system cache is used to optimize the data flow between Gateway and PassPort AM.

Related topics

About PassPort AM

About Access Control Management

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Access Control using exits

Axway Gateway: Managing Security

Access control mechanisms

Customizing user Access control

Troubleshooting user access problems

Access control mechanisms

This section provides basic descriptions of access control mechanisms within Gateway:

- *Login validation*
- *Object access validation*
- *File access validation*
- *Logout validation*

Login validation

Any access to Gateway, whether an interactive operator session or a batch command session, is preceded by a login request where users supply their login names and passwords.

The user login process involves the following steps.

1. Gateway calls the exit routine ExitLoginParam() to ask for login username and password.
2. From the Gateway Navigator, the user enters the user name and password.
3. When Gateway receives the login request, it calls the exit ExitCheckUserLogin() to validate the request. The exit routine returns one of three values: *Allowed*, *Denied* or *Undefined*. If the exit routine returns *Undefined*, Gateway continues the validation with its own user database definition. The request is accepted if, and only if, all of the following conditions are met:
    - The user name is defined
    - The password is not empty and matches the one defined in the User record (the process of matching user name and password is not case-sensitive)
    - The user is *not* disabled and the expiration date is *not* reached

If validation fails, Gateway rejects the user access request. If the user login request is rejected and the maximum consecutive retry count is reached, Gateway disables the user account. An administrator must then re-enable the account.

If the password is an empty string, a new password is required. You must enter a new password at login before you can proceed.

Object access validation

The following Gateway objects (or files) are protected objects:

- Site
- Application
- Transfer Request
- Model
- Diffusion List
- User
- Profile
- Log file
- CGate and CGateGroup
- VFD
- Rule Table
- Decision Rule
- Proxy
- Trading Partner
- Transfer security
- Configuration
- Audit

A protected object is protected from user access for any of the following actions:

- Read
- Write
- Update
- Delete
- Admin

When a user attempts to access a protected object for any of the above actions, Gateway calls the exit routine ExitCheckUserAccessByObject() to validate the request. The exit routine returns one of three values: *Allowed*, *Denied* or *Undefined*. If the exit routine returns *Undefined*, (default action in the supplied sample codes), Gateway continues the validation with its own user database definition.

Gateway accepts the request if, and only if, at least one of the following conditions is met:

- The user is an administrator
- The associated Profile definition grants to the user the required access to the object in question

File access validation

When a user requests a file transfer, Gateway calls the exit routine ExitCheckFileAccess() to validate access to the required file. If the exit routine denies access, the Transfer Request is rejected. Otherwise, it is accepted. By default, in the supplied sample code, access is always accepted.

Logout validation

When a user ends a login session, Gateway calls the exit routine ExitCheckUserLogout(). This exit routine enables you to clean up the environment, if necessary. The return code from the exit routine is ignored.

Customizing user Access control

This section describes the cases in which you customize the access procedure to correspond to your network requirements.

Login parameters

During a login request to Gateway, a user (or a user program) supplies the required login user name and password.

Gateway calls the following user exit routine to validate the user data:

int ExitLoginParam(char *name, int nm_size, char *password, int pw_size, char *newpassword, int npw_size)

This user exit returns the required user name, password and new password in the buffer provided.

Caution **The default action in the supplied sample codes are not secure enough for a production environment! You must customize this routine to meet your precise functional and security needs. Please see the *Security Guide* > *Identified threats and possible mitigations* section for additional information.**

The ExitLoginParam() exit routine is called on the client (navigator) side either in the address space of the operator interface or in the user program. The other exit routines are called on the server side (in the address space of the monitor).

Note To use the online commands with user access security, by default Gateway uses the following environment variables to define the user login and user password:

`p_cs_username`
p_cs_dk_file
p_cs_encpass_file
(Storing the cleartext password in the environment variable p_cs_passwordis is no longer supported.)
For more information regarding password encryption, read Password management.

Transfer Request owner

Every Transfer Request is associated with a creator, also known as an owner. Two cases are possible:

- A user creates a Transfer Request: the owner is set to the login user name.
- Gateway creates the Transfer Request (for example, on reception of a Transfer Request from a Remote Site): the owner is set to the Transfer Request owner name defined in the Remote Site record

If no user name is defined, the owner is set to the **Default user name** that you specify during configuration.

Setting up Access control

User Access control is only effective if users access Gateway via the GUI or via a command line. The confidentiality and integrity of your installation data, as well as Gateway itself, must still be protected by normal system security. Make sure that this data is protected against unauthorized direct access.

To set up User Access control:

1. In the configuration menu, enable the **User access control** option and set the **Default user name** to an

appropriate value.
Refer to Configuration: Gateway parameters: Gateway > Connection control.
If you do not enable this option, access control and exit routines are not activated.

2. The first time you start Gateway, and before you define any users, the installation program creates a default user named ADMIN, with the password ADMIN. By default, this user is the product administrator. The ExitLoginParam() in the supplied sample codes uses this entry by default.
3. Using either the GUI or the **peladm** command, set up your user and access rights profile database. It is recommended that you create an additional administrator besides the default ADMIN. You should also change the default ADMIN password.
4. Edit the ExitLoginParam() routine to satisfy your production environment requirements. The exact procedure used to compile and link exit routines depends on the platform.
5. Update all other optional server-side exit routines if you are connecting Gateway to your system security package.

Troubleshooting user access problems

This section outlines the various cases where access is denied and explores possible causes for this refusal. If you experience a problem not covered in this section, contact your system administrator or technical support staff.

Login access denied

If the login request returns an *access denied* message, check the following:

- Are the specified user name and password correct?
- Were the user name and password overwritten by the ExitLoginParam() routine?
- Is access denied by an ExitCheckUserAccess() routine?
- Is the user not enabled?
- Is the expiration date reached?
- Is a new password required?

Operation access denied

If the login request is successful but an operation is refused, check the following:

- Is the Profile (access rights) field defined in the user record?
- Does the specified Profile (access rights) record exist? If a specific implementation exists, check it.
- Does the Profile record allow the required operation on the object?
- Are you the owner of the requested objects?

User forgets password

Ask a system administrator to assign a new password or to erase the old one (in this second case, a new password is required at the next login).

Administrator forgets password

If you have a backup administrator account, use it to re-assign a new password. Otherwise, disable the security

feature, change the password in question and then re-enable the security feature.

Related topics

About Access Control Management

Configuration: Gateway parameters

Internal user exits

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Transfer Security Management

Axway Gateway: Managing Security

Gateway integrates a set of objects that you can use to manage your transfer security requirements.

- *Certificates and keys*: to authenticate the remote party
- *Network Profiles*: to determine whether or not to associate an incoming connection with a Security Profile (either TLS or SSH)
- *TLS Security Profiles*: to secure a connection with TLS protocol
- *SSH Security Profiles*: to secure a connection with SSH protocol (only used with SFTP)

You can manage these objects using either the GUI or command lines.

This book describes the Gateway functions and components that are specifically devoted to operations security.

This topic introduces Gateway profile processes. This book is divided into sub-books that describe the architecture and management of the Security Server (SECS) module of Gateway and how to create, manage and monitor the objects listed above:

Sub-book	Describes...
Security Server (SECS)	The role of the Security Server in the Gateway architecture, and how to manage it using online commands.
Certificates and keys	The use and management of security certificates and keys.
Network Profiles	The use and management of Network Profiles.
SSH Protocol	Features of the SSH protocol and how to deploy them via Gateway.
TLS Protocol	Features of the TLS protocol and how to deploy them via Gateway.

How Gateway establishes security

Gateway uses **Network Profile** objects to determine whether an incoming connection must be secured. Network Profiles contain matching rules that Gateway applies to the incoming connection parameters (origin address, SAP, and so on). From the Network Profile that best matches the incoming connection, Gateway determines which Security Profile to use that provides all required security elements.

The **Security Profile**, either TLS or SSH, is used to establish security for both incoming and outgoing connections. The Security Profile is a configuration set that determines which security protocol to use, which encryption method, and so on.

Incoming connections

In most cases, when you use TLS/SSL or SSH protocols, the connection is secured as soon as you establish a network connection.

When an incoming connection occurs, Gateway must determine whether to use TLS/SSL or SSH, otherwise data received may be misunderstood.

Incoming connections use Security Profiles twice:

- Once to secure the connection when the connection is established
- Once during connected Site authentication, to check that the parameters assigned to the Remote Site match the established level of security

In an incoming connection, only the process of identification of the Security Profile is common to TLS/SSL and SSH.

Outgoing connections

The Remote Site configuration determines whether or not to secure an outgoing connection.

If you specified a Security Profile in the outgoing **Security Profile** field (tls_sprof_out) of the Site, then this Security Profile is used to secure the connection.

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Security Server

Axway Gateway: Managing Security

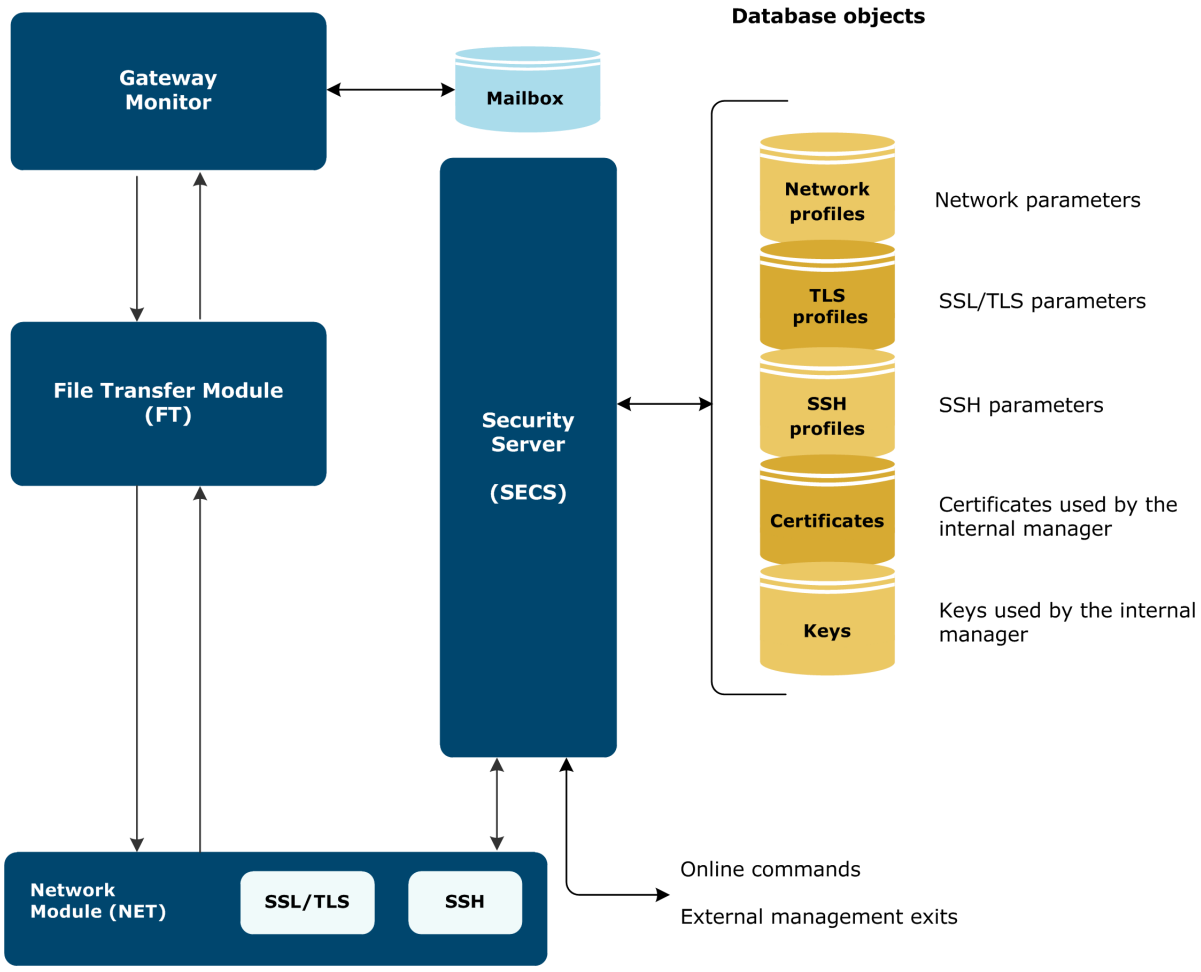
TLS/SSL and SSH protocol implementation is divided into three parts:

- *TLS and SSL protocols*
- *SSH protocol*
- Security server

As shown in the illustration below, TLS/SSL and SSH protocols are implemented in the network entity so that they are transparent for application protocols. The independent **Security server** (or **SECS**) groups the security elements in its database to perform the tasks relating to remote communicating partner mutual authentication (providing/verifying certificates, signing, encrypting/decrypting data). The Security server can operate using an internal manager or external exits.

Security Implementation within Gateway

The following diagram illustrates security implementation within Gateway.



To operate correctly, the Security server requires at least two kinds of object:

- *Network Profiles* that match incoming connections and retrieve the Security Profile to use
- **Security Profiles** that determine security configuration for connections. They can be either:
  - *TLS Profiles*
  - *SSH Profiles*

The **Security Profiles** store your and your partner's authentication information. When you use the internal manager, and depending on the level of security you require, you may need any or all of the following:

- Key exchange algorithms (SSH only)

- Public key algorithms
    - X.509 certificates
    - RSA public key (SSH only)
    - DSA public key (SSH only)
    - ECDSA_NISTP256 public key (SSH only)
    - ECDSA_NISTP384 public key (SSH only)
    - ECDSA_NISTP521 public key (SSH only)
- Encryption algorithms
- MAC algorithms

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Overview of SECS online commands

Axway Gateway: Managing Security

Gateway provides two online commands that enable you to manage all Security server configuration elements (Certificates, Security Profiles and Network Profiles):

- **secadm**: to manage security elements
- **secdsp**: to view and select security elements

Before you execute these commands, confirm that you have:

- Set the global environment variables
- Started Gateway (the **p_secs** process is running)

List of secadm and secdsp commands

Object	secadm commands	secdsp commands
Certificates	import_cert update_cert delete_cert	display_cert select_cert status_cert
Keys	import_key update_key delete_key	display_key select_key status_key
TLS Profiles	import_sprof	display_sprof

Object	secadm commands	secdsp commands
	delete_sprof	select_sprof status_sprof
SSH Profiles	import_sshprof delete_sshprof	display_sshprof select_sshprof status_sshprof
Network Profiles	create_netprof update_netprof delete_netprof	display_netprof select_netprof status_netprof

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Certificates and Keys

Axway Gateway: Managing Security

About certificates and keys

Certificate fields

Distinguished Names

Establishing a trust relationship

Certificate Authority (CA) hierarchy

Certificate chains

Syntax and formats

Related topics

Certificate restrictions

Managing certificates

Managing keys

Managing certificates (command line)

Managing keys (command line)

About certificates and keys

A certificate is an electronic document that:

- Identifies an individual, a server, a company, or some other entity
- Associates that identity with a public key

When you communicate with any remote party, you must first determine the identity of that correspondent, before dealing with data integrity and confidentiality. In other words, forbidding access to data and verifying that the data received has not been modified is meaningless if you are not sure of the identity of the other party. Certificates help prevent the use of fake public keys for impersonation. Only the public key certified by the certificate works with the corresponding private key that belongs to the entity identified by the certificate.

A certificate (more precisely, an X.509 certificate as defined by ISO) is a data set that:

- Provides information about the owner
- Links the owner to a public key

To ensure that this information is correct, a third party referred to as the *Certificate Authority*, or *CA*, signs the whole certificate. In other words, data is hashed and the generated hash is encrypted using an asymmetric algorithm such as DSA or RSA. The CA owns a public/private key pair. It uses its private key to create the certificate signature and makes its public key (required to decrypt the signature) available to anyone through its own certificate.

Certificate fields

Every certificate comprises:

- *Data section*
- *Signature section*

Data section

The **data section** includes the following information:

Field	Description
Version	Version of the encoded certificate value for X.509 V3 is 2.
Serial Number	An integer assigned by the CA. The issuer name and serial number must identify a unique certificate.
Issuer Signature Algorithm	The signature algorithm identifier used by the CA to sign the certificate (for example, RSA with SHA-1).
Issuer Distinguished Name	The DN of the entity that signed and issued the certificate.

Field	Description
Validity Period	The validity period for the certificate.
Subject Distinguished Name	The entity associated with the public key stored in the public key field for the Subject.
Subject Public Key Information	The public key and the identifier algorithm used with the key.
Extensions (*optional*)	Extensions are additional information about the certificate. Extensions comprise three fields: type, criticality, and value. • Type field is the ASN.1 data type (for example, text string, numerical, date). • The criticality flag is either critical or non-critical. If the PKI client application cannot process a critical extension, the certificate is rejected. • The actual value is the value associated with the extension.

Signature section

The **signature section** includes the following information:

- Type of cryptographic algorithm or cipher that the issuing CA uses to create its own digital signature.
- Digital signature for the CA, obtained by hashing all of the data in the certificate and encrypting it with the CA private key.

The following example shows the data and signature sections of a certificate in human-readable format:

Certificate:
Data:
Version: v3 (0x2)
Serial Number: 3 (0x3)
Signature Algorithm: PKCS #1 MD5 With RSA Encryption
Issuer: OU=Sopra Certificate Authority, O=Sopra, C=US
Validity:
Not Before: Fri Oct 17 18:36:25 1997
Not After: Sun Oct 17 18:36:25 1999
Subject: CN=Albert Dupont, OU=Marketing, O=Sopra, C=US
Subject Public Key Info:
Algorithm: PKCS #1 RSA Encryption
Public Key:
Modulus:
00:ca:fa:79:98:8f:19:f8:d7:de:e4:49:80:48:e6:2a:2a:86:
ed:27:40:4d:86:b3:05:c0:01:bb:50:15:c9:de:dc:85:19:22:
43:7d:45:6d:71:4e:17:3d:f0:36:4b:5b:7f:a8:51:a3:a1:00:
98:ce:7f:47:50:2c:93:36:7c:01:6e:cb:89:06:41:72:b5:e9:
73:49:38:76:ef:b6:8f:ac:49:bb:63:0f:9b:ff:16:2a:e3:0e:
9d:3b:af:ce:9a:3e:48:65:de:96:61:d5:0a:11:2a:a2:80:b0:
7d:d8:99:cb:0c:99:34:c9:ab:25:06:a8:31:ad:8c:4b:aa:54:
91:f4:15
Public Exponent: 65537 (0x10001)
Extensions:
Identifier: Certificate Type
Critical: no
Certified Usage:
SSL Client
Identifier: Authority Key Identifier
Critical: no
Key Identifier:
f2:f2:06:59:90:18:47:51:f5:89:33:5a:31:7a:e6:5c:fb:36:
26:c9
Signature:
Algorithm: PKCS #1 MD5 With RSA Encryption
Signature:
6d:23:af:f3:d3:b6:7a:df:90:df:cd:7e:18:6c:01:69:8e:54:65:fc:06:
30:43:34:d1:63:1f:06:7d:c3:40:a8:2a:82:c1:a4:83:2a:fb:2e:8f:fb:
f0:6d:ff:75:a3:78:f7:52:47:46:62:97:1d:d9:c6:11:0a:02:a2:e0:cc:
2a:75:6c:8b:b6:9b:87:00:7d:7c:84:76:79:ba:f8:b4:d2:62:58:c3:c5:
b6:c1:43:ac:63:44:42:fd:af:c8:0f:2f:38:85:6d:d6:59:e8:41:42:a5:
4a:e5:26:38:ff:32:78:a1:38:f1:ed:dc:0d:31:d1:b0:6d:67:e9:46:a8:
dd:c4

Distinguished Names

An X.509 certificate binds a **Distinguished Name (DN)** to a public key. A DN is a series of name-value pairs, such as cn=Albert Dupont, that uniquely identify an entity. In this case, it is referred to as the **certificate subject**.

For example, a typical DN for an Axway employee could be:

e=dupont@axway.com,cn=Albert Dupont ,o=Axway Corp.,c=FR

where:

- *e* = Email address
- *cn* = Common name for this user
- *o* = Organization
- *c* = Country

DNs may include a variety of other name-value pairs, used to identify both certificate subjects and entries in directories that support the Lightweight Directory Access Protocol (LDAP: protocol generally used to access X.500 directories).

As defined by the X.500 standard, a Distinguished Name is a set of Relative Distinguished Names, each of which comprises an Attribute and Value set (**AVA**).

Note: Each Relative Distinguished Name (**RDN**) typically contains only one AVA. If more than one AVA is present, an error occurs when you import the corresponding certificate.

Establishing a trust relationship

Certificate authorities (CAs) are entities that validate identities and issue certificates. They can either be independent third parties or organizations running their own certificate-issuing server software.

Any client or server software that supports certificates maintains a collection of **trusted CA certificates**. These CA certificates determine which other certificates the software can validate, in other words, which issuers of certificates the software can trust. In the simplest case, the software can only validate certificates issued by one of the CAs for which it has a certificate.

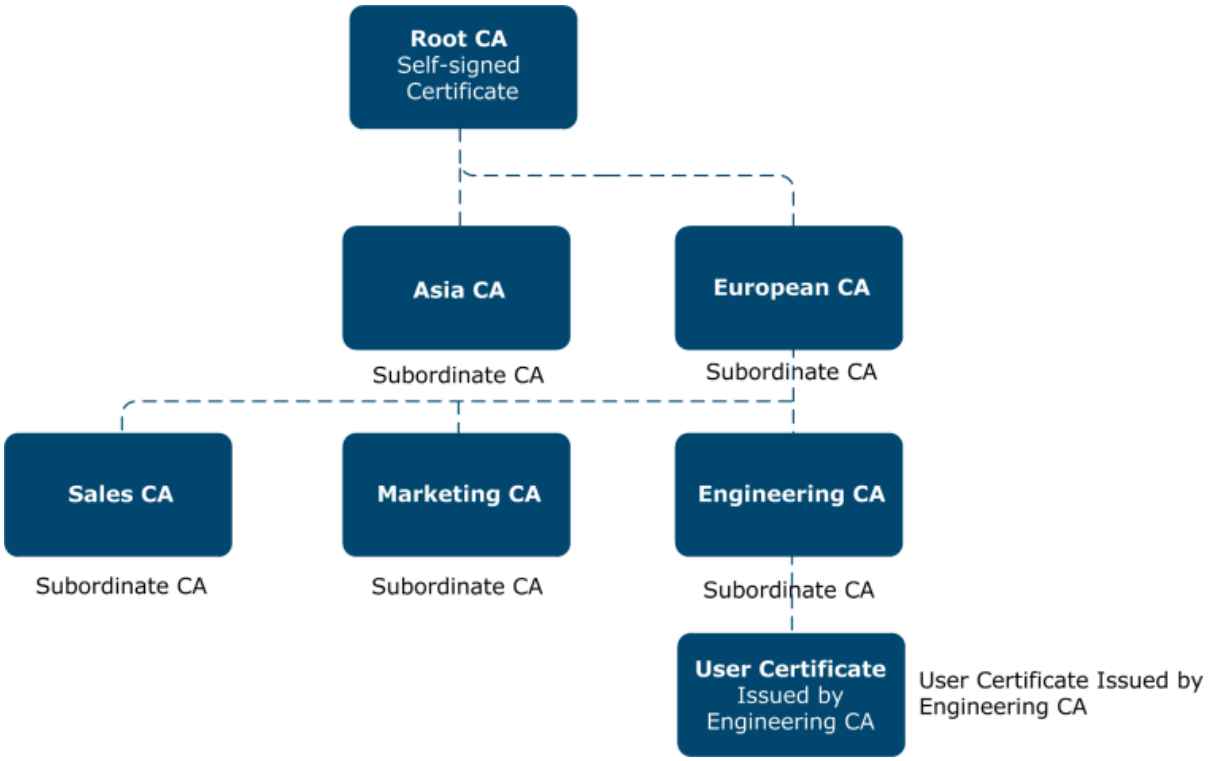
A trusted CA certificate can also be part of a chain of CA certificates, each one issued by the CA above it in a certificate hierarchy.

Certificate Authority (CA) hierarchy

In large organizations, it may be appropriate to delegate the responsibility for issuing certificates to several different certificate authorities. For example, the number of certificates required may be too large for a single CA to maintain. Different organizational units may have different policy requirements, or it may be important for a CA to be physically located in the same geographic area as the people to whom it is issuing certificates.

You can delegate certificate-issuing responsibilities to subordinate CAs. The X.509 standard includes a model for setting up a CA hierarchy, as illustrated in the following diagram:

CA hierarchy



In this model, the root CA is at the top of the hierarchy. The certificate of the root CA is a **self-signed certificate**. That is, the certificate is digitally signed by the same entity, the root CA, that the certificate identifies. The CAs that are directly subordinate to the root CA have CA certificates signed by the root CA. CAs under the subordinate CAs in the hierarchy have their CA certificates signed by the higher-level subordinate CAs.

Organizations have a great deal of flexibility in terms of the way they set up their CA hierarchies. The above illustration shows just one example, but many other arrangements are possible.

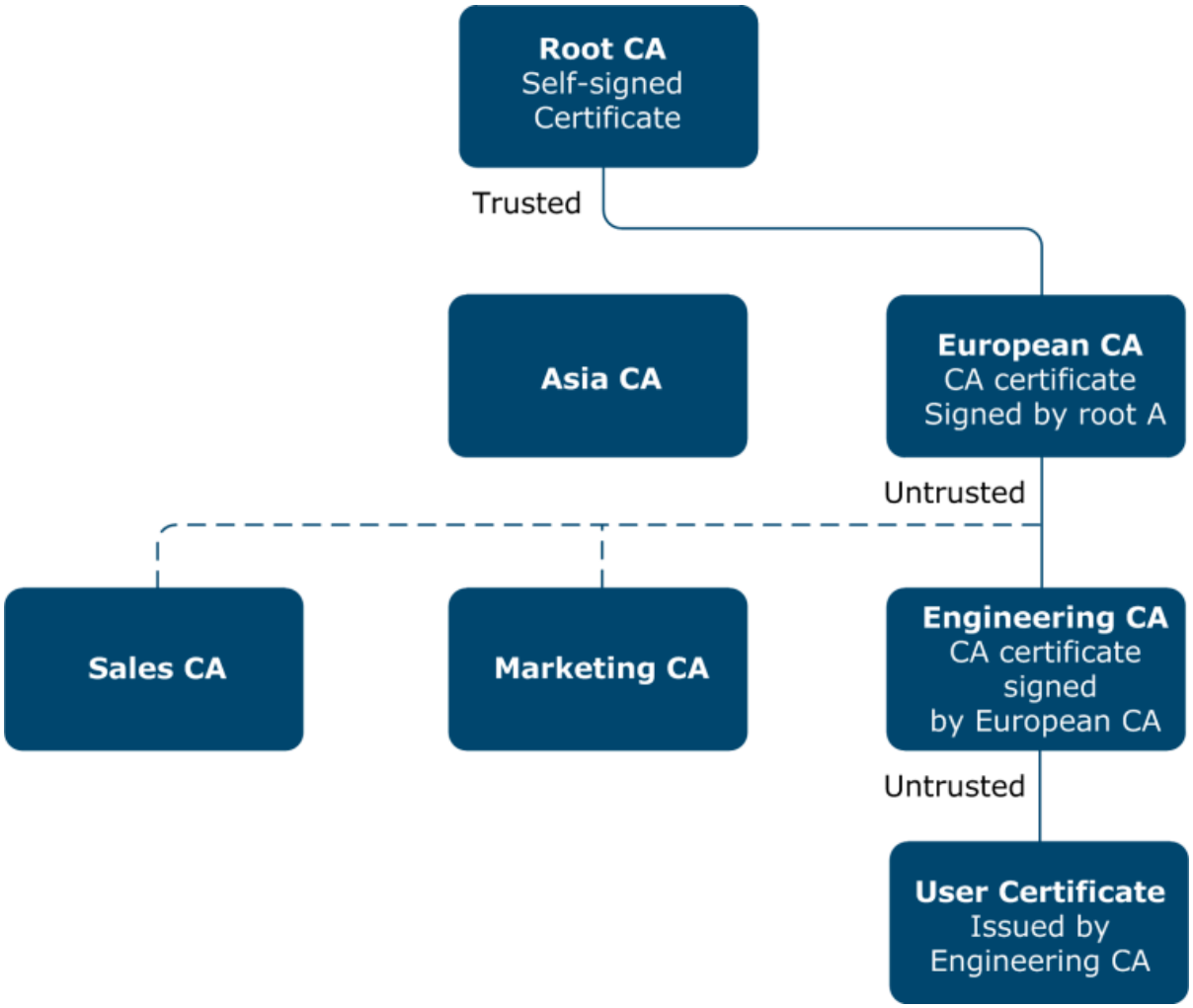
Certificate chains

CA hierarchies are reflected in certificate chains. A **certification path** is an ordered sequence of certificates between the end entity (a leaf) and the trusted point in the hierarchy (the root). The result forms a certificate chain that begins at the end entity and ends at the root CA.

A certificate chain traces a path of certificates from a branch to the root of the hierarchy. A certificate chain is formed as follows:

- Each certificate is followed by the certificate of its issuer.
- Each certificate contains the name (DN) of the issuer of that certificate, which is the same as the subject name of the next certificate in the chain.

Certificate chains



In the illustration above, the Engineering CA certificate contains the DN of the CA (European CA), that issued that certificate. The DN of the European CA is also the subject name of the next certificate in the chain.

Each certificate is signed with the private key of its issuer. The signature can be verified with the public key in the certificate from the issuer (which is the next certificate in the chain).

In the illustration above, the public key in the certificate for the European CA can be used to verify the digital signature of the European CA on the certificate for the Engineering CA.

Verifying a certificate chain

Certificate chain verification is the process of making sure a given certificate chain is well-formed, valid, properly signed, and trustworthy. The verification process:

1. Checks the certificate validity period against the current time on the system clock of the verifier.
2. Locates the certificate of the issuer. The source can either be the local certificate database for the verifier (on that client or server), or the certificate chain provided by the subject (for example, over an SSL connection).
3. Uses the public key in the issuer's certificate to verify the certificate signature.

4. Stops successfully at this point if the certificate of the issuer is trusted in the certificate database for the verifier. Otherwise, the verification procedure must verify the certificate of the issuer. The verification process returns to step 1) to verify this new certificate.

Syntax and formats

X.509 certificates are defined in *ISO x509* and correspond to a descriptive syntax known as **ASN1** (*Abstract Syntax Notation One*). The purpose of this syntax is to provide a system with an independent object description tool. In other words, this syntax allows you to describe objects used by applications in heterogeneous environments.

This document does not describe ASN1. For more information, refer to ISO/IEC8824-1. To be exploitable by a remote application (supported by any kind of machine), transmitted data (described with ASN1) must be encoded. The encoding rule must be the same on each communicating application, but the way this encoding rule is implemented is case-dependant (hardware-dependent, OS-dependent, and so on).

Distinguished **E**ncoding **R**ule (**DER**) is the encoding rule commonly used by applications that deal with X.509 certificates and public key infrastructures. The purpose of DER is to reduce ambiguities by providing a unique encoded value for a data set described with ASN1 syntax.

RSA public key cryptography standards

RSA laboratories defined various standards for security material, based on ASN1 descriptions. The purpose of these standards is to enable systems that participate in Public Key Infrastructures to inter-operate.

In addition, RSA provides a complete description of two data structures (PKCS7 and PKCS12). These data structures are useful when you want to transfer Certificate Information.

PKCS7 standard

Basically, the PKCS7 standard defines an envelope for transferring all kinds of data. This transfer may include cryptographic parameters, since data may be encrypted, signed, and so on. The cryptographic information could include:

- the encryption algorithm used to encode the data
- the tools required to authenticate the identity of the sender when data is signed: the user certificate and the whole certification chain

In a simplified case, when the PKCS7 structure data field is empty, it provides a means of distributing certificates.

A commonly used file extension for this standard is p7s.

PKCS8 Standard

The PKCS8 standard defines a structure used to store private key information. Private key information includes a private key for a public-key algorithm and a set of attributes. Only the PBE-MD5-DES password-based algorithm may be used to encrypt the key. The password is usually communicated offline.

PKCS12 standard

After processing the certification operation (which generally consists in generating a pair of keys, signing the

user certificate and sending the requested information back to the user), the certificate authority requires a secure way to send the user both the generated certificate and the private key. In the same way as PKCS7, the PKCS12 standard defines an envelope. This envelope contains password-based encrypted data.

To open the envelope and access the data, you must give the password required to generate a symmetric key used for deciphering. The password is usually communicated offline.

A commonly used file extension for this standard is p12.

Base 64 encoding

When you use email to transfer data, you may need to encode it with printable character representation only. Base 64 encoding converts binary data to printable characters. This enables you to DER-encode a certificate or a key and to encode the result using base 64. Base 64 is mandatory for compatibility with MIME.

A commonly-used file extension is .pem.

Note: Gateway does not support the import of DER-encoded keys protected by passphrase.

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Managing certificates

Axway Gateway: Managing Security

Overview

The Security Certificate object enables you to import a certificate from a file into the local database. Additionally, you can use the GUI Certificate object to check certificate status, export a certificate, delete or update a certificate.

The name of the certificate must be unique. The DN of the read certificate is checked to ensure that the certificate name is unique in the database.

When you import a user certificate, you must also import its associated private keys.

Importing a certificate

To import a certificate:

1. In the left pane of the GUI main window, click ⊞ to expand the nodes:
   Security Management > Transfer Security Management
2. Right-click the **Certificate** sub-node, and then select **Import...** from the context menu.
   Gateway displays the *Welcome to the Certificate Import Wizard* window.
3. Click **Next** to continue.
   Gateway displays the *File to import* window with these fields:

Field	Definition
Certificate file name	Enter the name of the file to import. Alternatively, use the **Browse** button to select the certificate to import.
Certificate file type	Select the type of file.
Key in separate file	Click box to enable. This activates the following fields.
Key file name	Enter the name of the key file. Alternatively, use the **Browse** button to select a key. This is the name of the separate file that contains private key elements. The key file name is not required with full PKCS12 certificates.
Key file type	Select the type of file.
Key password	Enter the key password. The password is required to decode an encrypted PKCS8 structure.

4. Complete the *File to import* window. Click **Next** to continue with the *Certificate Import Wizard*.
   Gateway displays the *Completing the certificate import* window.
5. Confirm the certificate information, and then click **Finish**.

Displaying a certificate

To display a certificate:

1. In the left pane of the GUI main window, click ⊞ to expand the nodes:
   Security Management > Transfer Security Management > Certificate
   Gateway displays nodes for the certificate types:
   - Root: Root certification authority certificate
   - CA: Intermediate Certification Authority certificate
   - User: User certificate
   - Other: Undetermined type of certificate
2. Click the node that represents the category of certificate to display.
   Gateway displays the certificates of the selected category in the display (right) pane.

Modifying certificate status

The certificate status is listed in the right pane work area. To modify the certificate status:

1. Display certificates (as described above).
2. In the display (right) pane, right-click the icon that precedes the name of the certificate for which you want to change status, and then choose **Select...** in the context menu.
   Gateway displays the **Select Certificates** tab.
3. Select a **State**:
   - Enabled
   - Disabled
   - To Check
   - Other
4. Click **OK** to confirm your selection.
   Gateway updates the status if the certificate.

Exporting a certificate

Not available in the current version.

Deleting a certificate

To remove a certificate from the database:

1. Display certificates (as described above).
2. In the display (right) pane, right-click the icon that precedes the name of the certificate you want to delete.
3. Select **Delete** from the context menu.
   Gateway removes the certificate from the database and removes the entry from the certificates list in the display pane.

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Certificate restrictions

Axway Gateway: Managing Security

Certificate types

Supported certificate content (X.509)

Signature algorithms

Certificate extension

Certificate types

The **Security server** certificate database stores both certificates and private keys used during secure file transfers with TLS/SSL or SSH.

Certificate Records fall into four categories:

- **Users:** End entity certificate whose private key is available
- **Others:** End entity certificate whose private key is unknown
- **Intermediary CA:** Authority certificate (not self-signed)
- **Root CA:** Authority self-signed certificate

Automatic type detection

When you import a certificate into the database, the server automatically detects the types to assign to the certificate, by:

- *comparing* the Certificate Issuer name with the subject name, to determine whether this certificate is a root certificate or not.
- *reading* the Basic Constraints Extension (if present), to determine whether the certificate is an authority certificate or not.
- *detecting* a Private Key: if a private key is provided with the certificate that you import, by default the server assigns the type **User** to the certificate.

Supported certificate content (X.509)

This section provides an overview of certificate fields as defined by ISO in the X.509 standard.

Users do not typically need to know the exact contents of a certificate. However, system administrators working with certificates may require some familiarity with the information provided here.

Version

Only X.509 certificates are supported. The certificate version may be any of those defined by the X.509 standard (V1 or V3).

Note: To automatically determine the type of the certificate to import into the database, the server first compares its subject and issuer name and then looks for the Basic Constraints Extension (only present in X.509 V3 certificates). That means that if the certificate that you import is V1, you will probably have to force the certificate type while importing it.

Validity period

Two formats are used to represent time values in X.509 certificates. These are UTC Time and generalized time. (Refer to X.509). Currently, only UTC Time format is supported. (You must take this into account before you import any certificate).

Distinguished names

For purposes of DN manageability, only the following AVA types are supported:

Distinguished Names: attributes and values

Component	Representation Symbol	Description
Country Name	C	The two-letter international country code (specified in ISO 3166)
Organization	O	The organization for which the certificate is issued Example: Axway
Organizational Unit- Name	OU	The division of the organization for which the certificate is issued
StateOrProvince Name	SP	The state or province in which the certificate owner is located
Locality Name	L	The locality in which the certificate owner is located
Common Name	CN	The name of the certificate owner that can be either a person such as "Robert Dupont" or a business role such as "technical manager"
Email address (deprecated)	E	The certificate owner email address Example: "asmith@axway.com"

In addition, for internal purposes, the Security server follows the RFC 1485 syntax for representing a distinguished name as a character string. The string is limited to 512 characters.

Signature algorithms

Only RSA-based signature algorithms are supported. Currently, two Hash algorithms are available and can be associated with RSA:

- MD5
- SHA-1

Supported RSA key lengths are 512, 1024, 2048, 4096 and 8192 bits.

Certificate extension

Extensions can only be present in X.509 V3 certificates. As described above, the Basic Constraint Extension, which indicates whether the certificate is an authority certificate or not, is the only extension used to automatically detect the certificate type.

This extension is not mandatory because X.509 V1 certificates are supported. However, automatic detection does not work for Intermediary CA in these cases (you may need to force the certificate type).

Import formats

Five distinct formats of certificates are supported:

- **Simple DER format:** This is the most commonly used certificate format and corresponds to the DER-encoded certificate structure.
- *Base 64*: This format consists of a simple DER certificate or a DER PKCS8 key, Base 64 encoded.
- *PKCS7*: This format (DER-encoded) allows you to import certificates from PKCS7.
  As described in the *PKCS7 standard*, only the simplified case, where the structure contains signed data, is taken into account. In this instance, only the first certificate in the chain is imported.
- *PKCS8*: This format describes syntax for private-key information. It includes private keys that may be encrypted, and additional attributes. The supported encryption method is PKCS#5 V1.5. This format is used to import a certificate and its associated private key separately.
- *PKCS12*: This format (DER-encoded) allows a one-shot import of a user certificate and its associated private key. The supported deciphering algorithms are:
  - RC2 CBC 40 bits
  - 3 DES CBC 128 bits

For more information on transferring cryptographic data, such as Certificates and Private keys, refer to Syntax and formats.

Activation flag

You can enable or disable every certificate in the local database using the GUI or the command line interface.

Check the activation flag when you define the certificate chain that you want to send. Your selection is also taken into account when you define the list of accepted authorities. To inhibit a certificate, you must remove it from the corresponding Security Profile or update the certificate database record with the status flag disabled.

You cannot send a disabled certificate in a certificate chain.

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Managing keys

Axway Gateway: Managing Security

Importing a Key

Displaying keys

Renaming a Key

Deleting a Key

Enabling and disabling a Key

Importing a Key

To import a Key using the GUI:

1. In the left pane of the GUI main window, click ⊞ to expand the nodes:
   Security Management > Transfer Security Management
2. Right-click the **Key** sub-node, and then select **Import...** from the context menu.
   Gateway displays the *Key management* window:

Field	Description
Key Name	Name of the key (must be unique). Maximum: 31 characters.
Enable	Status of the key. Check box to enable.
Group	Key group
Comment	Comments about the key.
File containing the key	Name and path for the file containing the key. You can use the Browse button to select the file.
File format	Key file encoding. Possible values: • DER • PEM
Key Type	Key type. Possible values: • PRIVATE • PUBLIC
Algorithm	Key file algorithm. Possible values: • RSA • DSS

Field	Description
	• ECDSA_NISTP256 • ECDSA_NISTP384 • ECDSA_NISTP521 • ED25519

3. Complete the fields and then click **OK** to confirm.
   Gateway displays the imported key in the right-hand pane.

Note Restrictions apply in FIPS mode when importing a key in Gateway. For more details see Using FIPS

Displaying keys

To display all keys in the database

1. In the left pane of the GUI main window, click ⊞ to expand the nodes:
   Security Management > Transfer Security Management
2. Click the **Key** sub-node.
   Gateway displays all existing Keys in the display pane (right pane). The display comprises columns containing the following key information:
   - Name
   - State
   - Group
   - Date created
   - Type
   - Comment
   - Fingerprint

Note In FIPS mode, the keys that do not meet the FIPS 140-2 standard are marked in Navigator GUI with a **FIPS restricted** label and icon.

Renaming a Key

To modify a Key name:

1. Display Keys (as described above).
2. Right-click to select the key to rename, and select **Rename** from the context menu.
   Gateway highlights the name in the display pane.
3. Type the new name over the existing name, and click outside of the name field to complete the process.

Deleting a Key

To remove a Key from the database:

1. Display Keys (as described above).
2. In the right pane, right-click to select the key to delete, and select **Delete** from the context menu.
3. In the confirmation window, click **YES**.
   Gateway removes the selected key from the database.

Enabling and disabling a Key

To disable a key that exists in the database:

1. Display Keys (as described above).
2. In the right pane, right-click to select the key to modify, and select **Enable/Disable** from the context menu.
   If the Key was:
   - *disabled*, Gateway changes the state to *enabled*
   - *enabled*, Gateway changes the state to *disabled*

   Gateway displays a tag in the **State** column for that key to indicate the status of the key:
   - Red = disabled
   - Green = enabled

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Command line operations

Axway Gateway: Managing Security

secadm import_cert

secadm update_cert

secadm delete_cert

secbase export cert

secbase export_cert_chain

secdsp select_cert

secdsp display_cert

secdsp status_cert

Importing a certificate: secadm import_cert

Syntax **secadm import_cert {-certificate_name} {-certificate_file}** [-certificate_group] [-status] [-certificate_file_type] [-pos] [-list_only] [-certificate_type] [-certificate_password] [-private_key_file] [-private_key_file_type] [-private_key_password]

Description Use this command to import a certificate from a file into the local database.

Parameters *Mandatory*

-certificate_name (-ctfn): Enter the name of the certificate to import.

Maximum: 31 characters

Mandatory

-certificate_file (-ctff): Enter the name of the file that contains the certificate to import.

-status (-cst): Enter the Status of the certificate you want to import.

Possible values:

- E = Enabled (default)
- D = Disabled
- C = To check

-certificate_file_type (-ctfft): Enter the encoding certificate file type.

Possible values:

- DER
- BASE64
- PKCS7
- PKCS12

-pos: Use this parameter to import only the certificate situated at this numbered position in the list.

-list_only: Use this parameter to list the certificates contained in the file without importing them (PKCS12 only).

-certificate_type (-ctft): Enter the certificate type. This parameter is mandatory if the imported certificate is an X.509 V1 certificate. For V3 certificates, Gateway determines the type automatically.

Possible values:

- USER = User certificate
- INTERMEDIATE = Intermediate certification authority certificate
- ROOT = Root certification authority certificate
- OTHER = Undetermined type of certificate

-certificate_password (-ctfp): Enter the password necessary to decrypt PKCS12 file.

Maximum: 16 characters.

This field is mandatory with the PKCS12 certificate file type. If it is not present, a password prompt is displayed.

Syntax **secadm import_cert {-certificate_name} {-certificate_file}** [-certificate_group] [-status] [-certificate_file_type] [-pos] [-list_only] [-certificate_type] [-certificate_password] [-private_key_file] [-private_key_file_type] [-private_key_password]

-certificate_group (-ctfg):

-private_key_file (-prkf): Enter the name of the separate file that contains private key elements.
Not required with full PKCS12 certificates

-private_key_file_type (-prkft): Enter a type for the password file.
This parameter is mandatory when the `-private_key_file` parameter is present.
Possible values:

- DER = RSA private key structure
- PKCS8 = PKCS8 structure
- CR_PKCS8 = Encrypted PKCS8 structure
- BASE64 = One of the above structures encoded with base64

-private_key_password (-prkp): Enter the password required to decode an encrypted PKCS8 structure.

Example The following command imports the certificate contained in the file cert.cer into the database and names it my_cert. Since no file type is entered, the default value is used (*DER*).
secadm import_cert –certificate_name "my_cert"
–certificate_file "cert.cer"
Note: The name of the certificate must be unique. The DN of the read certificate is checked to ensure that the certificate name is unique in the database.
When you import a user certificate, you must also import its associated private keys.

Updating a certificate: secadm update_cert

Syntax **`secadm update_cert {-certificate_name}`** `[-certificate_group] [-status] [-certificate_type]`

Description Use this command to modify a certificate.

Parameters *Mandatory*
-certificate_name (-ctfn): Enter the name of the certificate to modify.

-certificate_group (-ctfg):

-status (-cst): Enter a Status for the specified certificate:

- E = Enabled
- D = Disabled
- C = To check

-certificate_type (-ctft): Enter the certificate type for the specified certificate.
Possible values:

Syntax **`secadm update_cert {-certificate_name}`** `[-certificate_group] [-status] [-certificate_type]`

- USER = User certificate
- INTERMEDIATE = Intermediate certification authority certificate
- ROOT = Root certification authority certificate
- OTHER = Undetermined type of certificate

Example The following command disables the use of the certificate my_cert by resetting the status to *D (Disabled)*:
secadm update_cert –certificate_name "my_cert" –status D

Deleting a certificate: secadm delete_cert

Syntax **`secadm delete_cert {-certificate_name}`**

Description Use this command to remove a certificate from the database.

Parameters *Mandatory*
-certificate_name (-ctfn): Enter the name of the certificate you want to delete.

Example The following command removes the certificate named my_cert:
secadm delete_cert –certificate_name "my_cert"

Exporting a certificate: secbase export_cert

Syntax **`secbase export_cert [-certificate_name] [-status] [-certificate_type] [-certificate_group] [-subject_name] [-subject_alt_name] [-issuer_name] [-output] [-append]`**

Description Use this command to export a certificate from the local database to a file

Parameters `-certificate_name (-ctfn)`: Enter the name of the certificate to export.
Maximum: 31 characters

-status (-cst): Enter the Status of the certificate you want to export.
Possible values:

- **E** = Enabled (default)
- **D** = Disabled
- **C** = To check

`-certificate_type (-ctft)`: Enter the certificate type for the certificate you want to export
Possible values:

Syntax **secbase export_cert [-certificate_name] [-status] [-certificate_type] [-certificate_group] [-subject_name] [-subject_alt_name] [-issuer_name] [-output] [-append]**

- USER = User certificate
- INTERMEDIATE = Intermediate certification authority certificate
- ROOT = Root certification authority certificate
- OTHER = Undetermined type of certificate

-certificate_group (-ctfg):

-subject_name (-csn): Enter the name stored in the Subject field of the public key as selection criteria for the certificate you want to export.

`-subject_alt_name (-csa)`: Enter the name stored in the Subject_Alt field of the public key as selection criteria for the certificate you want to export.

`-issuer_name (-cin)`: Enter the Distinguished Name of the entity that signed and issued the certificates as selection criteria for the certificate you want to export.

`-output (-f)`: Name of the output file.
If you do not specify this parameter, the standard output file is used.
Maximum: 127 alphanumeric characters.

`-append (-ap)`: Concatenation at the end of the file (if it exists).
This parameter takes no value.

Example The following command exports the certificate with the name `my_cert` from database into the `file my_cert.outFile`.

secbase export_cert -certificate_name "my_cert" -f my_cert.outFile

Exporting a certificate chain: secbase export_cert_chain

Syntax **secbase export_cert_chain [-certificate_name] [-status] [-certificate_type] [-human_readable] [-print_certificate_content] [-certificate_path_build_mode] [-output] [-append]**

Description Use this command to display/export a certificate chain from the local database to a file

Parameters `-certificate_name (-ctfn)`: Enter the name of the certificate for which you want the certificate chain.
Maximum: 31 characters

`-status (-cst)`: Enter the Status of the certificate for which you want the certificate chain.
Possible values:

Syntax **`secbase export_cert_chain [-certificate_name] [-status] [-certificate_type] [-human_readable] [-print_certificate_content] [-certificate_path_build_mode] [-output] [-append]`**

- E = Enabled (default)
- D = Disabled
- C = To check

`-certificate_type (-ctft)`: Enter the certificate type of the certificate for which you want the certificate chain.
Possible values:

- USER = User certificate
- INTERMEDIATE = Intermediate certification authority certificate
- ROOT = Root certification authority certificate
- OTHER = Undetermined type of certificate

`-human_readable (-hr)`: displays the names of the certificates that builds the chain in an easier to view mode.
This parameter takes no value.

`-print_certificate_content (-pcc)`: displays the content of the certificates that builds the chain, useful for moving a specific chain from a Gateway to another.
This parameter takes no value.
Note: print_certificate_content and human_readable cannot be used together.

`-certificate_path_build_mode (-cpbm)`: select the information used to build the certificate chain
Possible values:

- **legacy** = based on matching issuer dn with Subject dn
- **rfc4158**= based on matching subject key ID with authority key ID certificate extensions.

`-output (-f)`: Name of the output file.
If you do not specify this parameter, the standard output file is used.
Maximum: 127 alphanumeric characters.

`-append (-ap)`: Concatenation at the end of the file (if it exists).
This parameter takes no value.

Example The following command displays the certificate chain for certificate with the name `my_cert`, in a parsable format, from database into the file `my_cert.outFile`.

`secbase export_cert_chain  -certificate_name "my_cert" -f my_cert.outFile`
#?BEGIN_CERT_CHAIN
[0][my_cert][C=RO, O=Axway]
[1][root][C=RO, O=Axway]

Syntax **secbase export_cert_chain [-certificate_name] [-status] [-certificate_type] [-human_readable] [-print_certificate_content] [-certificate_path_build_mode] [-output] [-append]**

#?END_CERT_CHAIN

Note Line format is :

[number in chain] [gateway certificate name] [subject dn]

Note If the chain cannot be completed due to a missing certificate, a line like the following will be printed at the end of the chain:

[-][----][dn of the missing certificate]

The following command displays the certificate chain in a human readable format:

secbase export_cert_chain -ctfn "my_cert" -f my_cert.outFile -hr

[1][root][C=RO, O=Axway]

\

-->[0][my_cert][C=RO, O=Axway]

Displaying a certificate: secdsp display_cert

Syntax **secdsp display_cert {-certificate_name}** [-display_cert] [-display_key]

Description Use this command to display names of certificates filtered by selection criteria.

Parameters *Mandatory*

-certificate_name (-ctfn): Enter the name of the certificate to display.

-display_cert (-dspc): If present, certificate is displayed. No value required.

-display_key (-dspk): If present, key is displayed. No value required.

Example The following command displays attributes of the certificate named my_cert:

secdsp display_cert –certificate_name "my_cert"

Listing certificate names: secdsp select_cert

Syntax **secdsp select_cert** [-subject_name] [-subject_alt_name] [-issuer_name] [-status]

Description Use this command to display a certificate.

Parameters -subject_name (-csn): Enter the name stored in the Subject field of the public key as selection criteria for the certificate you want to display.

-subject_alt_name (-csa): Enter the name stored in the Subject_Alt field of the public key as selection criteria for the certificate you want to display.

-issuer_name (-cin): Enter the Distinguished Name of the entity that signed and issued the certificates as selection criteria for the certificate you want to display.

Syntax **secdsp select_cert** [-subject_name] [-subject_alt_name] [-issuer_name] [-status]

-status (-cst): Enter the Status as selection criteria for the certificate you want to display.
Possible values:

- E = Enabled
- D = Disabled
- C = To check

Example The following command displays the names of all the certificates in the database:
secdsp select_cert

Listing certificate abstracts: secdsp status_cert

Syntax **secdsp status_cert** [-subject_name] [-subject_alt_name] [-issuer_name] [-status]

Description Use this command to display abstracts of the matched certificates.

Parameters -subject_name (-csn): Enter the name stored in the Subject field of the public key as selection criteria for the certificate for which you want check the status.

-subject_alt_name (-csa): Enter the name stored in the Subject_Alt field of the public key as selection criteria for the certificate for which you want check the status.

-issuer_name (-cin): Enter the Distinguished Name of the entity that signed and issued the certificates as selection criteria for the certificate for which you want check the status.

-status (-cst): Enter the Status as selection criteria for the certificate for which you want check the status.
Possible values:

- E = Enabled (default)
- D = Disabled
- C = To check

Example The following command displays an abstract of each certificate in the database:
secdsp status_cert

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Axway Gateway: Managing Security

secadm import_key

Importing a key: secadm import_key

Syntax	**secadm import_key {-name} {-file_name} {-file_format}** [-algorithm] [-type] [-group] [-state] [-comments]
Description	Use this command to import a key into the local database.
Parameters	*Mandatory* -name (-n): Enter the name of the key you are importing. The name of the key must be unique. (max 31 characters).
	Mandatory -file_name (-fn): Enter the name of the file that contains the key to import.
	Mandatory -file_format (-ff): Enter the Key file encoding. Possible values: • DER • PEM (default) • TXT • OPENSSH
	-algorithm (-algo): Enter the Key file algorithm. Possible values: • RSA • DSS • ECDSA_NISTP256 • ECDSA_NISTP384 • ECDSA_NISTP521 • ED25519
	-type (-t): Enter the key type. Possible values: • PRIVATE • PUBLIC

Syntax	**secadm import_key {-name} {-file_name} {-file_format}** [-algorithm] [-type] [-group] [-state] [-comments]
	-group (-g): Enter the group of the Key.
	-state (-s): Enter the operational status of the Key: • E = Enabled • D = Disabled • C = To check
	-comments (-cts): Enter free text comments about the key.
Example	The following command imports the key contained in the file rsaprv1024.pem into the database and names it rsaprv1024. The file type is the default value (PEM). **secadm import_key –name "rsaprv1024" –filename "rsaprv1024.pem"**

Updating a key: secadm update_key

Syntax	**secadm update_key {-name}** [-state] [-group] [-new_name] [-comments]
Description	Use this command to modify a key.
Parameters	*Mandatory* -name (-n): Enter the name of the key you are modifying.
	-state (-s): Enter the operational status of the Key: • E = Enabled • D = Disabled • C = To check
	-group (-g): Enter the group of the Key.
	-new_name (-nn): Enter a new name for the Key.
	-comments (-cts): Enter free text comments about the Key.
Example	The following command disables the use of the key rsaprv1024: **secadm update_key –name "rsaprv1024" –status DISABLED**

Deleting a key: secadm delete_key

Syntax	**`secadm delete_key {-name}`**
Description	Use this command to remove a Key from the database.
Parameters	*Mandatory* -name (-n): Enter the name of the Key you want to delete.

Syntax	**`secadm delete_key {-name}`**
Example	The following command removes the key named rsaprv1024: secadm delete_key –name "rsaprv1024"

Displaying a key: secdsp display_key

Syntax	**`secdsp display_key {-name}`**
Description	Use this command to display the attributes of a Key.
Parameters	*Mandatory* -name (-n): Enter the name of the Key for which you want to display the attributes.
Example	The following command displays attributes of the key named rsaprv1024: secdsp display_key –name "rsaprv1024"

Listing key names: secdsp select_key

Syntax	**`secdsp select_key`**
Description	Use this command to display the names of the keys in the database.
Parameters	`-name` (`-n`): Enter the name of the Key for which you want to display the attributes.
	`-group` (`-g`): Enter the group of the Key.
	`-state` (`-s`): Enter the operational status of the Key: • E = Enabled • D = Disabled • C = To check
	`-algorithm` (`-algo`): Enter the Key file algorithm. Possible values: • RSA • DSS • ECDSA_NISTP256 • ECDSA_NISTP384 • ECDSA_NISTP521 • ED25519
	`-type` (`-t`): Enter the key type. Possible values: • PRIVATE • PUBLIC
Example	The following command displays a list of the names of all enabled keys in the database:

Syntax	**secdsp select_key**
	secdsp select_key -state ENABLED

Listing key status: secdsp status_key

Syntax	**secdsp status_key**
Description	Use this command to display the status (Enabled/Disabled/To_check) of the Keys in the database.
Parameters	`-name` (`-n`): Enter the name of the Key for which you want to display the attributes.
	`-group` (`-g`): Enter the group of the Key.
	`-state` (`-s`): Enter the operational status of the Key: • E = Enabled • D = Disabled • C = To check
	`-algorithm` (`-algo`): Enter the Key file algorithm. Possible values: • RSA • DSS • ECDSA_NISTP256 • ECDSA_NISTP384 • ECDSA_NISTP521 • ED25519
	`-type` (`-t`): Enter the key type. Possible values: • PRIVATE • PUBLIC
Example	The following command displays a list of the names of all enabled keys in the database: **secdsp select_key -state ENABLED**

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Network Profiles

Axway Gateway: Managing Security

Network Profiles associate an incoming network connection request with a Security Profile. If no Network Profile matches the incoming connection, or if the matched Profile does not refer to an existing Security Profile, the session is interrupted. If TLS or SSH is not enabled in the selected Network Profile, the connection does not use the Security server.

Matching an incoming connection with a Network Profile is based on:

- Destination address and SAP (local)
- Origin address and SAP (remote)

A Network Profile comprises:

- Destination address pattern (address/SAP)
- Origin address pattern (address/SAP)
- Type option that indicates whether or not to use TLS or SSH security
- Security Profile name (optional and only used if TLS or SSH option is activated)

All patterns can contain the following substitution characters:

- * matches any number of characters
- ? matches a single character or no characters

You can define Network Profiles either via the GUI or via command line.

Related topics

Managing Network Profiles

Managing Network Profiles (command line)

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Managing Network Profiles

Axway Gateway: Managing Security

Creating Network Profiles

Displaying Network Profiles

Filtering the Network Profile display

Deleting a Network Profile

Updating a Network Profile

Creating Network Profiles

To create a Network profile:

1. In the left pane of the GUI main window, click ⊞ to expand the nodes:

Security Management > Transfer Security Management

2. Right-click the **Network Profile** sub-node, and then select **New** from the context menu.

Gateway displays the *New Network Profile* window.

3. Complete the fields of the *New Network Profile* window as described in the following table. Click **OK** to confirm.

Depending on the fields and options that you select, the other options may change dynamically.

Field	Description
Network Profile	Enter a name for the Network Profile. The Profile name must be unique in the Network Profile database. Maximum: 31 characters.
Origin address/ Port	Enter the originating address and **Port** number that this Profile is associated with.
Destination address/ Port	Enter the destination address and **Port** number that this Profile is associated with.
Network type	Select the Network type: • TCP/IP
Network security	Select the type of security required for this Profile: • None • TLS • SSH
Security Profile	Select the Security Profile to associate with this Network Profile.

Displaying Network Profiles

To display all Network Profiles in the database:

1. In the left pane of the GUI main window, click ⊞ to expand the nodes:
Security Management > Transfer Security Management
2. Click to select the **Network Profile** sub-node.

Gateway displays all existing Network Profiles in the display pane (right pane). The following information is listed for each Network Profile:

- Name
- Type
- Communication network type
- Origination address
- Destination address
- Security profile

Filtering the Network Profile display

To regenerate a display of Network Profiles according to filtering criteria:

1. In the left pane of the GUI main window, click ⊞ to expand the nodes:

Security Management > Transfer Security Management

2. Right-click the **Network Profile** sub-node, and then choose **Select** from the context menu.

Gateway displays the *Select Network Profiles* window.

3. In the *Select Network Profiles* window, enter values in the fields you want to use as selection criteria. For example, to display all Network Profiles for TCP/IP networks, select **TCP/IP** in the Network type selection box.
4. When you have completed the fields you want to use as filtering criteria, click **OK** to confirm.

Gateway refreshes the display list of Network Profiles, applying your filtering criteria to determine the display content.

Deleting a Network Profile

To remove a Network Profile from the database:

1. In the left pane of the GUI main window, click ⊞ to expand the nodes:

Security Management > Transfer Security Management

2. Click to select the **Network Profile** sub-node.

Gateway displays all existing Network Profiles in the display pane (right pane).

3. In the display pane (right pane), right-click the icon at the beginning of the line that represents the Network Profile you want to delete, and then select **Delete** from the context menu.

Gateway displays a confirmation dialog box.

4. In the confirmation dialog box, click **Yes** to continue with the deletion process.

Gateway removes the Network Profile from the database and from the display pane.

Updating a Network Profile

To modify and update a Network Profile:

1. In the left pane of the GUI main window, click ⊞ to expand the nodes:

Security Management > Transfer Security Management

2. Click to select the **Network Profile** sub-node.

Gateway displays all existing Network Profiles in the display pane (right pane).

3. In the display pane (right pane), right-click the icon at the beginning of the line that represents the Network Profile you want to modify and update.

Gateway displays the a context menu.

4. Select **Modify** from the context menu.

Gateway opens the *Network Profile* editing window. This window contains the same fields as the New Network Profile window, used for creating a Network Profile above.

Modify the contents of any of the displayed fields as required.

5. Click **OK** to validate your changes.

Gateway closes the *Network Profile* editing window, and updates the Network profile. The new values are visible in the display pane.

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Command line operations

Axway Gateway: Managing Security

Commands for creating and managing Network Profiles:

- *secadm create_netprof* – Create a Network Profile
- *secadm update_netprof* – Update a Network Profile
- *secadm delete_netprof* – Delete a Network Profile

Commands for displaying Network Profiles:

- *secdsp select_netprof* – Display a selected list of Network Profiles
- *secdsp display_netprof* – Display the parameters of a Network Profile

- *secdsp status_netprof* – Display a selected list of Network Profiles with a subset of information for each selected Profile

Commands for exchanging Network Profiles between Gateway installations or Gateway versions:

- *secbase export_netprof* – Create an output file containing Network Profile definitions
- *secbase import* – Import a Network Profile

Creating Network Profiles: secadm create_netprof

Syntax	**secadm create_netprof {-nprof_name}** [-security_option] [-comm_type] [-org_address] [-dest_address] [-user_data] [-state] [-sprof_name]
Description	Use this command to create a Network Profile.
Parameters	*Mandatory* -nprof_name (-npn): Enter a name for the Network Profile you are creating. The name of the Network Profile must be unique. -security_option (-sec_opt): Enter the security option for the Network Profile. Possible values are: • None (default) • TLS • SSH -comm_type (-ct): Enter the type of network connection associated with the profile. Enter: • TCPIP = TCP/IP (default) -org_address (-oa): Enter the originating address and SAP. Maximum: 31 characters. -dest_address (-da): Enter the destination address and SAP. Maximum: 31 characters. -sprof_name (-spn): Enter the Security Profile name in the Network Profile. Maximum: 31 characters.
Example	The following command creates a Network Profile named local_netprof. This Profile activates the use of TLS for any network connection coming from addresses 10.10.10.* towards address 10.10.10.1 and port 2205. The associated Security Profile is sprof_1.

Syntax	**secadm create_netprof {-nprof_name}** [-security_option] [-comm_type] [-org_address] [-dest_address] [-user_data] [-state] [-sprof_name]
	secadm create_netprof -nprof_name "local_netprof" -org_address "10.10.10.*/*" -dest_address "10.10.10.1/2205" -sec_opt TLS -sprof_name "sprof_1"

Updating Network Profiles: secadm update_netprof

Syntax	**secadm update_netprof {-nprof_name}** [-security_option] [-comm_type] [-org_address] [-dest_address] [-user_data] [-state] [-sprof_name]
Description	Use this command to modify and update a Network Profile.
Parameters	*Mandatory* -nprof_name (-npn): Enter a name for the Network Profile you are updating.
	-security_option (-sec_opt): Enter the security option for the Network Profile. Possible values are: • None (default) • TLS • SSH
	-comm_type (-ct): Enter the type of network connection associated with the profile. Enter : • TCPIP = TCP/IP (default)
	-org_address (-oa): Enter the originating address and SAP. Maximum: 31 characters.
	-dest_address (-da): Enter the destination address and SAP. Maximum: 31 characters.
	-sprof_name (-spn): Enter the Security Profile name in the Network Profile. Maximum: 31 characters.
Example	The following command modifies the Network Profile named local_netprof by replacing the original address pattern with 10.10.*.*/*. secadm update_netprof -nprof_name "local_netprof"

Syntax	**secadm update_netprof {-nprof_name}** [-security_option] [-comm_type] [-org_address] [-dest_address] [-user_data] [-state] [-sprof_name]
	-org_address "10.10.*.*/*"

Deleting Network Profiles: secadm delete_netprof

Syntax	**secadm delete_netprof {-nprof_name}**
Description	Use this command to remove a Network Profile.
Parameters	*Mandatory* -nprof_name (-npn): Enter the name of the Network Profile you want to delete.
Example	The following command deletes the Network Profile named local_netprof: secadm delete_netprof -nprof_name "local_netprof"

Displaying Network Profiles: secdsp display_netprof

Syntax	**secdsp display_netprof {-nprof_name}**
Description	Use this command to display the attributes of a Network Profile.
Parameters	*Mandatory* -nprof_name (-npn): Enter the name of the Network Profile for which you want to display the attributes.
Example	The following command displays the attributes of the Network Profile named local_netprof: secdsp display_netprof -nprof_name "local_netprof"

Listing Network Profile names: secdsp select_netprof

Syntax	**secdsp select_netprof** [-security_option] [-comm_type] [-org_address] [-dest_address] [-user_data] [-sprof_name]
Description	Use this command to display a filtered list of the names of the Network Profiles in the database.
Parameters	-security_option (-sec_opt): Enter the security option for the Network Profile as display filtering criteria. Possible values are:

Syntax	**secdsp select_netprof** [-security_option] [-comm_type] [-org_address] [-dest_address] [-user_data] [-sprof_name]

- None (default)
- TLS
- SSH

-comm_type (-ct): Enter the type of network connection associated with the profile as display filtering criteria.
Enter :

- TCPIP = TCP/IP (default)

-org_address (-oa): Enter the originating address and SAP as display filtering criteria. Maximum: 31 characters.

-dest_address (-da): Enter the destination address of the connection as display filtering criteria.

-sprof_name (-spn): Enter the Security Profile name in the Network Profile as display filtering criteria.
Maximum: 31 characters.

Example	The following command lists the names of all Network Profiles that activate TLS and use the Security Profile sprof_1: **secdsp select_netprof -sec_opt TLS -sprof_name "sprof_1"**

Listing Network Profile abstracts: secdsp status_netprof

Syntax	**secdsp status_netprof** [-security_option] [-comm_type] [-org_address] [-dest_address] [-user_data] [-sprof_name]
Description	Use this command to display lists of Network Profile abstracts, filtered according to selected criteria.
Parameters	-security_option (-sec_opt): Enter the security option for the Network Profile as display filtering criteria. Possible values are:

- None (default)
- TLS
- SSH

Syntax	**secdsp status_netprof** [-security_option] [-comm_type] [-org_address] [-dest_address] [-user_data] [-sprof_name]
	-comm_type (-ct): Enter the type of network connection associated with the profile as display filtering criteria. Enter either: • TCPIP = TCP/IP (default)
	-org_address (-oa): Enter the originating address and SAP as display filtering criteria. Maximum: 31 characters.
	-dest_address (-da): Enter the destination address of the connection as display filtering criteria.
	-sprof_name (-spn): Enter the Security Profile name in the Network Profile as display filtering criteria. Maximum: 31 characters.
Example	The following command lists abstracts of all Network Profiles that activate the use of TLS: secdsp status_netprof -tls_option Y

Exporting and importing Network Profiles

Gateway provides two commands for exchanging security objects between Gateways and Gateway versions:

- *secbase export_netprof*
- *secbase import*

Exporting Network Profiles to a file: secbase export_netprof

Syntax	**secbase export_netprof** [optional parameters - listed below]
Description	Use this command to export the fields and field values of Network Profile objects to an ASCII format output file. After running the command, you can open and edit the output file. Use the secbase import command to import the file contents to Gateway.
Parameters	-security_option (-sec_opt): Enter one of the security option values as filtering criteria: • **NONE** • **TLS** • **SSH**

Syntax	**secbase export_netprof** [optional parameters - listed below]
	-comm_type (-ct): Enter the communication type as filtering criteria: • TCPIP
	-org_address (-oa): Enter the originating address and SAP of the connection as filtering criteria.
	-dest_address (-da): Enter the destination address and SAP of the connection as filtering criteria.
	-sprof_name (-spn): Enter the name of the associated Security Profile as filtering criteria.
	-format (-of): Select an output format. Enter one of the following values: • S = Shell (default). For example, field = *value* • C = C-Shell. For example, set field = *value* • U = User format
	-user_fmt (-uf): User format definition. Enter a character string including the keywords %N and %V, where: • %N = name of field • %V = value of field
	-output (-f): Enter the name of the output file that results from the command. If you do not specify this parameter, the standard output file is used. Maximum: 127 alphanumeric characters.
	-append (-ap): Concatenation at the end of the file (if it exists). This parameter takes no value.
Example	Enter the command: **secbase export_netprof -sec_opt SSH -f ExpProf** Gateway creates the ExpProf export file. In the database, it locates Network Profile objects named with SSH security options. From these objects, it extracts each field and its corresponding value, and writes them to the ExpProf file.

Importing Network Profiles from a file: secbase import

Syntax	**secbase import {-input}** [-netprof] [-sprof] [-sshprof] [-cert] [-key] [-error_free]
Description	Use this command to import the fields and field values of one or more security objects (for example Network Profiles) contained in an importable object file. You can create importable files of Network Profiles using the secbase export_netprof command.
Parameters	*Mandatory* -input (-if): Enter the name of the file containing the Network Profile objects you want to import.
	-netprof (-np): Enter the name of a Network Profile contained in the output file named in the -input parameter.
	-sprof (-sp): Enter the name of a Security Profile contained in the output file named in the -input parameter.
	-sshprof (-shp): Enter the name of an SSH Security Profile contained in the output file named in the -input parameter.
	-error_free: Use this parameter to create the different objects while ignoring the new parameters and without stopping the process at each error.
Example	Enter the following command to import the object contents of the ExpProf file: **secbase import -if ExpProf**

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

SSH

Axway Gateway: Managing Security

About SSH protocol

Establishing an SSH connection

Establishing security with SSH

Checking the login_user

Using SSH with SFTP

SSH renegotiation

About SSH protocol

SSH (Secure **Sh**ell) was created by SSH Communications Security Inc. (www.ssh.com) to provide secure login over an insecure network. The SSH2 protocol is being standardized by the Secsh Working Group of the Internet Engineering Task Force (IETF).

SSH provides the following services:

- *Authentication*
- *Integrity*
- *Confidentiality*

These services are performed on three communication layers, as defined in the SSH RFC:

- Transport Layer Protocol (TLP) layer
- User Authentication Protocol (UAP) layer
- Connection Protocol (CP) layer

The Transport Layer Protocol and User Authentication Protocol allow you to establish the secure tunnel (via authentication, encryption and hash comparison).

The Connection protocol layer relies on the other two layers. It allows you to open one or several channels (multiplexing) above the first two layers, and controls the data flow.

Establishing an SSH connection

SSH is a protocol based on mutual authentication. The client authenticates the server and the server authenticates the client so that both parties are assured of the identity of the other party.

To establish an SSH connection, the client must have server identification details (obtained by a method external to SSH). If the server authentication is by certificate, the client must have the certificate of the certification authority that issued the certificate and the certificate path (all the certificates up to the root).

For a successful connection, the server must also have the following client authentication parameters:

- Client name
- Authentication parameters (key, certificate or password) depending on negotiated authentication method

The connection procedure:

- Sets up Transport Layer Protocol: Negotiation of encryption methods and integrity check (authentication of server)
- Negotiates authentication method
- Authenticates the client

When both the server and client are authenticated, the secured channel is open and you can begin transferring files.

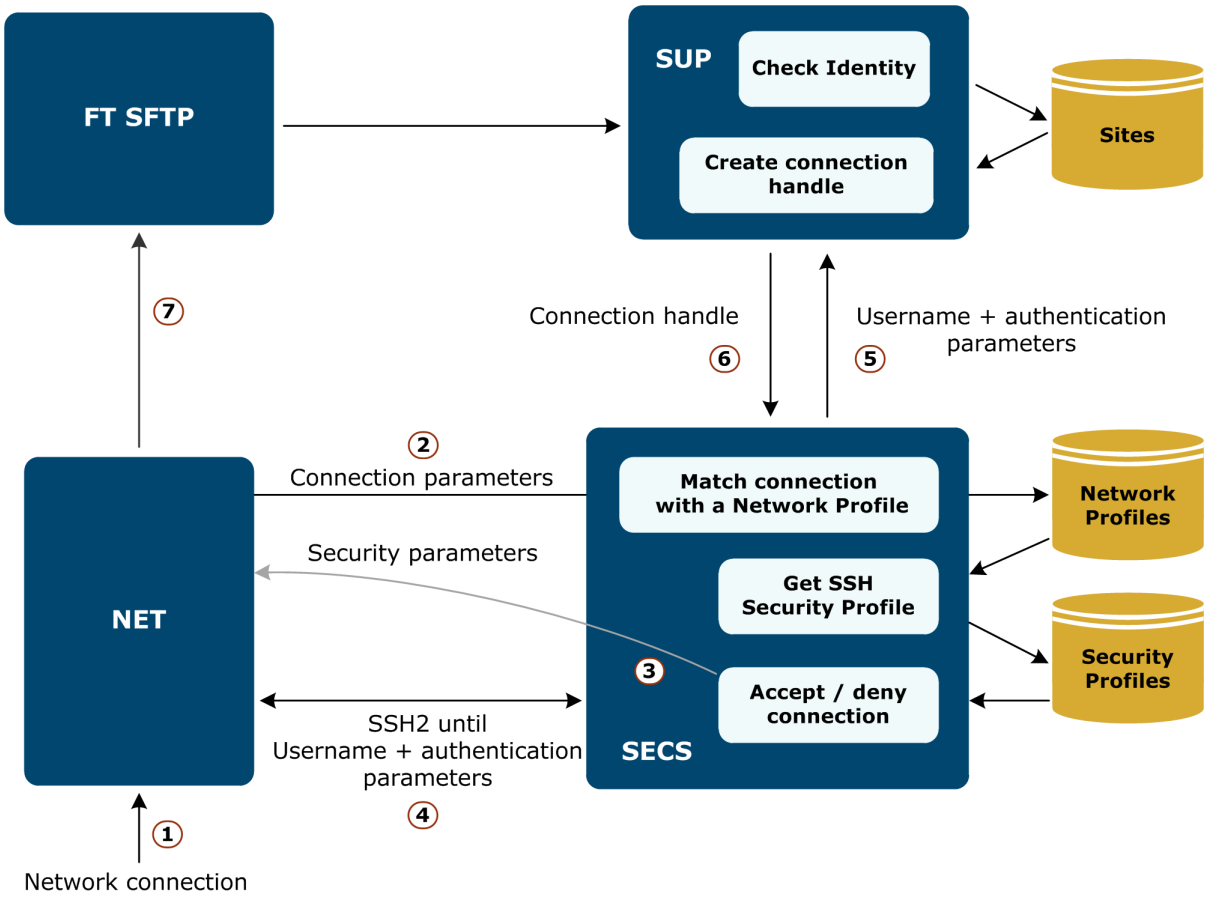
Establishing security with SSH

Incoming connection

Following the connection, the network sends the session initialization parameters to the Security server. For incoming connections, network parameters are used to retrieve a Network Profile object.

If the security option is set to SSH, the Network Profile indicates the SSH Profile to use.

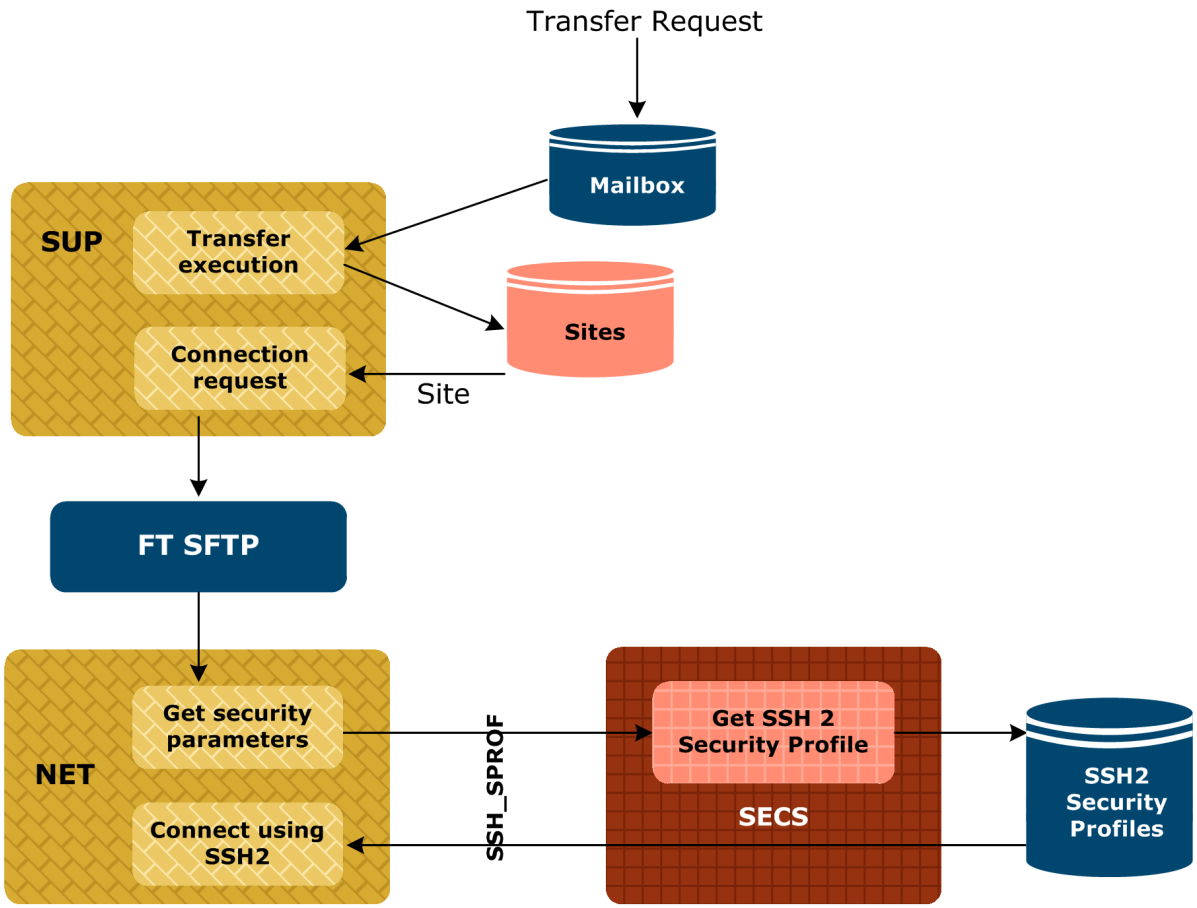
The following diagram schematically illustrates Gateway implementation of an incoming SSH call:



Outgoing connection

The Remote Site parameter **sshprof_out** indicates which SSH Security Profile to use for the outgoing connection.

The following diagram schematically illustrates a Gateway implementation of an outgoing SSH call:



Checking the login_user

If Gateway successfully identifies a Network Profile to use for an incoming connection, it then uses the **login_user** parameter to identify a CGate. A Remote Site template is defined in the CGateGroup. The authentication control parameters of the partner are found in the Remote Site.

Using SSH with SFTP

SSH is only used with the SSH File Transfer Protocol (SFTP). SFTP has no integral identification mechanism and therefore relies on the SSH protocol for user authentication and a secure connection.

The two protocols are separate and technically different, but are exclusively used together in Gateway.

The configuration of an SFTP/SSH link requires the configuration of several compatible objects in Gateway:

- *Remote Site*
- *Network Profile (netprof) object*
- *SSH Profile (sshprof) object*

In the SSH Profile object you must define a **key exchange algorithm** and a **DSS**, **ECDSA** or **RSA** key (or an **X.509** certificate).

The SFTP layer is above the three protocol layers used by SSH (TLP, UAP and CP).

Channel management is necessary, but typically SFTP does not need several channels, because it is simpler to open several requests in parallel within one unique channel. Opening the channels occurs on the initiative of the client.

Gateway supports SFTP in both client and server modes.

SSH renegotiation

SSH renegotiation is the renewal of the encryption keys, authentication, and compression during an SSH session. The term *renegotiation* is used because the protocol initiates at that moment a real negotiation between the client and the server about the choice of encryption, authentication and compression algorithms to be used for the continuation of the session.

SSH makes it possible to establish a secure channel between two computers allowing the remote control of one of the computers, file transfer, and so on. The world of Internet has realized that even though SSH offers a very high level of security, it is theoretically possible for a hacker to crack the keys and intercept the communication. To be able to do that a hacker needs a certain volume of data and/or time. Nowadays it is therefore recommended to change encryption keys after transferring 1 GB of data or after one hour, whichever occurs first.

Response to a key change request

Without the need to set any parameters, Gateway automatically responds to key change requests originating from a third party (server or client).

Initiation of a key change

Gateway can initiate a key change renegotiation. The behavior depends on the values of two parameters set in the SSH Security Profile:

- **Minutes before New Keys** - the time period after which Gateway initiates an SSH renegotiation
- **MB before New Keys** - the data volume transmitted in one direction after which Gateway initiates an SSH renegotiation

Gateway initiates key renegotiation when either of these two limits is reached.

To set these parameters, refer to Managing SSH Security Profiles.

Related topics

Managing Keys and Certificates in SSH

Using SSH

Managing SSH Security Profiles

SSH authentication

Managing SSH Security Profiles (command line)

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release*

Notes

Managing Keys and Certificates in SSH

Axway Gateway: Managing Security

This topic describes how to import and manage partner keys (and certificates).

Overview

For known partners, the Security server can run in complete autonomy. It requires no external services. The Security server uses information stored locally in its database to perform the remote communicating partner mutual authentication.

When a new partner attempts to connect, there are several possible scenarios:

- Known partner sends a certificate from a known CA: the connection is accepted.
- Known partner sends a public key with signature: the connection is accepted.
- Unknown partner: key or certificate is imported or not depending on how you configure your automatic import parameter. Gateway rejects the connection and allows you to manually enable the key or certificate.

Managing key pairs

SSH uses DSS, ECDSA and RSA keys (or RSA keys in certificates) to authenticate partners. Gateway supports DSS, ECDSA and RSA keys and certificates with RSA keys. Supported ECDSA keys are the ones required by RFC5656 (nistp256, nistp384 and nistp521).

The private key is a local key that is attributed to Gateway (or an application using Gateway). It has a public and private part.

A public key originates from a partner. This partner must have the private key to be able to authenticate itself to Gateway. The partner public keys are referred to as hosted keys.

The partner public keys are stored in the Gateway key database.

A key pair can be either a local key (public and private key parts) or a partner key (public key only). A Gateway database exists for storing the key pairs.

A **KeyPairRecord** consists of the following fields:

- **Alias**: unique logical name to identify the key
- **Group**: group name to organize keys by group
- **Comment**: user comment
- **State**: *ENABLED*, *DISABLED* or *TO_CHECK*
- **Algorithm**: *RSA*, *DSS*, *ECDSA_NISTP256*, *ECDSA_NISTP384*, or *ECDSA_NISTP521*
- **CreatedOn**: import / creation date
- **IsPrivateKeySet**: value is 1 if the key is local (private part is completed) otherwise 0 (public only)
- **Fingerprint**: hash of the public part of the key
- **Value of the Key**: triplet (n, e, d) for RSA (max length 4096 bits), quintuplet (p, q, g, y, x) for DSS (max length 1024 bits)

In an imported key, you can only modify the fields **Alias**, **Group** and **Comment**.

Importing key pairs automatically

You can configure Gateway to import remote keys automatically in the **SSH Profile**, by setting **remote_key_import** parameter to Y.

If you set the automatic remote key import (**remote_key_import** = Y), and the public key of the partner is not present in the database, then this public key is stored in the database and the connection is closed.

Note: Unknown partners are always rejected even if their public keys are imported into the database. Imported keys remain in the *TO_CHECK* state until they are manually validated.

Gateway generates an alias automatically for the new key. The comment zone includes the partner name. You can modify the alias and the comment zone of an imported key.

When you import a key automatically it is associated with the group **public_key_group** defined in the Site (or the SSH Profile). If no **public_key_group** is defined, they are associated with the group GDEFAULT.

The public keys must be unique in the database. Therefore if a client tries to connect several times with the same key, only one key record will be created in the database.

The automatic import process applies both to client and server mode.

This method also applies to importing and validating certificates.

Importing key pairs manually

Alternatively, you can use the secadm import_key command to manually import known partner keys (DER or PEM format).

If you do not define a **public_key_group**, the key is associated with the group GDEFAULT by default.

Validating (Enabling) a key

When you import a new key, the key has a **state** of *TO_CHECK*. You must validate the partner public key explicitly via the GUI or **secadm update_key** command, by setting its state to *ENABLED*. To refuse the key, you can set its state to *DISABLED* or delete it from the database.

Unlike a certificate, a public key does not contain any information on its owner. The only reliable method to validate a key is to manually check the value of the key or its fingerprint with external (telephone or mail) confirmation from the partner in question.

To facilitate exchange of this value, you can use a fingerprint of the public key. The fingerprint is a hash of the key on 32 octets (using the SHA256 algorithm). The advantage of a hash is that you only have 32 octets to check as opposed to 404 octets for a DSS 1024-bit key or for an RSA 1024-bit key or 256 octets for a ECDSA_NISTP521 key

Trusting hosted keys

The parameter trust_hosted_public_keys allows the acceptance of the authentication of a partner if a public key (or a certificate) belonging to it is already present in the Gateway database in ENABLED state. The term **hosted** applies to a public key (or a certificate) hosted in the Gateway database, and for which the origin (owner of the associated private key) is known and authorized to connect via SSH.

When the parameter trust_hosted_public_keys is activated, Gateway searches for a hosted key equivalent to the key received.

The following three conditions must be fulfilled to accept a connection:

- The two keys have the same fingerprint
- The hosted key is in ENABLED state
- If a group is completed in the authentication control, only a key in this group is trusted

Related topics

SSH protocol

Using SSH

Managing SSH Security Profiles

SSH authentication

Managing SSH Security Profiles (command line)

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Using SSH

Axway Gateway: Managing Security

Installing SSH

Configuring CGate objects for SSH use

Configuring Remote Sites for SSH use

Installing SSH

To install SSH processing on Gateway, select the SFTP file transfer protocol in the configuration menu.

Configuring CGate objects for SSH use

Field	Description
Public key alias	Partner public key Defines the public key to use to authenticate the partner.
Public key group	Key group name Defines the group of the public key to use to authenticate the partner.
Subject certificate alias	Partner certificate Defines the certificate to use to authenticate the partner.
Issuer certificate alias	Issuer certificate name Defines the issuer (CA) certificate to use to authenticate the partner certificate.
Subject name pattern	Partner distinguished name pattern Defines a filter template that the subject certificate must follow to be authenticated.
Issuer name pattern	Issuer distinguished name pattern Defines a filter template that the issuer certificate must follow to be authenticated.

Strong ciphers / weak ciphers

Ciphers using the CBC

Ciphers using the CBC (Cipher-Block Chaining) are considered weak, and their use is **strongly discouraged**.

A configuration option, `[ssh]enable_weak_ciphers` (default set to *yes*) is provided; if set to *no*, the cipher algorithms 3DES_CBC, AES128_CBC, AES256_CBC and NONE are **not available** for the SSH protocol and configuring SSH profiles.

Ciphers using the MD5 hash algorithm

The parameter [monitor]enable_md5_hash_algorithm enables ciphers using the MD5 hash algorithm.

When `enable_md5_hash_algorithm` is set to *no* and a cipher suite using the MD5 hash algorithm is selected, the HMAC_MD5 and HMAC_MD5_96 hash algorithms are longer available for the SSH handshake and in the SSH security profile configuration.

Configuring Remote Sites for SSH use

To use SSH, you must configure Remote Sites to use the SFTP protocol and SSH network security type.

When you enable the SSH network security type for a Remote Site (server), you must define the SSH Profile.

The following table lists the SSH specific parameters to create or update Sites:

Field	Description
Parameters to connect to Remote Site	
SSH Profile	An existing SSH Profile name Specifies which SSH Profile to use with the Remote Site.
User name	Login identifier Specifies the user name you use to identify your Gateway client to the server partner this Site represents.
Password	Login password Specifies the password you use to identify yourself to the server partner.
Authentication alias Only one of these two elements will be used, depending on the public key algorithm selected.	
Private key alias	Local private key alias Defines local private key used for authentication to a partner.
Certificate alias	Local certificate alias Defines local certificate used for authentication to a partner.
Remote Site authentication (initiator and responder mode) Authentication Control parameters used to check the authenticity of a partner	
If the public key algorithm negotiated is a key pair algorithm:	
Public key alias	Partner public key Defines the public key to use to authenticate the partner.
Public key group	Key group name Defines the group of the public key to use to authenticate the partner.
Remote certificate filter (If the public key algorithm negotiated is a certificate algorithm)	
Subject certificate alias	Partner certificate name Defines the certificate to use to authenticate the partner.
Issuer certificate alias	Issuer certificate name Defines the issuer (CA) certificate to use to authenticate the partner certificate.
Subject name pattern	Partner distinguished name pattern Defines a filter template that the subject certificate must follow to be authenticated.
Issuer name pattern	Issuer distinguished name pattern Defines a filter template that the issuer certificate must follow to be

Field	Description
	authenticated.

SSH usage example

To set up SSH for an SFTP transfer:

1. Enable the SFTP option in the Gateway configuration menu.
2. Generate a DSS key with Open SSL. To generate the parameters file and private key dsaparam1024.pem and dsaprv1024.pem (includes a public key), enter the following commands:

```
openssl dsaparam -outform PEM -out dsaparam1024.pem -genkey 1024

openssl dsa -inform PEM -in dsaparam1024.pem -outform PEM -out dsaprv1024.pem
```

3. Import the key manually into the database. To import the generated keys into the database, enter the following commands:

```
secadm import_key  -name GW_DSS_PRV  -fn dsaprv1024.pem  -ff PEM  -algo DSS  -t
PRIVATE  -cts "Generated with openssl DSS 1024"
```

4. Create a CGateGroup and a CGate via the following commands:

```
peladm create_cgategroup  -n CG_SFTP_GR  -state E  -pr SFTP  -sap "*"  -ceda "*"
 -ts TSFTP  -vr /:*:RWD:LRC:Y

peladm create_cgate  -n CG_SFTP  -pg CG_SFTP_GR  -s E  -lu "user1"  -cpo N  -ci "*"
 -cp "user1"
```

5. Import two SSH Security Profiles. In the loop test cases, the public key is specified in the SSH Profile. In test cases towards another server, the public key is not known ahead of time. It will be imported if the remote_certificate_import parameter is set to Y (yes) in the SSH Profile.

```
secadm import_sshprof  -shpf SSHPROF_LOOP_PWD_IN

secadm import_sshprof  -shpf SSHPROF_LOOP_PWD_OUT
```

The section Authentication Alias contains the local authentication information.

The section Authentication Control defines the characteristics of the Remote machine. These parameters are also defined in the Remote Site object.

6. Create a Network Profile. To create a Network Profile to check incoming connections (port 6710 is declared on Gateway for SFTP) use the following commands:

```
secadm create_netprof  -npn NPSSH_ALL  -oa "*.*.*.*"  -da "*.*.*.*/6710"  -sec_opt
SSH  -sprof_name SSHPROF_LOOP_PWD_IN
```

7. Create a Remote Site for a Gateway in Initiator mode to connect to a Gateway SFTP server.

```
peladm create_site  -a SFTP_LP_CLT  -pi SV_SFTP1  -pr SFTP  -nsopt SSH  -sshprf
SSHPROF_LOOP_PWD_OUT  -l_id user1  -l_pswd user1  -ct TCP  -da test.pc.pa.sopra/
6710  -r_max 10  -i_max 10
```

8. Create a Remote Site for a Gateway in Responder mode.

```
peladm create_site  -a SFTP_LP_SRV  -pr SFTP  -nsopt SSH  -sshprf
SSHPROF_PC_PWD_OUT  -chkl_id user1  -chkl_password user1  -ct TCP  -r_max 10
 -i_max 10
```

9. Process a file transfer towards the Site configured in the previous step.

```
peltrans  -oa GW1  -da SFTP_LP_SRV  -fc toto-ban.txt  -appli FTP_B  -mode I  -dir O
```

Related topics

SSH protocol

Managing Keys and Certificates in SSH

Managing SSH Security Profiles

SSH authentication

Managing SSH Security Profiles (command line)

SSH user exits

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Managing SSH Security Profiles

Axway Gateway: Managing Security

An SSH Security Profile contains all security configuration elements used to establish an SSH connection.

To import an SSH Profile, use the secadm import_sshprof online command.

Defining SSH Security Profiles

To add a new SSH Security Profile to the profile database:

1. In the left pane of the GUI main window, click ⊞ to expand the nodes:
   Security Management > Transfer Security Management > Security Profile

2. Right-click the **SSH Profile** sub-node, and then select **New** from the context menu.
   Gateway displays the *New SSH Profile* window.

3. Complete the fields of the following tabs as described below:
   - *General*
   - *Authentication*
   - *PassPort PS*

SSH Profile: General tab

The **General** tab contains the parameters for:

- Key exchange algorithms
- Encryption algorithms
- MAC algorithms
- Public key algorithms

Field	Description
Name	Enter a unique name for the SSH Profile. Maximum: 31 alphanumeric characters. The Profile is referenced with this name in the Remote Site, the Network Profile, and in log messages.
Profile type	This option limits the use of a Profile to incoming or outgoing calls. From the drop-down list, select CLIENT or SERVER.
Enable	Check this option to enable the SSH Profile. A deactivated Profile cannot be used and any operation calling it fails (except for activation).
Key exchange algorithms	Click to access list of available key exchange algorithms. This parameter is negotiated between the client and the server. Select one or both of the following: • DH_GROUP1_SHA1 **(**)** • DH_GROUP_EXCHANGE_SHA1 **(*) (**)** • **DH_GROUP_EXCHANGE_SHA256 (recommended)** • DH_GROUP14_SHA1 **(**)** • ECDH_SHA2_NISTP256 **(*)** • ECDH_SHA2_NISTP384 **(*)** • ECDH_SHA2_NISTP521 **(*)** • CURVE25519 **(*)** **(*)** These cipher suites **cannot** be used with **Secure Relay** if you use **SSH termination in the DMZ.** **(**) SHA1** is a weak key exchange algorithm **(not recommended)** **Selection order :** The default Key Exchange Algorithm selection order is the first Key Exchange Algorithm in common, using the client preferred order. To change this behavior and select Key Exchange Algorithm in server preferred order, please use the global static setting [ssh]ssh_cs_server_order: • **0** (default value): client preferred order (the Key Exchange Algorithm order in the Client message) • **1** server preferred order, the Key Exchange Algorithm order in the SSH server profile

Field	Description
	This parameter also affects the order for the Encryption algorithm, Mac algorithm and Public Key algorithm. **Requirement**: this change of algorithms order requires a Client version of at least Gateway 6.16.1-SP9 or Gateway 6.17.3-SP5.
Encryption algorithms	Click to access the list of available encryption algorithms. This parameter is negotiated between the client and the server. Select one or more of the following: • NONE • AES128_CBC • 3DES_CBC • AES256_CBC • AES_128CTR • AES_256CTR
MAC algorithms	Click to access the list of available MAC algorithms. This parameter is negotiated between the client and the server. Select one or more of the following: • NONE • HMAC-SHA1 **(**)** • HMAC-MD5 • HMAC-SHA1-96 **(**)** • HMAC-MD5-96 • HMAC_SHA256 **(**) SHA1** is a weak key exchange algorithm (**not recommended**) **Note** Restrictions apply to this list in FIPS mode. For more details see Using FIPS.
Public key algorithms	Click to access the list of available public key algorithms. The server uses this algorithm to authenticate itself to the client. The client only uses it if the negotiated Authentication method is public key (refer to **Client authentication methods**). This parameter is negotiated between the client and the server. Select one or more of the following: • SSH-DSS • SSH-RSA • X509V3-SIGN-RSA (*) (see below) • ECDSA_NISTP256 • ECDSA_NISTP384 • ECDSA_NISTP521

Field	Description
	• ED25519 (**) • SSH_RSA_SHA256 **(*)** • SSH_RSA_SHA512 **(*)** **(*) This key algorithm cannot be used with Secure Relay if you use SSH termination in the DMZ.** **(**) This key algorithm cannot be used with Secure Relay if you use SSH termination in the DMZ, nor with Passport PS as key repository**
Client authentication methods	This option defines the client authentication method(s). In server mode, the client must authenticate itself by one of the methods defined. In client mode, authentication occurs as requested by the server, on condition that the requested method is available. From the drop-down list, select one of the following: • Anonymous • Password • Public key • Password or Public key • Password and Public key
Key renegotiation Specify the time and data volume limits for SSH renegotiation. Gateway initiates a key renegotiation when either of these two limits is reached. **Notes:** • The quantities in MB correspond to SI recommendations: 1 MB = 1 000 000 bytes and 1 GB = 1000 MB. • The default values correspond to the maximum recommended limits defined by the RFC. • The parameters are valid for both the server and the client. • Even if these parameters are set to 0, Gateway responds to Key renegotiation requests.	
Minutes before New Keys	Specify the time limit for SSH renegotiation. Value between 0 and 1440 (24 hours). 0 = No limit. Default: 60 (1 hour).
MB before New Keys	Specify the data volume limit for SSH renegotiation. Value between 0 and 2000 (2 GB). 0 = No limit. Default: 1000 (1 GB).
GEX no compatibility mode	The type of request to use in the key exchange, when Gateway is SSH client and one of the DH group exchange algorithms is selected: • checked: use SSH2_MSG_KEX_DH_GEX_REQUEST • unchecked (default): use SSH2_MSG_KEX_DH_GEX_REQUEST_OLD

Field	Description
	Note most SSH servers (Gateway included) accept both types of requests for compatibility reasons. So using the default should be OK. But some implementers (such as OpenSSH 6.9 and newer) completely removed the OLD version of the request. **Note** this setting does not impact the SSH with XSR security termination case
Exit scheduling	Check this option to indicate that the Security server must schedule user exits when this Profile is used (the configuration file parameter exit_scheduling must also be set to Yes).
Data compression	Check this option to select SSH data compression. When this option is selected, the SSH parameter compression_algos = ZLIB NONE. Otherwise, compression_algos = NONE. When ZLIB is negotiated between partners (available on both sides), data is compressed in transfers. If not, no data is compressed.

SSH Profile: Authentication tab

The fields on this tab are only available if you select one or more Public key algorithms on the **General** tab. You can complete the **Local authentication** fields in both the SFTP Sites and SSH Profiles. However, it is highly recommended that you complete these fields in the:

- SSH Profile in *server mode*
- Remote Site object in *client mode*

In addition, the fields in the **Remote authentication** section are only available if you set the **Client authentication method** to include Public key.

Also refer to SSH authentication.

Field	Description
Local authentication This section is used for local authentication to the partner. The authentication mechanism will use either the private key alias or the certificate alias, depending on the public key algorithm negotiated.	
Private key alias	Enter the alias of your private key. Maximum: 31 alphanumeric characters. Alternatively, select a private key alias from the drop-down list. You must ensure that the algorithms provided in the list of **Public key algorithms** of the SSH Profile are compatible with the key that you specify here. Gateway provides the public part of the key to the partner if the public key algorithm negotiated is SSH_DSS, SSH_RSA, ECDSA_NISTP256, ECDSA_NISTP384 or ECDSA_NISTP521.
Certificate alias	Enter your certificate alias. Maximum: 31 alphanumeric characters. Alternatively, select a certificate alias from the drop-down list.

Field	Description
	Gateway uses this certificate if the public key algorithm negotiated is X509V3_SIGN_RSA. If you do not enter a certificate name, you must not include the algorithm X509V3_SIGN_RSA in the list of **Public key algorithms** of the SSH Profile.
Remote authentication	
Trust hosted public keys	The term **hosted** applies to a public key (or certificate) hosted in the Gateway database. The origin (owner of the associated private key) is known and is authorized to connect via SSH. Check this option to accept authentication of a partner if any public key (or certificate) belonging to the partner is already present in the Gateway database in the state ENABLED. Refer to Trusting hosted keys.
Automatic import of partner keys	Check this option to allow the automatic import of unknown public keys presented by partners that are authenticating themselves for the first time (or that have changed keys). Before any key import operation, Gateway checks (with the hash) that the key does not already exist in the database. Refer to Importing key pairs automatically.
Remote certificate filter The following parameters are used to check the authenticity of a partner. Complete these fields if the public key algorithm negotiated is a certificate algorithm.	
Issuer certificate alias	From the drop-down list, select the issuer (CA) certificate alias to use to authenticate the partner certificate.
Subject name pattern	Enter the subject distinguished name pattern. Maximum: 512 alphanumeric characters. You can use wildcard characters (* and ?) to filter accepted certificates.
Issuer name pattern	Enter the issuer distinguished name pattern. Maximum: 512 alphanumeric characters.
Trust hosted certificates only	Similar to **Trust hosted public keys**. Refer to SSH authentication.
Automatic import of partner certificates, depending on SSH Profile and Site filters	Similar to **Automatic import of partner keys**. Refer to SSH authentication.

SSH Profile: PassPort PS tab

Enter values in the fields of the **PassPort PS** tab if you are using PassPort security functions for connections with partners defined in PassPort PS.

Field	Description
Local entity	

Field	Description
Name	Enter the local entity name for access to the PassPort PS server.
Set password	This button opens a dialog box, where you enter the password associated to the entity name.
Confirmation	Confirm the password.
Object Type	Passport PS object type of the entity. Can be one of the following: • *Passport PS entity* • *Passport PS certificate* For profiles used with the XSR termination, the only valid selection is *Passport PS entity.*
Partner entity	
Name	Identity of the PassPort partner as provided by the PassPort PS server.
Object Type	Passport PS object type of the entity. Can be one of the following: • *Passport PS entity* • *Passport PS certificate* For profiles used with the XSR termination, the only valid selection is *Passport PS entity.*

Related topics

SSH protocol

Managing Keys and Certificates in SSH

Using SSH

SSH authentication

Managing SSH Security Profiles (command line)

SSH user exits

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

SSH authentication

Axway Gateway: Managing Security

This topic describes the server and client authentication processes in chronological order.

Server authentication

Authentication is the primary function of the SSH protocol. Although the client always initiates the connection, the server authentication occurs first. The server uses an RSA, ECDSA or DSS key or an X.509 certificate for its authentication.

When Gateway is server:

Gateway uses a parameter from the Authentication Alias section of the SSH Profile selected during the incoming connection. The two options are **Private key alias** and **Certificate alias**.

[AuthenticationAlias]

private_key_alias = "KEY_SRV01_PRV"

certificate_alias = ""

If the negotiated public key algorithm is a...	Then...
Key algorithm • SSH_DSS • SSH_RSA • ECDSA_NISTP256 • ECDSA_NISTP384 • ECDSA_NISTP521	Gateway uses a private key defined in the **Private key alias** field to authenticate itself.

Note: At least one of the available private keys must have the same algorithm as the key negotiated. Therefore any key algorithm provided in the list of **public key algorithms** in the SSH Profile must have a compatible **Private key alias**. Otherwise, the Security server will cancel the connection and return an error.

Certificate algorithm • X509V3_SIGN_RSA	Gateway uses the certificate indicated in the **Certificate alias** field.

Note: If the **Certificate alias** is not completed, you must not include the algorithm X509V3_SIGN_RSA in the list of **public key algorithms** in the SSH Profile. Otherwise, the Security server will cancel the connection and return an error.

Client verification

The client receives a public key (or a certificate) from the server.

When Gateway is client:

The verification is performed with the Authentication Control.

Set the following parameters in the Remote Site SFTP tab that represents the server:

- **Public key alias**
- **Public key group**
- **Subject certificate alias**
- **Issuer certificate alias**
- **Subject name pattern**
- **Issuer name pattern**

Note: You can also set the Authentication Control parameters in the SSH Profile used for the connection. However the Site object parameters have priority over the same parameters contained in the SSH Profile. The SSH Profile only supplies the fields that are not completed in the Site.

Set the following parameters (used in both client and server mode for partner verification) in the SSH profile:

- *Trust hosted public keys*
- *Automatic import of partner keys*

In client mode: Verification procedure

If you are using keys rather than certificates, Gateway handles processing in two ways:

Case 1: You previously specified the **Public key alias** in the Authentication Control of the Remote Site.

Gateway receives a key from a Remote Server and compares the received key with the key (defined by the **Public key alias**) that you specified in the Remote Site (a hosted public key).

- If the keys match, Gateway accepts the connection.
- Otherwise, it returns an error.

Case 2: You did not previously specify a **Public key alias** in the Authentication Control of the Remote Site.

Gateway receives a key from a Remote Server and checks the setting of **trustHostedPublicKeys**.

- If **trustHostedPublicKeys** is set to yes, Gateway searches for a hosted key equivalent to the one it is receiving, and accepts the connection if they match. Otherwise it disconnects.
- If **trustHostedPublicKeys** is set to no, Gateway disconnects.

Before Gateway disconnects, it checks the setting of Automatic import of partner keys and imports the key if this parameter is enabled and the same key is not already in the database.

Client authentication

If the client's server verification is successful, it is the client’s turn to authenticate itself.

When Gateway is client:

The client always sends a User name to the server, regardless of the Authentication method defined. The server defines the Authentication method, previously specified in the General section of the SSH Profile.

- **User to send** (-login_ident) in the SFTP Remote Site

In addition to the user name, the client also sends:

- **Password to send** (-login_password) of the SFTP Site, if the server requests the PASSWORD method
- **Key** (the public part of the private key specified in **Private key alias**) if the server requests the PUBLIC_KEY method

Use the **Authentication Alias** of the SFTP Site (or of the SSH Profile if it is not completed in the SFTP Site).

[AuthenticationAlias]

private_key_alias = "KEY_CLI01_PRV"

certificate_alias = ""

If the negotiated public key algorithm is a...	Then...
Key algorithm • SSH_DSS • SSH_RSA • ECDSA_NISTP256 • ECDSA_NISTP384 • ECDSA_NISTP521	Gateway uses a private key defined in the **Private key alias** field to authenticate itself.

Note: At least one of the available private keys must have the same algorithm as the key negotiated. Therefore any key algorithm provided in the list of **public key algorithms** in the SSH Profile must have a compatible **Private key alias**. Otherwise, the Security server will cancel the connection and return an error.

Certificate algorithm • X509V3_SIGN_RSA	Gateway uses the certificate indicated in the **Certificate alias** field.

Note: If the **Certificate alias** is not completed, you must not include the algorithm X509V3_SIGN_RSA in the list of **public key algorithms** in the SSH Profile. Otherwise, the Security server will cancel the connection and return an error.

Server verification

The server receives a public key (or a certificate) from the client.

When Gateway is server:

The process is slightly different from when a client checks the authenticity of the server, because the Security server does not recognize the SFTP Site associated with the client. It therefore sends a login request to the supervisor to obtain from the Site the data required to authenticate the client.

The supervisor uses CGates and Template Sites to process this request, depending on the client login user name.

In return, the supervisor provides the Security server with the authentication control data found in the Site. If the user is not known or not authorized, the supervisor can also return a negative response and end the connection.

Gateway performs the verification using the Authentication Control structure from the Remote Site object representing the client (the one that the supervisor returned).

Note: Authentication Control is also available in the SSH Profile selected on connection, but not recommended since it is not partner-specific in most implementations. The SSH Profile only supplies the fields that are not completed in the Site.

Set the following parameters in the Remote Site SFTP tab representing the client:

- **Username to check** (-check_login_ident)
- **Password to check** (-check_login_password)
- **Public key alias**
- **Public key group**
- **Subject certificate alias**
- **Issuer certificate alias**
- **Subject name pattern**
- **Issuer name pattern**

The following parameters (used in both client and server mode for partner verification) are configured in the SSH Profile:

- *Trust hosted public keys*
- *Automatic import of partner keys*

In server mode: Verification procedure

If you are using keys rather than certificates, Gateway handles processing in two ways:

Case 1: You previously specified the **Public key alias** in the Authentication Control of the Remote Site.

Gateway receives a key from a Remote Server and compares the received key with the key (defined in the **Public key alias**) that you have defined in the Remote Site (a hosted public key).

- If the keys match, Gateway accepts the connection.
- Otherwise, it returns an error.

Case 2: You did not previously specify a **Public key alias** in the Authentication Control of the Remote Site.

Gateway receives a key from a Remote Server and checks the setting of Trust hosted public keys.

- If **Trust hosted public keys** is set to yes, Gateway searches for a hosted key equivalent to the one it is receiving, and accepts the connection if they match. Otherwise it disconnects.
- If **Trust hosted public keys** is set to no, Gateway disconnects.

Before Gateway disconnects, it checks the setting of Automatic import of partner keys and imports the key if this parameter is enabled and the same key is not already in the database.

Related topics

SSH protocol

Managing Keys and Certificates in SSH

Using SSH

Managing SSH Security Profiles

Managing SSH Security Profiles (command line)

SSH user exits

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Managing SSH Security Profiles (command line)

Axway Gateway: Managing Security

secadm create_sshprof

secadm update_sshprof

secadm import_sshprof

secadm delete_sshprof

secdsp display_sshprof

secdsp select_sshprof

secdsp status_sshprof

secbase export_sshprof

Importing an SSH Profile: secbase import

SSH Profile text file format

Creating an SSH Profile: secadm create_sshprof

Syntax	**`secadm create_sshprof {-sshprof_name (-shpf)}`**
Description	Use this command to create an SSH Security Profile .

Parameters	*Mandatory*
	-sshprof_name (-shpn): Enter the name of the SSH Profile to create.
	Maximum: 31 characters.
	Notes:
	• Although specifying the public key algorithms and the authentication details are not mandatory, it is **highly recommended** that you do so. An SSH profile without the accepted public key algorithms and without the authentication details is not usable.
	• To get the complete parameters list, execute the command: **secadm create_sshprof ?**
	• If `client_auth_methods` has both `PASSWORD` and `PUBLIC_KE`Y set:
	  • if perform_all_auth_method is set to ***N***, the authentication method translates in Password **or** Public key
	  • if `perform_all_auth_method` is set to ***Y***, the authentication method translates in Password **and** Public key

Example	The following command creates an SSH Profile of type SERVER which accepts an RSA key: **`secadm create_sshprof –shpn “sshprof1” –shpt “SERVER” –sshppka “SSH_RSA”`**

Updating an SSH Profile: secadm update_sshprof

Syntax	**`secadm update_sshprof {-sshprof_name (-shpn)}`**
Description	Use this command to modify the parameters of a SSH Profile.
Parameters	*Mandatory* -sshprof_name (-shpn): Enter the name of the SSH profile to update. **Note**: the name of the SSH profile cannot be changed. **Note**: to get the complete parameters list, execute the command: **`secadm update_sshprof ?`**
Example	The following command modifies an SSH Profile to extend the list of accepted cipher suites and specify authentication details (private key): **`secadm update_sshprof –shpn "sshprof1" -shpprvka “RSA_KEY”`**

Importing an SSH Profile: secadm import_sshprof

Syntax	**secadm import_sshprof {-sshprof_file (-shpf)} {-encryption_key_file(-ekf)} {-import_pwd_cleartext}**
Description	Use this command to import an SSH Security Profile from a text file into the local database.
Parameters	*Mandatory* -sshprof_file (-shpf): Enter the name of the file that contains the SSH Profile to import.
	encryption_key_file (-ekf): Enter the path to the encryption key file to be used to decrypt passwords contained by the object to be imported OR -import_pwd_cleartext: Specifies a legacy import file containing unencrypted passwords. **Notes:** • Only one of the -encryption_key_file or -import_pwd_cleartext options should be used. Using one of the parameter is mandatory. • To get the complete parameters list, execute the command **secadm import_sshprof ?**
Example	To import an SSH Security Profile contained in the file sshprof_desc_file.txt, enter: **secadm import_sshprof –shpf <sshprof_desc_file.txt> -ekf <decryption_key.dat>** The Profile is described in the text file provided in the **-shpf** parameter, in this case: <sshprof_desc_file.txt>. All of the Profile parameters are in this file. The decryption key is specified in the -ekf parameter. When importing an SSH Security Profile export file containing clear-text passwords, generated with an older version, use the -import_pwd_cleartext parameter instead.

Deleting an SSH Profile: secadm delete_sshprof

Syntax	**`secadm delete_sshprof {-sshprof_name (-shpn)}`**
Description	Use this command to delete an SSH Security Profile from the local database.
Parameters	*Mandatory* -sshprof_name (-shpn): Enter the name of the SSH Profile. Maximum: 31 characters. **Note**: to get the complete parameters list, execute the command: **`secadm delete_sshprof ?`**
Example	The following command deletes an SSH Profile named *sshprof1*: **secadm delete_sshprof -shpn sshprof1**

Displaying an SSH Profile: secdsp display_sshprof

Syntax	**`secdsp display_sshprof {-sshprof_name (-shpn)}`**
Description	Use this command to display an SSH Security Profile in the database.
Parameters	*Mandatory* -sshprof_name (-shpn): Enter the name of the SSH Profile. Maximum: 31 characters. **Note**: to get the complete parameters list, execute the command: **`secadm display_sshprof ?`**
Example	The following command displays the Security Profile named SSHPROF_CLI_01: secdsp display_sshprof -shpn sshprof_cli_01 Result: [General] shp_name = 'SSHPROF_CLI_01' shp_type = 'CLIENT' shp_state = 'ENABLED' shp_exit_scheduling = 'N' shp_key_exchange_algos = 'DH_GROUP1_SHA1' shp_cipher_algos = 'AES128_CBC' '3DES_CBC' 'AES256_CBC' shp_MAC_algos = 'HMAC_SHA1' shp_compression_algos = 'ZLIB NONE' shp_public_key_algos = 'SSH_DSS' shp_client_auth_methods = 'PASSWORD' shp_perform_all_auth_method = 'N' shp_trust_hosted_public_keys = 'Y' shp_remote_key_import = 'Y' shp_trust_hosted_certificates = 'Y' shp_remote_certificate_import = 'Y' shp_MBytes_Before_New_Keys='1000' shp_minutes_Before_New_Keys='60' shp_xpp_entity_name = '' shp_xpp_entity_password = '' shp_xpp_object_type='Entity' shp_xpp_partner_entity_name = '' shp_xpp_partner_object_type='Entity' [AuthenticationAlias] shp_private_key_alias = 'KEY_CLI01_PRV' shp_certificate_alias = '' [AuthenticationControl] shp_issuer_certificate_alias = '' shp_subject_name_pattern = '' shp_issuer_name_pattern = ''

Syntax	**`secdsp display_sshprof {-sshprof_name (-shpn)}`**
	Note: The format is similar to the description file that you import, except that the fields have the prefix shp_.

Listing SSH Profile names: secdsp select_sshprof

Syntax	**secdsp select_sshprof**
Description	Use this command to list the names of all existing SSH Profiles in the database.
Parameters	None
Example	Enter the command: **secdsp select_sshprof** Result: SSHPROF_CLI_001 SSHPROF_SRV_001

Listing SSH Profile abstracts: secdsp status_sshprof

Syntax	**`secdsp status_sshprof`**
Description	Use this command to display the status of each SSH Profile in the database.
Parameters	None
Example	Enter the command: **secdsp status_sshprof** Result: Name Type Exit Status SSHPROF_CLI_001 CLIENT N ENABLED SSHPROF_SRV_001 SERVER N ENABLED

Exporting SSH Security Profile objects: secbase export_sshprof

Syntax	**secbase export_sshprof** {-encryption_key_file} [-format] [-user_fmt] [-output] [-append]
Description	Use this command to export the fields and field values of one or more SSH Security Profiles (SSHProfs). Gateway either displays the results on the screen or copies the results to an ASCII file. Use SSHProf attributes as selection criteria, as described below.
Parameters	*Mandatory* -encryption_key_file(-ekf): Enter the path to the encryption key file to be used to encrypt passwords contained by the object.
	Optional
	-format (-of): Output format.

Syntax	**secbase export_sshprof** {-encryption_key_file} [-format] [-user_fmt] [-output] [-append]
	Enter one of the following values: • S = Shell (Default). Example: field = *value* • C = C-Shell. Example: set field = *value* • U = User format
	-user_fmt (-uf): User format definition. Enter a character string including the keywords %N and %V, where: • %N: name of field • %V: value of field
	-output (-f): Name of the file that results from the **secbase export_sshprof** command. If you do not specify this parameter, the standard output file is used. Maximum: 127 alphanumeric characters.
	-append (-ap): Concatenation at the end of the file (if it exists). This parameter takes no value.
Example	Enter the command: **secbase export_sshprof -ekf <encryption_key.dat> -n OldSSHProf -f ExpSSHProf** Gateway creates the ExpSSHProf export file. From the SSHProf object named OldSSHProf, it extracts all fields and their corresponding values, and writes them to the ExpSSHProf file. The encryption key file <encryption_key.dat> is used to encrypt passwords contained by the SSH Security Profile.

Importing an SSH Profile: secbase import

Syntax	secbase import {-input(-if)} [-encryption_key_file(-ekf)] [-import_pwd_cleartext] [-sshprof(-shp)] [-error_free]
Description	Use this command to import an SSH Security Profile from a text file into the local database.
Parameters	*Mandatory*
	-sshprof (-shp): Enter this parameter without entering a value to import all SSH Security Profile objects contained in the file specified in the input parameter.
	-encryption_key_file (-ekf): Enter the path to the encryption key file which will be used to decrypt passwords contained by

Syntax	secbase import {-input(-if)} [-encryption_key_file(-ekf)] [-import_pwd_cleartext] [-sshprof(-shp)] [-error_free]	
	the object to be imported. OR -import_pwd_cleartext : Specifies a legacy import file which contains unencrypted passwords. **Note** Only one of the -encryption_key_file or -import_pwd_cleartext options should be used. Using one of the parameters is mandatory.	
	Optional	
	-error_free: Enter this parameter to create the different objects while ignoring the new parameters and without stopping the process at each error.	
	Example	To import an SSH Security Profile contained in the file sshprof_desc_file.txt, enter: secbase import –if <sshprof_desc_file.txt> -ekf <decryption_key.dat> -shp The Profile is described in the text file provided in the -if parameter, in this case: <sshprof_desc_file.txt>. All of the Profile parameters are in this file. The decryption key is specified in the -ekf parameter. When importing an SSH Security Profile export file containing cleartext passwords, generated with an older version, use the -import_pwd_cleartext parameter instead.

SSH Profile text file format

Import SSH Security Profiles via a file that contains all the fields described in the Security Profile section. All unknown fields are ignored. If a field is absent or if the value is not correct, an error message is displayed and the import is aborted.

Example of an SSH Profile text file

[General]
shp_name='SSHPROF_CLI_001'
shp_type='CLIENT'
shp_state='ENABLED'
shp_exit_scheduling='N'
shp_key_exchange_algos = 'DH_GROUP_EXCHANGE_SHA256'
shp_cipher_algos = 'AES256_CTR' 'AES128_CTR'
shp_mac_algos = 'HMAC_SHA256'
shp_compression_algos = 'NONE'
shp_public_key_algos = 'SSH_DSS'
shp_client_auth_methods = 'PASSWORD'
shp_perform_all_auth_method='N'
shp_trust_hosted_public_keys='N'
shp_remote_key_import='N'
shp_trust_hosted_certificates='N'
shp_remote_certificate_import='N'
shp_MBytes_Before_New_Keys='1000'
shp_minutes_Before_New_Keys='60'
shp_xpp_entity_name='passport_entity'
shp_xpp_entity_password='hPN6ANc5uWiHZB0yU48dq6nR/gNwWeag0aJ9Bl03oYBKsiVR3dHpms1
KB2+I7o0NAAIDAgMACg8KDw=='
shp_xpp_object_type='Entity'
shp_xpp_partner_entity_name=''
shp_xpp_partner_object_type='Entity'

[AuthenticationAlias]
shp_private_key_alias=''
shp_certificate_alias=''

[AuthenticationControl]
shp_issuer_certificate_alias=''
shp_subject_name_pattern=''
shp_issuer_name_pattern=''

Related topics

SSH protocol

Managing Keys and Certificates in SSH

Using SSH

Managing SSH Security Profiles

SSH authentication

SSH user exits

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

TLS

Axway Gateway: Managing Security

About the TLS protocols

Using the TLS with transfer protocols

TLS instance

Incoming connection

Outgoing connection

TLS Extensions

Related topics

Managing TLS

Using TLS

Managing TLS Security Profiles

Managing TLS Security Profiles (command line)

About the TLS protocol

These TLS protocol operates between network protocol layers (for example, TCP/IP...) and application protocol layers (for example, HTTP). They both provide the following services:

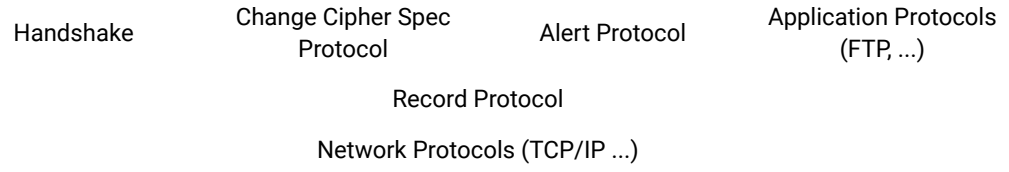
- *Authentication*
- *Integrity*
- *Privacy*

TLS is based on four protocols:

- **Handshake Protocol**: sets up the client and server session
- **Change Cipher Spec Protocol**: establishes new agreement on the cipher suite used for the session
- **Alert Protocol**: transmits warning and error messages between client and server
- **Record Protocol**: comprises the above protocols and the application protocols

Protocol stack

When you use, the protocol stack can be represented as follows:



The **Record Protocol layer** is responsible for:

- Splitting data (maximum data block size is 16 KB)
- Compressing data (optional)
- Adding an integrity checksum (MAC)
- Encrypting data

To establish a TLS connection, the connection procedure:

- Sets up the record protocol layer: the layer is initialized with no integrity check and no encryption method (record layer state is then referred to as *null with null null*)
- Executes the "handshake" protocol: client and server exchange their certificates and agree on an encryption method
- Executes the "change cipher" protocol: following this exchange, the record layer is reconfigured using the security parameters exchanged during the handshake

To test the connection, a *finish* message is sent to test the established security parameters. The message contains a checksum of all the data sent since the beginning of the "handshake".

Payload splitting

Gateway supports payload splitting for the following TLS handshake messages:

- CERTIFICATE, containing as payload Certificate Chains
- CERTIFICATE REQUEST, containing as payload Authority DN lists

Gateway can thus handle messages of this type with payloads up to 64KB. The payload is split into several TLS handshake messages sent (or received) in a row, each carrying a fragment of the total payload not more than 16KB in size.

Note Gateway does not support TLS Handshake messages longer than 16KB, that are not RFC-compliant. Be aware of the fact that current versions of multiple third party software such as openSSL, fileZilla, Mozilla and others, implement support for long (>16KB) Certificate Chains and Authority DN lists, by using handshake messages longer than 16KB in order to accommodate the entire size of the payload.

Using TLS with transfer protocols

Use of TLS is transparent for application protocols (for example, HTTPS, FTP, OFTP, and PeSIT). In most cases the TLS negotiation is triggered as soon as network connection is established.

Note: To secure a protocol with TLS you must use a different **SAP** (Service Access Point) than used for a

connection in that protocol that is not secure.

TLS and FTP

FTP implements the use of TLS by introducing commands that start a TLS session on both the client and the server.

You can configure Gateway to use TLS in implicit and explicit mode for an FTP connection using the parameter **ft_sec_option**. In the implicit mode the session starts directly in a secured mode.

An FTP session involves two connections:

- Control connection
- Data connection

Both connections must be secured.

You can establish the data connection in either of the two following modes:

- Passive mode (the client initiates it)
- Active mode (the server initiates it)

In passive mode, the data connection:

- Uses TLS session caching
- Reuses the security parameters negotiated for the control connection (there is a single handshake for both control and data connections)

TLS and SMTP/POP3

SMTP implements the use of TLS by introducing commands that start a TLS session on both the client and the server.

SMTP servers are usually secured using TLS explicitly and most will not accept an implicit TLS connection. However an implicit TLS connection is possible with some SMTP servers. As well, SMTP servers for a proxy mailer may impose TLS in implicit mode.

You can configure Gateway to use TLS in implicit and explicit mode for an SMTP or POP3 connection using the parameter **-ft_sec_option**.

POP3 can be secured using implicit TLS if the server requires it. Since implicit TLS connections are established during the protocol connection, immediate TLS negotiation is imposed.

TLS and secure protocols

Some protocols are already secured. If you use TLS with those protocols, then the machine load is increased, but security is not increased. Recent protocols (or recent protocol extensions) implement the use of TLS by introducing commands that start a TLS session on both the client and the server (for example, SMTP, FTP). When you use TLS with FTP, all sessions (control and data) are encrypted, and, if possible, security parameters are reused to avoid a completely new handshake.

TLS instance

As described in the TLS protocol, a TLS session involves two steps:

- A handshake session, where the communicating partners exchange security parameters
- The data transfer itself

You can configure cryptographic and authentication parameters via the TLS Profile object. Gateway selects the Profile to use for a TLS session depending on the way the connection between the partners is established (incoming or outgoing connection).

TLS handshake protocol initialization

You can impose the following handshake parameters via the **TLS Security Profile** configuration:

- Negotiation/Renegotiation
- Cache
- Cipher suites
- Protocol version

Incoming connection

An incoming connection in TLS has a minimum of two levels of verification. The first is identification of a corresponding Network Profile object and the second is the identification and check of the Remote Site object and its parameters.

Retrieving the Network Profile object (netprof)

For incoming connections, the network parameters are used to retrieve a Network Profile object. The Network Profile object indicates:

- Whether to use TLS for this connection (**Network security** option set to **TLS**)
- The name of the Security Profile to use

Checking the Remote Site

Following the TLS Handshake and the protocol connection, Gateway locates the Remote Site object involved in the current transfer.

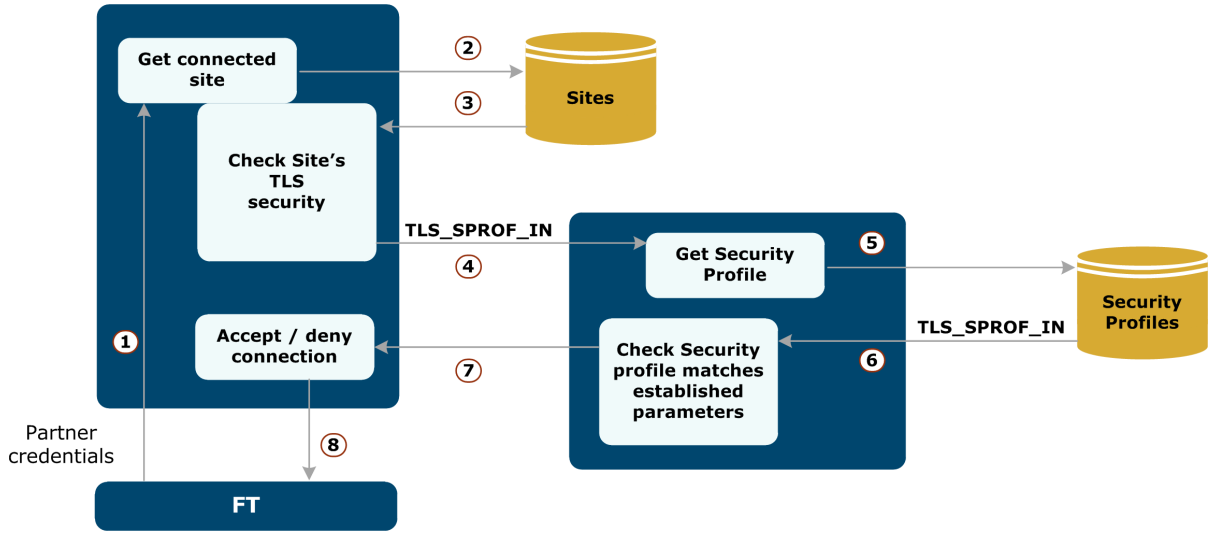
Gateway then checks that the parameters negotiated during the TLS Handshake and the incoming Security Profile fields of the located Site (tls_sprof_in) match. If the tls_sprof_in parameter value is empty, no additional control is processed. If this control fails, the transfer is aborted.

Matching the Security Profile relies on:

- Security Profile type (always server for incoming connections)
- Exit scheduling value
- Client and server authentication
- Cipher suite used

- Protocol used (TLSV1 or SSLV3)
- User certificate sent
- Remote certification chain received

The following diagram illustrates how Gateway implements security for incoming TLS connections:

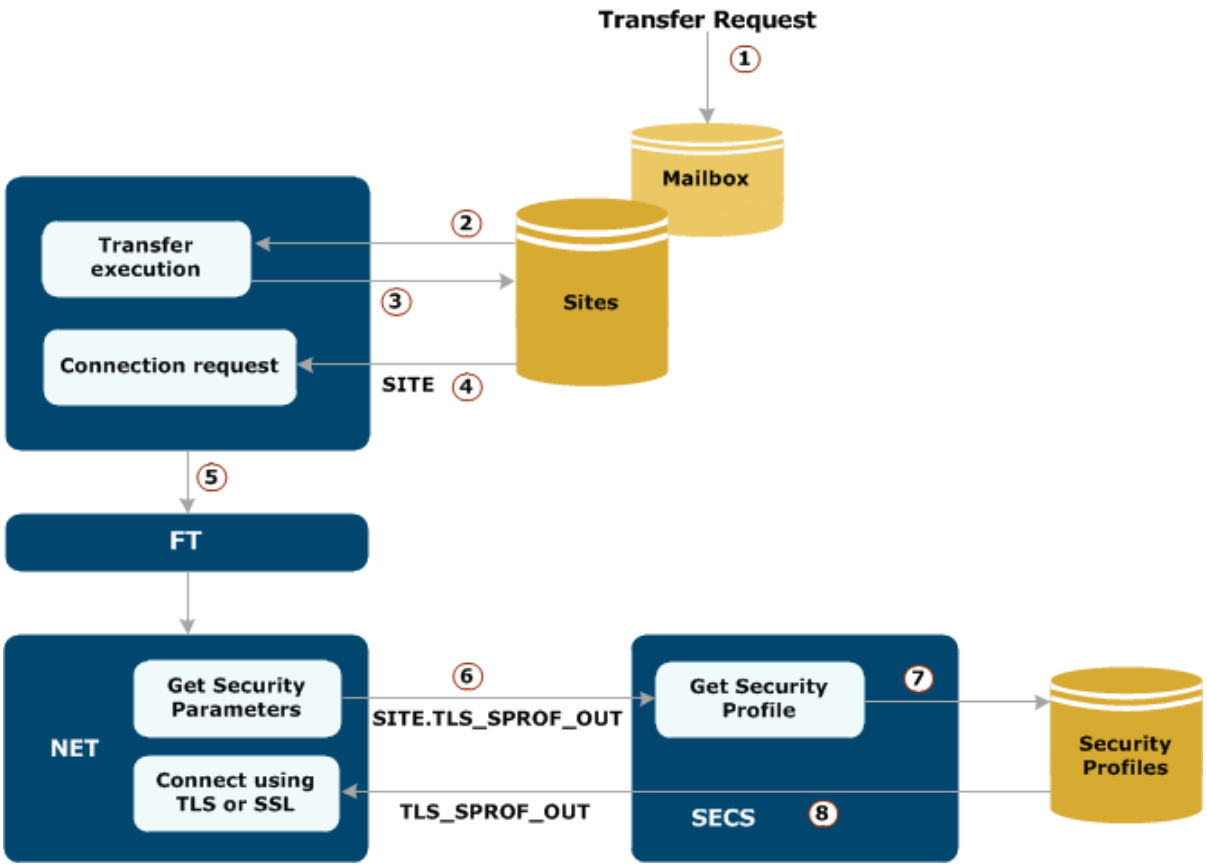


Outgoing connection

The Remote Site configuration determines whether or not to secure an outgoing connection.

If you specified a Security Profile in the outgoing Security Profile field (**tls_sprof_out**) of the Site, then this Security Profile is used to secure the connection.

The following diagram schematically illustrates how Gateway implements security for outgoing TLS connections:



TLS Extensions

When connecting as a client, Gateway uses the host name of the remote partner as the value for the Server Name Indication extension, as specified by the RFC6606 from the IETF.

Gateway uses this value as is, as long as it doesn't resemble an IP address (IPV4 or IPV6), and as long as it is exclusively written with ASCII characters. Also, the RFC requires that the value sent with this extension must be the fully qualified domain name of the server.

Restrictions:

- In server mode, this extension is ignored.
- This extension is sent only when Gateway attempts to negotiate version 1.2 of TLS.
- The `server_name` extension is not added if TLS is done in Secure Relay (XSR termination for outgoing connections)

Related topics

Managing TLS

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release*

Notes

Managing TLS

Axway Gateway: Managing Security

Accepted authorities list

Building a list

To send a trusted authorities DN list to the client, the server refers to the current profile and builds the list according to the corresponding parameters. Each accepted authority is referenced by its local name (its authority certificate must be present in the local database). For each name given, the Security server finds the corresponding certificate and DER-encodes its subject name. It then sends the resulting buffer to the client.

Receiving a list

When the Security server receives the accepted authorities distinguished names list from the server, it parses its local database and determines whether each corresponding authority certificate is present or not. If the authority certificate is present, the certificate reference is memorized for later use when the certification path to send is built. If it is not present, this means that the authority is unknown and cannot be trusted. The distinguished name is ignored.

Certificate chain management

Sending a certificate chain

Server case

The server can send any of the current Security Profile user certificates. If the server can send more than one user certificate, it selects the first certificate in the Security Profile (sprof) **UserCertificate** field.

The Security server builds the corresponding certificate chain step by step. For each certificate, it identifies its

issuer certificate and so on, until it reaches a root certificate.

Each certificate must be activated before you can send it. Otherwise, an error occurs.

To optimize performance, there are no signature, validity, or other checks made on the certificate chain that the server sends. The receiver is responsible for verifying that the certificate chain is correct.

Client case

The client builds a certificate chain that matches the received distinguished name list of accepted authorities received. The available authority references that correspond to those sent are stored in the memory. The Security server refers to the user certificates defined in the current Profile. For each user certificate, the Security server locates the root authority that this certificate is linked to and sends the first matching authority of the known accepted authorities.

Checking a certificate chain

For each received certificate, the Security server checks for the validity of the whole chain. It checks:

- Each signature
- The time validity of each certificate
- Whether matching criteria are defined in the Profile (template matching). Each received certificate must verify one or more of them.
- The root certificate that ends the certificate chain is known and present in the local database. If not, the received certificate chain is considered invalid.

Server authentication policy

Since the TLS protocol indicates that the server must send a certificate whenever the agreed-upon key exchange is not anonymous, the **server_auth_policy** Security Profile field is meaningless. As a result, whatever its value in the Security Profile for the current secured connection, it will be internally forced to *TLS_AUT_MANDATORY*.

For compatibility purposes, this field is still mandatory when you create a new Security Profile. It is recommended to set this parameter to *TLS_AUT_MANDATORY* in the text file that describes the Security Profile.

Client authentication policy

This section describes Gateway behavior depending on the value of the **client_auth_policy** field in the current TLS (or SSL) connection Security Profile.

Client side

As a client, **client_auth_policy** is taken into account and comes from the Remote Site outgoing Security Profile.

If you set **client_auth_policy** to *TLS_AUT_OPTIONAL*, and if it is not possible to build a local certification path from one of the available user certificates (defined in the **UserLocalCertificate** section of the current Profile) to one of the accepted authorities list previously received from the server, then an alert (SSL V3) or an empty certification path (TLS V1) is sent to the server. The server is then responsible for closing the connection or continuing the handshake phase.

If you set the **client_auth_policy** field to *TLS_AUT_MANDATORY*, Gateway sends an alert (handshake failure) and

closes the connection if no valid certification path can be built according to the accepted authorities list received from the server. In this case, connection closure (on the client side) occurs regardless of the client authentication policy on the server side. This option is useful if you want reciprocal authentication for both incoming and outgoing connections (secured with TLS or SSL).

Server side

As a server, the **client_auth_policy** field is taken into account and comes from the Security Profile selected depending on the selected Network Profile (*netprof)* field.

If you set **client_auth_policy** to *TLS_AUT_OPTIONAL*, the server will not close the current connection, regardless of whether the client has succeeded its authentication or not. The TLS (or SSL) handshake phase will continue.

If you set **client_auth_policy** to *TLS_AUT_MANDATORY*, the server closes the current TLS (or SSL) connection if no valid certificate (matching one of the locally accepted authorities list) is received from the client.

In addition, Gateway implements another control after the protocol connection. At this stage, the TLS (or SSL) handshake is finished and Gateway knows whether the current connection client is authenticated or not. Gateway checks the **client_auth_policy** attribute of the Remote Site (corresponding to the connected client) incoming Security Profile. If this parameter is set to *TLS_AUT_MANDATORY* and if the client failed its authentication, the server closes the connection and aborts the transfer.

Note: The incoming Security Profile parameter in the Remote Site is an optional field. If you do not set a value for this field, no additional control is processed.

Defining User Certificate list

This section describes the rules applied when you use several user certificates in Gateway Security Profile configuration.

Public Key Infrastructure technology and certificate-based authentication are relevant for securing data exchanges over the Internet. This means that it may be useful for a TLS client to have several user certificates for its authentication and to select which user certificate to send depending on the accepted authorities list received from the server.

Gateway enables you to define a user certificate list via the Security Profile UserLocalCertificate section definition.

Example

```
name = "SPROFTEST"
type = CLIENT
cache_enabled = Y
renegotiate_enabled = Y
exit_scheduling = N
client_auth_policy = TLS_AUT_MANDATORY
server_auth_policy = TLS_AUT_MANDATORY
tls_versions = TLSv1 SSLv3
path_depth = 7
cipher_suites = RSA_WITH_3DES_EDE_CBC_SHA
[UserLocalCertificate]
certificate_name = "USER1"
certificate_name = "USER2"
[AcceptedAuthorities]
```

In this sample, the User1 certificate is linked to the root certificate ROOT1 and the User2 certificate is linked to the root certificate ROOT2.

Server Side

The TLS (or SSL) server sends its accepted authorities list.

Client Side

If ROOT1 is included in the accepted authorities list received from the server whereas ROOT2 is refused, the User1 certificate will be sent.

If the server refuses ROOT1 and accepts ROOT2, the User2 certificate is sent.

If the server accepts both ROOT1 and ROOT2, the first valid user certificate chain is sent, in this case User1.

Note: When you use a CLIENT_AND_SERVER Security Profile, the server sends the first certificate of the UserLocalCertificate section, since there are no constraints on the server sending one particular certificate rather than another for its authentication.

Using security without certificates

This feature enables TLS-secured communications with encrypted and MAC-controlled data without requiring any X.509 certificates. The client and server are both anonymous. This method uses an RSA anonymous key exchange algorithm. To configure this feature in Security Profiles, use the new set of cipher suites known as TLS_RSA_ANON_WITH_xxx_yyy.

Using security without certificates provides the same high level of encryption and MAC-checking on the data flow, while avoiding the certificate management process.

Principles

An RSA public key provided by the server implements the RSA anonymous key exchange, rather than the RSA key

included in the server certificate.

The "ServerKeyExchange" handshake message transmits the server public key used to encrypt the session key (premaster secret) sent by the client. All certificate-related messages are discarded (Certificate, CertificateRequest and CertificateVerify).

Supported server RSA key lengths: 512, 1024, 2048 4096 and 8192 bits.

New cipher suites

The binary values of the RSA anonymous cipher suites are:

RSA_ANON_WITH_NULL_MD5 { 0xFF, 0x01 }
RSA_ANON_WITH_NULL_SHA { 0xFF, 0x02 }
RSA_ANON_WITH_RC4_128_MD5 { 0xFF, 0x04 }
RSA_ANON_WITH_RC4_128_SHA { 0xFF, 0x05 }
RSA_ANON_WITH_DES_CBC_SHA { 0xFF, 0x09 }
RSA_ANON_WITH_3DES_EDE_CBC_SHA { 0xFF, 0x0A }
RSA_ANON_WITH_AES_128_CBC_SHA { 0xFF, 0x2F }

Security Profile sample

Client side:

name = "PROFILE_CLIENT_1"
type = CLIENT
tls_versions = TLSv1
cache_enabled = Y
renegotiate_enabled = N
exit_scheduling = N
client_auth_policy = TLS_AUT_ANONYMOUS
server_auth_policy = TLS_AUT_ANONYMOUS
path_depth = 2
cipher_suites = RSA_ANON_WITH_AES_128_CBC_SHA
[UserLocalCertificate]
[AcceptedAuthorities]

Server side:

name = " PROFILE_SERVER_1"
type = SERVER
tls_versions = TLSv1
cache_enabled = Y
renegotiate_enabled = N
exit_scheduling = N
client_auth_policy = TLS_AUT_ANONYMOUS
server_auth_policy = TLS_AUT_ANONYMOUS
path_depth = 2
cipher_suites = RSA_ANON_WITH_AES_128_CBC_SHA
[UserLocalCertificate]
[AcceptedAuthorities]

Restrictions

This feature is fully compliant with the TLS specification in RFC 2246. However, it is specific to Axway Transfer products and may not interoperate with other TLS products.

Users should be aware that the lack of authentication may make an attack easier, especially those known as "man in the middle" attacks. As a result, this feature is mostly intended for the case where the overlaying protocol or application provides an authentication or login mechanism.

Disabling close notify messages in TLS

This is a workaround for TLS interoperability. Some FTP clients (CuteFTP for example) report an error when they receive a close notify alert message on the data connection. This message is supposed to be exchanged before closing a connection (refer to *RFC 2246*), thus ensuring that the TLS session parameters can be cached by peers.

The workaround consists in closing TLS connections without sending the close notify message.

However, if a close notify message is received, it is acknowledged with another close notify message.

To *disable* close notify messages, set the parameter **close_notify_disabled** in the *sprof* files to Y.

close_notify_disabled = 'Y'

secadm import_sprof -sprof_file <*filename*>

To *enable* close notify messages (default mode), set the parameter to N.

close_notify_disabled = 'N'

secadm import_sprof -sprof_file <*filename*>

Secure renegotiation in TLS

A TLS renegotiation vulnerability has been discovered in the TLS and the SSL 3.0 protocols which allows a man-in-the-middle (MITM) attacker to inject an unauthenticated request of his choice as a prefix to a victim's session.

Different studies show that protocols on top of TLS like HTTP, FTP or SMTP are all affected to varying degrees, but most of the attacks are documented on HTTPS. The fix for this issue was proposed by the engineering task force as a TLS extension described in the RFC 5746 document.

Gateway fully supports RFC 5746 and the changes can be seen at the Security Profile level and at the Log level as detailed below. The extension will always be active in Gateway since there are no consequences to interoperating with implementations without the upgrade.

Security Profile level

There is one additional TLS profile parameter 'Refuse legacy partners' (or sp_un-upgraded_partner_refused in command line tools) which allows a user to close a connection from a TLS un-upgraded partner from the start (from the initial TLS handshake).

In addition, there is a redesigned parameter - 'Renegotiation enabled' (or sp_renegotiate_enabled in the command line tools) which has increased the number of values from a Boolean type (with yes or no as possible options) to a 4-option type: disabled, unrestricted, secure and legacy.

When dealing with an upgraded partner, the only situation where renegotiation is refused is when it is disabled from the TLS profile.

On the other hand, if connecting to an un-upgraded partner, for compatibility reasons, the response to a renegotiation request will depend on the 'Renegotiation enabled' flag from the TLS profile (although in this case RFC 5746 recommends refusing renegotiations):

- **disabled, secure**: Refuse renegotiation (send the `no_renegotiation` warning alert) and continue the connection.
- **unrestricted**: Both in client and server mode, accept renegotiation handshake. The connection is vulnerable to attacks.
- **legacy**: This option means that Gateway in client mode will accept renegotiation requests, although the connection is vulnerable to attacks. In server mode, Gateway will refuse renegotiation and continue the connection. This setting will force partners to upgrade their client applications to be able to renegotiate with Gateway, but will not interrupt renegotiation when Gateway is in client mode connection to a legacy server.

Although refusing renegotiation is a TLS alert at warning level and the connection can continue afterwards, some implementations are geared towards security with a lack of compatibility and choose to end the connection.

Related topics

Managing TLS Security Profiles

DHE key exchange in TLS

When using DHE cipher suites in server security profiles, Gateway, as TLS server, needs a [P,G] pair of Diffie-Hellman parameters for the ephemeral DH keys generation. The way in which Gateway obtains these parameters is configurable through the command line, using the peluconf command.

The configuration parameters related to DH params generation are as follows:

[monitor]secs_dh_params_mode

Values are from *0* to *2*. Default value is *2*. Meaning of values are:

- *0* - DH params are generated at Gateway startup, using the bitsize indicated in the parameter [monitor]secs_dh_params_bitnum_idx (see below).
  Caution: even for minimum-size DH parameters, Gateway's SECS process may be blocked a long time at Gateway start, depending on the hosts entropy. This option is **to be used with care**.
- *1* - DH params are loaded from a file indicated by the parameter [monitor]secs_dh_params_file.
- *2* - DH params are provided from OpenSSL, and the used values are specified in a RFC (it is the 2048/256 MODP group in http://www.ietf.org/rfc/rfc5114)

[monitor]secs_dh_params_bitnum_idx

Values are from *0* to *2*. Default value is **2**. Parameter has a meaning only if [monitor]secs_dh_params_mode=*0* (generated DH params). Meaning of values are:

- *0* - 1024 bits for P
- *1* - 2048 bits for P
- *2* - 4096 bits for P

Value for DH generator (G) will be 2 or 5, as these are the only supported values in OpenSSL.
Note: the generation in Gateway of parameters longer than 1024 bits may take a very long time (minutes for 2048 bits, or even hours for 4096 bits).

[monitor]secs_dh_params_file

This parameter indicates the file where Gateway is to find the DH params at startup. The file MUST be in DER format.

- This parameter has a meaning only if [monitor]secs_dh_params_mode=*1*, meaning DH mode is *load DH params from file*.

  Note: If error occurs when loading/generating DH params, DHE ciphers cannot be used in TLS server mode.

  Note2: If DHE ciphers are to be used in client mode only, there is no need for DH params, as the client will use the server communicated DH params to generate the client ephemeral keys.

[monitor]secs_dh_params_file

This parameter indicates the file where Gateway is to find the DH params at startup. The file MUST be in DER format.

- Note: If error occurs when loading/generating DH params, DHE ciphers cannot be used in TLS server mode.
- Note2: If DHE ciphers are to be used in client mode only, there is no need for DH params, as the client will use the server communicated DH params to generate the client ephemeral keys.

Avoiding weak cryptography

Using DHE parameters of 2048 bits or larger is strongly recommended. To enforce this security recommendation, Gateway has added a configuration parameter: `[tls]enable_weak_crypto`. This parameter, when set to **no**, effectively prevents the TLS DHE key exchange with an insecure (less than 2048 bits) DHE key size.

If Gateway encounters DH key generation parameters with a P parameter of less than the recommended size:

- a message is logged
- added to that, if `[tls]enable_weak_crypto` is set to **no**, the TLS_DHE ciphers cannot be used in TLS server mode.

Note For compatibility reasons, the parameter's default value is `yes`. We **strongly recommend** to set `[tls]enable_weak_crypto` to **no**, and to set DH key generation configuration parameters accordingly.

Note If DHE ciphers are to be used in client mode, Gateway will apply the `[tls]enable_weak_crypto` parameter on the server communicated DH parameters.

Certification path building mode in TLS

Gaetway has a new way of building certificate paths, both in server and client mode. It is an implementation of the Internet X.509 Public Key Infrastructure: Certification Path Building specification, also known as RFC4158.

This mode of operation can be configured on individual security profiles using the attribute `sprof_path_build_mode`, (short version `sppbm`), having 2 possible values:

- *legacy*
- *rfc4158*

The purpose of this algorithm is to support multiple PKI structures as well as easy certificate renewal across all levels of the certification path. Note that when renewing a certificate, as long as the legacy one is valid, it will continue to be used for securing TLS connections (to ease the transition to the new RFC4158). This however depends on the order in which the certificates are selected on the security profile (this is valid for end user certificates). Renewed intermediary or root certificates are handled transparently when the certification path is built. Basically once one of the certificates expires, it will be ignored by the path building algorithm and the product will attempt to use (re)new(ed) certificates.

One mandatory requirement of this mode of operation is that all certificates MUST have the subject key identifier and authority key identifier extensions (the second extension is optional on root and self signed user certificates). This is not enforced by the product to allow users to operate the product with certificates that might not have these extensions (using the legacy path building mode). If the certificates already present in the product have these extensions, it is possible to use them with the new path building mode.

Local certificates on negotiated cipher: only if compatible

On a negotiated cipher, Gateway selects a local certificate **only** if its key is compatible with the key exchange algorithm of the negotiated cipher.

This allows for multiple-certificate configurations, covering different security requirements: old software that only supports RSA ciphers, alongside new software supporting elliptic curve cryptography.

If Gateway finds no matching certificate, a warning is logged, and the connection continues only if the authentication policy configured on the security profile is optional.

Related topics

Managing TLS Security Profiles

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Using TLS

Axway Gateway: Managing Security

Installing TLS

To install TLS, select at least one TLS-enabled file transfer protocol when configuring protocols.

Configuring CGate objects for TLS

CGate (Connection Gate) objects are identification filters used internally in server mode at the protocol connection level. CGates determine whether a client has the right to connect to Gateway and exchange files with it or not. CGates can also determine if transport security must be implemented. Set the following parameters in the CGate Security tab.

Field	Possible values	Description
TLS profile name tls_sprof (-tlss)	TLS Profile name. This field can contain wildcards ('*' and '?').	TLS Security Profile name used for the incoming call Handshake negotiation.
TLS profile user parameter tls_sprof_user_param (-tlssup)	User parameter. This field can contain wildcards ('*' and '?').	User parameter that you can complete in the Security Profile definition.
TLS Client Authentication tls_client_auth (-tlscauth)	Y | (N)	Indicates whether authentication is required for the client trying to connect.

Configuring Site objects for TLS use

To use TLS, you must configure the Remote Site to use a TLS network security type. When you enable the TLS network security option for a Remote Site, you must set two parameters:

- An outgoing Security Profile that is used to secure connections towards that Site
- An incoming Security Profile that is used (if present) to check security of connections from that Site

The three TLS-specific parameters to set in the Remote Site Net security tab are:

- **Network security option** (network_security_option or -nsopt): set to TLS
- **Security Profile for outgoing connection** (tls_sprof_out): secures the outgoing connection
- **Security Profile for incoming connection** (tls_sprof_in): Incoming connections are first secured by selecting a Security Profile from the best Network Profile that matches the incoming connection parameters. Once the connected Site is known (at file transfer protocol level), a comparison is made between the session security parameters already used (from the network), and those of the session security parameters declared as the incoming Security Profile of the Site. If they do not match, the transfer is aborted and the connection is closed.

When a secured transfer is made, four pieces of information are added to the Transfer Request contained in the Mailbox:

- Security Profile used (sprof)
- Cipher suite used to encrypt connection
- A client authentication indicator (**Yes** or **No**)
- A server authentication indicator (**Yes** or **No**)

These parameters correspond to the last negotiated security parameters and are updated when necessary.

TLS usage example

To set up the TLS secured loop transfers:

1. Enable the TLS option when configuring protocols.
2. Start Gateway and create basic loop configuration.
3. Import the certificates:

```
secadm import_cert  -certificate_name "SOPRACA"
  -certificate_file "SopraCA.pem"
  –certificate_file_type BASE64
secadm import_cert  -certificate_name "USER1"
  -certificate_file "user1.p12"
  –certificate_file_type PKCS12
secadm import_cert  -certificate_name "USER2"
  -certificate_file "user2.p12"
  –certificate_file_type PKCS12
```

   These commands import three certificates:

   - **"SOPRACA" contained in the file "SopraCA.pem"**: The file, in this example,

contains the BASE64 encoded certificate of the certification authority. This certificate has been used to sign the two others.

- **"USER1" contained in the file "user1.p12"**: The file contains a PKCS12 structure that holds both the certificate and its asymmetrical key pair. Keys are encrypted using the password phrase "user1". Prompt for password occurs while importing.
- **"USER2" contained in the file "user2.p12"**: Keys are encrypted using the password phrase "user2". Prompt for password occurs while importing.

4. Import Security Profiles:

    - **secadm import_sprof -sprof_file "server.sprof"**
    - **secadm import_sprof -sprof_file "client.sprof"**

Example of the command secdsp display_sprof -spn AS2_TLS_OUT in the server.sprof file.

sp_name='AS2_TLS_OUT'
sp_type='SERVER'
sp_tls_versions = 'TLSv11' 'TLSv12'
sp_cache_enabled='N'
sp_exit_scheduling='N'
sp_renegotiate_enabled='N'
sp_close_notify_disabled='N'
sp_client_auth_policy='TLS_AUT_OPTIONAL'
sp_path_depth='7'
sp_user_param=''
sp_cipher_suites = 'RSA_WITH_3DES_EDE_CBC_SHA'
'RSA_WITH_AES_128_CBC_SHA'
sp_ft_sec_option='implicit'
sp_ft_session_policy='optional'
sp_ft_data_policy='optional'
sp_protocol_header_policy='unused'
sp_trust_hosted_certificates='Y'
sp_remote_certificate_import='N'
sp_xpp_entity_name=''
sp_xpp_entity_password=''
sp_xpp_partner_entity_name=''
[UserLocalCertificate]
sp_certificate_name='GW_REF'
[AcceptedAuthorities]
[RemoteUserCertificate]
[RemoteCACertificate]
[RemoteRootCertificate]

Example of the command secdsp display_sprof -spn AS2_TLS_OUT in the client.sprof file.

```
sp_name='AS2_TLS_OUT'
sp_type='CLIENT'
sp_tls_versions = 'TLSv11' 'TLSv12'
sp_cache_enabled='N'
sp_exit_scheduling='N'
sp_renegotiate_enabled='N'
sp_close_notify_disabled='N'
sp_client_auth_policy='TLS_AUT_OPTIONAL'
sp_path_depth='7'
sp_user_param=''
sp_cipher_suites = 'RSA_WITH_3DES_EDE_CBC_SHA'
'RSA_WITH_AES_128_CBC_SHA'
sp_ft_sec_option='implicit'
sp_ft_session_policy='optional'
sp_ft_data_policy='optional'
sp_protocol_header_policy='unused'
sp_trust_hosted_certificates='Y'
sp_remote_certificate_import='N'
sp_xpp_entity_name=''
sp_xpp_entity_password=''
sp_xpp_partner_entity_name=''
[ UserLocalCertificate ]
sp_certificate_name='GW_REF'
[ AcceptedAuthorities ]
[ RemoteUserCertificate ]
[ RemoteCACertificate ]
[ RemoteRootCertificate ]
```

The [AcceptedAuthorities] section contains all the certificates the server uses to form the list of accepted authority DNs. This list is first sent to the client.

The [UserLocalCertificate] section contains the list of certificates that the client or the server can send.

The previous commands result in the creation of two Security Profiles:

- "SRV_SPROF" is used to secure incoming network connections. It sends the DN of "SOPRACA" as the accepted authority and "USER1" as the certificate.
- "CLT_SPROF" is used to secure outgoing connections. It sends "USER2" as the certificate (a check is made to ensure "USER2" is signed by "SOPRACA").

5. Create Network Profiles (for the incoming connection).

```
secadm create_netprof  –nprof_name SEC_NETPROF
-origin_address "*/*"  –destination_address "*/2205"
```

```
–tls_option Y  –sprof_name "SRV_SPROF"

secadm create_netprof  –nprof_name DEFAULT_NETPROF

-origin_address "*/*"  –destination_address "*/*"  –tls_option N
```

These commands create two Network Profiles:

- "SEC_NETPROF" that enables TLS with the Security Profile "SRV_SPROF" for any incoming TCP/IP connection on port 2205
- "DEFAULT_NETPROF" that accepts every connection as non-secured (it matches any distant address with any distant SAP and any local address with any SAP)

6. Configure every Remote Site that points to your Gateway address with an SAP of 2205 with the following options:
   - network_security_option TLS
   - tls_sprof_in "SRV_SPROF"
   - tls_sprof_out "CLT_SPROF"
7. Process a file transfer toward one of the Sites configured in the previous step.

Related topics

TLS cipher suites

Managing TLS

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

TLS cipher suites

Axway Gateway: Managing Security

Gateway supports the cipher suites listed in this section. The availability of the ciphers is conditioned by various factors, as shown in the table.

Cipher suite	SSL V3	TLS 1.0	TLS 1.1	TLS 1.2	Is strong Cipher	Usable with Secure Relay (TLS termination)	Usable on Gateway direct connections	Requires unlimited crypto-graphy	Available in FIPS mode	Requires certificate
ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	✗	✗	✗	✓	✓	✓	✓		✓	✓
ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	✗	✗	✗	✓	✓	✓	✓		✗	✓
ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	✗	✗	✗	✓	✓	✓	✓		✗	✓
ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	✗	✗	✗	✓	✓	✓	✓		✗	✓

Cipher suite	SSL V3	TLS 1.0	TLS 1.1	TLS 1.2	Is strong Cipher	Usable with Secure Relay (TLS termination)	Usable on Gateway direct connections	Requires unlimited crypto-graphy	Available in FIPS mode	Requires certificate
ECDHE_ECDSA_WITH_AES_256_CBC_SHA	✗	✓	✓	✓	✓	✓	✓		✗	✓
ECDHE_ECDSA_WITH_AES_128_CBC_SHA	✗	✓	✓	✓	✓	✓	✓		✗	✓
ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	✗	✓	✓	✓	✓	✓	✓		✗	✓
ECDHE_ECDSA_WITH_RC4_128_SHA	✗	✓	✓	✓	✗	✓	✓		✗	✓
ECDHE_RSA_WITH_AES_256_GCM_SHA384	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓
ECDHE_RSA_WITH_AES_128_GCM_SHA256	✗	✗	✗	✓	✓	✓	✓		✓	✓
DHE_RSA_WITH_AES_256_GCM_SHA384	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓
DHE_RSA_WITH_AES_128_GCM_SHA256	✗	✗	✗	✓	✓	✓	✓		✓	✓
ECDHE_RSA_WITH_AES_256_CBC_SHA384	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓
ECDHE_RSA_WITH_AES_128_CBC_SHA256	✗	✗	✗	✓	✓	✓	✓		✓	✓
DHE_RSA_WITH_AES_256_CBC_SHA256	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓
DHE_RSA_WITH_AES_128_CBC_SHA256	✗	✗	✗	✓	✓	✓	✓		✓	✓
RSA_WITH_AES_256_GCM_SHA384	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓
RSA_WITH_AES_128_GCM_SHA256	✗	✗	✗	✓	✓	✓	✓		✓	✓
RSA_WITH_AES_256_CBC_SHA256	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓
RSA_WITH_AES_128_CBC_SHA256	✗	✗	✗	✓	✓	✓	✓		✓	✓
ECDHE_RSA_WITH_AES_256_CBC_SHA	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓
ECDHE_RSA_WITH_AES_128_CBC_SHA	✗	✓	✓	✓	✓	✓	✓		✓	✓
DHE_RSA_WITH_AES_256_CBC_SHA	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DHE_RSA_WITH_AES_128_CBC_SHA	✓	✓	✓	✓	✓	✓	✓		✓	✓
RSA_WITH_AES_256_CBC_SHA	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
RSA_WITH_AES_128_CBC_SHA	✓	✓	✓	✓	✓	✓	✓		✓	✓
ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	✗	✓	✓	✓	✗	✓	✓		✓	✓
DHE_RSA_WITH_3DES_EDE_CBC_SHA	✓	✓	✓	✓	✗	✓	✓		✓	✓
ECDHE_RSA_WITH_RC4_128_SHA	✗	✓	✓	✓	✗	D	✓		✗	✓
ECDHE_RSA_WITH_NULL_SHA	✗	✓	✓	✓	✗	D	✓		✓	✓
DHE_RSA_WITH_DES_CBC_SHA	✓	✓	✓	✓	✗	D	✓		✗	✓
RSA_WITH_3DES_EDE_CBC_SHA	✓	✓	✓	✓	✗	✓	✓		✓	✓

Cipher suite	SSL V3	TLS 1.0	TLS 1.1	TLS 1.2	Is strong Cipher	Usable with Secure Relay (TLS termination)	Usable on Gateway direct connections	Requires unlimited crypto-graphy	Available in FIPS mode	Requires certificate
RSA_WITH_RC4_128_SHA	✓	✓	✓	✓	✗	D	✓		✗	✓
RSA_WITH_DES_CBC_SHA	✓	✓	✓	✓	✗	D	✓		✗	✓
RSA_WITH_RC4_128_MD5	✓	✓	✓	✓	✗	D	✓		✗	✓
RSA_WITH_NULL_SHA256	✓	✓	✓	✓	✗	D	✓		✓	✓
RSA_WITH_NULL_SHA	✓	✓	✓	✓	✗	D	✓		✗	✓
RSA_WITH_NULL_MD5	✓	✓	✓	✓	✗	D	✓		✗	✓
RSA_ANON_WITH_NULL_MD5		✓	✓	✓	✗		✓		✗	
RSA_ANON_WITH_NULL_SHA		✓	✓	✓	✗		✓		✗	
RSA_ANON_WITH_RC4_128_MD5		✓	✓	✓	✗		✓		✗	
RSA_ANON_WITH_RC4_128_SHA		✓	✓	✓	✗		✓		✗	
RSA_ANON_WITH_DES_CBC_SHA		✓	✓	✓	✗		✓		✗	
RSA_ANON_WITH_3DES_EDE_CBC_SHA		✓	✓	✓	✗		✓		✓	
RSA_ANON_WITH_AES_128_CBC_SHA		✓	✓	✓	✗		✓		✓	
RSA_ANON_WITH_AES_256_CBC_SHA		✓	✓	✓	✗	✓	✓	✓	✓	
NULL_WITH_NULL_NULL	✓	✓	✓	✓	✗	✗	✓		✗	

Comments about the table above:

D : Disabled by default in Secure Relay

Strong ciphers / weak ciphers

By default, on a new installation, only strong ciphers can be selected in TLS profiles. For compatibility reasons, a parameter was added: [tls]enable_weak_ciphers, which enables all supported ciphers.

When enable_weak_cipher is set to *yes* and a weak cipher is selected:

• a warning message is shown on selection

• it will no longer be available in transfers if the enable_weak_cipher option is switched to *no* (a warning message will be shown).

Ciphers using the MD5 hash algorithm: the parameter: `[monitor]enable_md5_hash_algorithm` (not recommended) enables ciphers using the MD5 hash algorithm. When `enable_md5_hash_algorithm` is set to *no* and a cipher suite using the MD5 hash algorithm is selected:

• The cipher is no longer available for the TLS handshake and in the TLS security profile configuration

Strong cryptography / weak cryptography: selecting strong cipher suites does not necessarily means all cryptographic parameters are strong. For example, the TLS_DHE ciphers might still accept weak 1024 bit DH key generation parameters. To control this behavior, a parameter was added: [tls]enable_weak_ciphers, which –

when set to “no” – would only allow the recommended 2048 bit or higher DH key generation parameters (please see the DHE key exchange in TLS section for more details) for DHE key exchange.

Note. For compatibility reasons, the default value for the [tls]enable_weak_ciphers is “yes”. It is strongly recommended to set this parameter to “no”.

Strong cryptography / weak cryptography

Selecting strong cipher suites does not necessarily imply that all cryptographic parameters are strong. For example, the TLS_DHE ciphers might still accept weak 1024-bit DH key generation parameters. To control this behavior, Gateway offers a parameter: `[tls]enable_weak_ciphers`, which – when set to **`no`** – only allows the recommended 2048-bit or higher DH key generation parameters.

Note For compatibility reasons, the parameter’s default value is `yes`. We **strongly recommend** to set this parameter to **`no`**.

For more information, see Managing TLS > Avoiding weak cryptography

Related topics

Managing TLS Security Profiles

Managing TLS

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Managing TLS Security Profiles

Axway Gateway: Managing Security

A TLS Security Profile contains all the security configuration elements used to establish a TLS connection.

To import a TLS Profile, use the secadm import_sprof online command.

Defining TLS Security Profiles

Deleting a TLS Security Profile

Certificate templates

Defining TLS Security Profiles

To add a new TLS Profile to the profile database:

1. In the left pane of the GUI main window, click ⊞ to expand the nodes:
   Security Management > Transfer Security Management > Security Profile
2. Right-click the **TLS Profile** sub-node, and then select **New** from the context menu.
   Gateway displays the *New TLS Profile* window.

3. Complete the fields of the following tabs as described below:
    - *General*
    - *Details*
    - *PassPort PS*

TLS Profile: General tab

Name	Description
Name	Enter a unique name for the TLS Profile. Maximum: 31 alphanumeric characters.
Profile type	Select the type of connection to use with the Security Profile. • CLIENT • SERVER
Client authentication	Indicates whether client authentication is mandatory, forbidden or optional. Select one of the following options: • TLS_AUT_ANONYMOUS • TLS_AUT_OPTIONAL • TLS_AUT_MANDATORY
Accepted SSL versions	List of accepted protocol versions. Select one or more of the following SSL/TLS versions: • TLSV1.1 • TLSV1.2 • TLSV1 • SSLV3 • SSL2HELLO **Note**: when creating a new TLS profile, TLSV1.1 and TLSV1.2 are selected by default. **Note**: SSL2HELLO has been added for backwards compatibility with older Java applications.
Accepted cipher suites	List of cipher suites that you can use to secure the connection. The list must be in *decreasing order* of preference. Note that Gateway only displays a cipher suite if it is supported by all the selected SSL/TLS versions. You can select up to 32 cipher suites. For an overview of accepted values for the list elements, see TLS cipher suites. **Note**: The default cipher suite selection order, in the case Gateway is the TLS server, is to choose the first cipher in common, using the client preferred

Name	Description
	(proposed) order. To change this behavior and select ciphers in server preferred order, please use the global static setting [tls]tls_cs_server_order: • 0 (default value): client preferred order (the cipher order in the ClientHello message) • 1: server preferred order, the cipher order in the TLS server profile.
Accepted authorities (server only)	List of local certificate names. Server Use Only: certificate local names of the accepted authorities (CA) whose DER-encoded subject names are sent to the client. The client must then send a certification path that matches one of the authorities provided. Refer to Accepted authorities list.
Certificate templates	Add certificate templates to use to customize verification of received certification paths. You can add USER, CA, or ROOT certificate templates.
Trust hosted certificates only	The term **hosted** applies to a certificate (or public key) hosted in the Gateway database. The origin (owner of the associated private key) is known and is authorized to connect via TLS. Check this option to accept authentication of a partner if any certificate (or public key) belonging to the partner is already present in the Gateway database in the state ENABLED.
Automatic import of partner certificate chains	Check this option to allow the automatic import of unknown certificates presented by partners that are authenticating themselves for the first time (or that have changed certificates). Before any certificate import operation, Gateway checks (with the hash) that the certificate does not already exist in the database.
User certificates	Server and client use: Set of certificate local names, first step of the certification path to send to the remote partner. You can either select a single user certificate or, alternatively, multiple certificates (up to ten). Refer to Defining User Certificate list. Gaetway has a new way of building certificate paths, both in server and client mode. It is an implementation of the Internet X.509 Public Key Infrastructure: Certification Path Building specification, also known as RFC4158. This mode of operation can be configured on individual security profiles using the attribute `sprof_path_build_mode`, (short version `sppbm`), having 2 possible values: • *legacy* • *rfc4158*

TLS Profile: Details tab

Name	Description
Cache enabled	Check this option to enable reuse of the session security parameters.
Renegotiate enabled	Check this option to enable cipher key renegotiation. Select one of the following options: disabled, unrestricted, secure, legacy. For details, see the section Secure renegotiation in TLS. **Note:** This feature may not be compatible with certain systems.
Refuse legacy partners	Close a connection from a TLS un-upgraded partner from the start (from the initial TLS handshake)
Exit scheduling	Check this option to enable the use of an external manager. If you check this option, subsequent parameters are ignored.
Close notify disabled (FTP and HTTP)	Check this option to disable the use of close notify messages. Enables inter-operability with products that do not support this feature.
Certificate path	Enter the maximum number of certificates accepted in a certification chain. Any certification chain that contains more elements is rejected.
User param	Enter a user parameter. This parameter is provided to SECS and Xfer exits. Maximum: 31 characters.
Protocol packet header policy in TLS	Usage of protocol packet headers for TLS. Possible values: • **Unused** - Gateway will not use protocol packet headers • **Used** - always use protocol packet headers for TLS • **Detect** - auto detect mode (depending on how the partner is sending) When using non-stream protocols over TLS (PeSIT, Odette FTP), it is recommended to set the value to **Used** (or at least Detect). This is because, for non-stream protocols, when transported over a TLS encrypted connection, Gateway may add (when transmitting) or expect (when receiving) some protocol packet headers. These headers are meant to delimit the different protocol units (ex FPDUs) inside a TLS stream. As a TLS stream works in 16K records, it may happen that a TLS record is not exactly fitting a protocol packet. Moreover, if TLS is not done inside Gateway (ex: TLS termination in XSR), the TLS record may be intentionally fragmented (also fragmenting the FPDU inside). For the Odette protocol, the RFC clearly states that this mechanism should be used over TLS, for PeSIT it is not clear from specification, so it depends on the implementation if the partner will support this mechanism or not.

Name	Description
Protocol packet header policy in TLS	Usage of protocol packet headers for TLS. Possible values: • **Unused** - do *NOT* use protocol packet headers • **Used** - *always* use protocol packet headers for TLS • **Detect** - auto detect mode (depending on how the partner is sending) **Note** When using non-stream protocols over TLS, (PeSIT or Odette FTP), it is recommended to set the value to **Used**, or at least to **Detect**. This is because Gateway may add (when transmitting) or expect (when receiving) some protocol packet headers when using non-stream protocols Odette and PeSIT transported over a TLS encrypted connection. These headers delimit the different protocol units (ex FPDUs) inside a TLS stream. As TLS streams use 16K records, it may happen that a TLS record is not exactly fitting a protocol packet. Moreover, if TLS is not done inside Gateway (for example, a TLS termination in XSR), the TLS record may be intentionally fragmented (also fragmenting the FPDU inside). This is why, when using non stream protocols over TLS (PeSIT, Odette FTP), it is recommended to set the value to "Used" (or at least Detect). While for the Odette protocol, the RFC clearly states that this mechanism should be used over TLS, for PeSIT it is not clear from specification, so it depends on the implementation if the partner will support this mechanism or not.
FTP/SMTP/HTTP:	
Connection securing mode	You can configure Gateway to handle incoming connections in *implicit* mode or *explicit* mode. FTP and SMTP security in a Gateway client can be implemented in either *implicit* or *explicit* mode. HTTP security must be set to *implicit* on both the client and server profiles. Select one of the following options: • Implicit: Gateway uses TLS in implicit mode by default. In implicit mode, a secure port is used before the connection takes place and the control session is opened directly in SSL/TLS. Note: This type of 'negotiation' has been deprecated and is no longer recommended by the IETF. • Explicit: In explicit mode, the connection occurs on a non-encrypted port, and the decision to connect in SSL/TLS is made by negotiation once the session is established.
Global session policy	*This field is only available if you set the **Connection securing mode** to Explicit.* Use this option to define whether control session security is mandatory or optional.
FTP data session	*This field is only available if you set the **Connection securing mode** to Explicit.*

Name	Description
policy	Use this option to define whether data session security is mandatory, optional, or disabled.

TLS Profile: PassPort PS tab

Enter values in the fields of the **PassPort PS** tab if you are using PassPort security functions for connections with partners defined in PassPort PS.

Name	Description
Local entity	
Name	Enter the local entity name for access to the PassPort PS server.
Set Password	Opens a dialog box to enter the password corresponding to the entity name.
Confirmation	Confirm the password.
Object Type	Passport PS object type of the entity. Can be one of the following: • *Passport PS entity* • *Passport PS certificate* For profiles used with the XSR termination, the only valid selection is *Passport PS entity*.
Partner entity	
Name	Identity of the PassPort partner as provided by the PassPort PS server.
Object Type	Passport PS object type of the entity. Can be one of the following: • *Passport PS entity* • *Passport PS certificate* For profiles used with the XSR termination, the only valid selection is *Passport PS entity*.

Deleting a TLS Security Profile

Equivalent online command: secadm delete_sprof.

To delete a TLS Profile from the profile database:

1. In the left pane of the GUI main window, click ⊞ to expand the nodes:

Security Management > Transfer Security Management > Security Profile
2. Click to select the **TLS Profile** sub-node.
   Gateway displays all existing TLS Profiles in the right pane.
3. In the right pane, right-click the TLS Profile that you want to delete. Select **Delete...** from the context menu.
   Gateway displays a confirmation window.
4. Click **Yes**.
   Gateway deletes the TLS Profile.

Certificate templates

Certificate Templates enable you to customize the verification of a received certification path.

For each certificate path, the Certificate Template:

- *Determines* the certificate type (User, Intermediary CA or Root CA).
- *Parses* the current Security Profile object.
- *Checks* the corresponding *Template List* (respectively RemoteUserCertificate, RemoteCaCertificate, and RemoteRootCertificate). If the list is not empty (which means that the certificate is accepted by default), the certificate must match one of the template records.

The matching criteria proposed for a template record are:

- **Certificate name:** If this field is completed, the certificate received must be present in the local database and is referenced here by its name.
- **Pattern matching:** If a certificate name is not present, the certificate issuer and subject distinguished names must match the following certificate template fields:
  - *subject_name*: pattern that contains the substitution characters ***** and **?**
  - *issuer_name*: pattern that contains the substitution characters ***** and **?**
  - *alternative*

Adding a certificate template

To add a certificate template:

1. In the left pane of the GUI main window, click ⊞ to expand the nodes:
   Security Management > Transfer Security Management > Security Profile
2. Click to select the **TLS Profile** sub-node.
   Gateway displays all existing TLS Profiles in the right pane.
3. In the right pane, right-click an existing Profile. Select **Modify...** from the context menu.
   Gateway opens the *TLS Profile* window.
4. In the **General** tab of the *TLS Profile* window, click the **Certificate templates** button.
5. In the **Certificate Templates** tab, select from:
   - **USER**
   - **CA**
   - **ROOT**
6. Click **Add**.
   Gateway displays the *Template* window.
7. Select from either:

- **Match the local certificate whose name is:** enter the certificate name.
- **Match the DN criteria:** complete the listed fields.

8. Click **OK** to confirm and close the *Template* window.

You can also remove a certificate in the **Certificate Templates** tab. Select the certificate and click **Remove**.

Certification path building mode in TLS

Gaetway has a new way of building certificate paths, both in server and client mode. It is an implementation of the Internet X.509 Public Key Infrastructure: Certification Path Building specification, also known as RFC4158.

This mode of operation can be configured on individual security profiles using the attribute `sprof_path_build_mode`, (short version `sppbm`), having 2 possible values:

- *legacy*
- *rfc4158*

Related topics

TLS cipher suites

Managing TLS

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Managing TLS Security Profiles (command line)

Axway Gateway: Managing Security

secadm create_sprof

secadm update_sprof

secadm import_sprof

secadm delete_sprof

secdsp display_sprof

secdsp select_sprof

secdsp status_sprof

secbase export_sprof

TLS Profile file format

Creating a TLS Profile : secadm create_sprof

Syntax	**secadm create_sprof** `{-sprof_name (-spn)}`
Description	Use this command to create a TLS Profile.
Parameters	Mandatory -sprof_name (-spn): Enter the name of the TLS profile to create. **Note**: although specifying the TLS versions is not mandatory, it is highly recommended that you do it. A TLS profile without the accepted TLS versions specified is not usable. **Note**: to get the complete parameters list, execute the command : **secadm create_sprof ?**
	Optional -sprof_ciph_suites (-spcsu) : enter a comma-separated string list of up to 32 cypher suites to secure the connection.
Example	The following command creates a basic TLS SERVER profile supporting TLS 1.1 and TLS 1.2 versions, with the default cipher suites selected: RSA_WITH_3DES_EDE_CBC_SHA and RSA_WITH_AES_128_CBC_SHA: **`secadm create_sprof –spn "sprof1" -sptv "TLSv11, TLSv12"`** **Note**: The name of the TLS Profile must be unique.

Updating a TLS Profile : secadm update_sprof

Syntax	**secadm update_sprof** `{-sprof_name (-spn)}`
Description	Use this command to modify the parameters of a TLS Profile.
Parameters	*Mandatory* -sprof_name (-spn): Enter the name of the file that contains the TLS Profile to import. Note: The name of the TLS Profile cannot be changed. Note: To get the complete parameters list, execute the command: secadm update_sprof ?
	Optional -sprof_ciph_suites (-spcsu) : enter a comma-separated string list of up to 32 cypher suites to secure the connection.
Example	The following command modifies a TLS Profile to extend the list of accepted cipher suites and to set client authentication mandatory: secadm update_sprof –spn "sprof1" -spclia TLS_AUT_MANDATORY -spcsu "RSA_WITH_3DES_EDE_CBC_SHA, RSA_WITH_AES_128_CBC_SHA, RSA_WITH_NULL_MD5, RSA_WITH_RC4_128_MD5"

Importing a TLS Profile : secadm import_sprof

Syntax	**secadm import_sprof** `{-sprof_file (-spf)} {-encryption_key_file (-ekf)} {-import_pwd_cleartext }`
Description	Use this command to import a TLS Profile from a file into the local database.
Parameters	*Mandatory* -sprof_file (-spf): Enter the name of the file that contains the TLS Profile to import. -encryption_key_file (-ekf): Enter the path to the encryption key file to be used to decrypt passwords contained by the object to be imported OR -import_pwd_cleartext: Specifies a legacy import file containing unencrypted passwords. **Notes:** • Only one of the -encryption_key_file or -import_pwd_cleartext options should be used. Using one of the parameter is mandatory. • To get the complete parameters list, execute the command **secadm import_sshprof ?**
Example	The following command imports the TLS Profile contained in the file sprof1.sprof into the database: secadm import_sprof –sprof_file "sprof1.sprof" -ekf <decryption_key.dat> Note: The name of the TLS Profile must be unique.

Deleting a TLS Profile: secadm delete_sprof

Syntax	secadm delete_sprof `{-sprof_name (-spn)}`
Description	Use this command to delete a TLS Profile.
Parameters	*Mandatory* -sprof_name (-spn): Enter the name of the TLS Profile. Maximum: 31 characters. **Note**: to get the complete parameters list, execute the command **`secadm delete_sprof ?`**
Example	The following command deletes the TLS Profile named *sprof_1*: secadm delete_sprof –sprof_name "sprof_1"

Displaying a TLS Profile: secdsp display_sprof

Syntax	**`secdsp display_sprof {-sprof_name (-spn)}`**
Description	Use this command to display a TLS Profile.
Parameters	*Mandatory* -sprof_name (-spn): Enter the name of the TLS Profile. **Note**: to get the complete parameters list, execute the command: **`secadm display_sprof ?`**
Example	The following command displays the TLS Profile named sprof_1: secdsp display_sprof -spn SPROF_INEXP sp_name='SPROF_INEXP' sp_type='SERVER' sp_tls_versions = 'SSLv3' 'TLSv1' sp_cache_enabled='N' sp_exit_scheduling='N' sp_un-upgraded_partner_refused='N' sp_renegotiate_enabled='disabled' sp_close_notify_disabled='Y' sp_client_auth_policy='TLS_AUT_MANDATORY' sp_server_auth_policy='TLS_AUT_MANDATORY' sp_path_depth='15' sp_user_param='' sp_cipher_suites = 'RSA_WITH_3DES_EDE_CBC_SHA' 'RSA_WITH_AES_128_CBC_SHA' sp_ft_sec_option='explicit' sp_ft_session_policy='optional' sp_ft_data_policy='optional' sp_trust_hosted_certificates='N' sp_remote_certificate_import='N' sp_xpp_entity_name='EU3' sp_xpp_entity_password='test' sp_xpp_object_type='Entity' sp_xpp_partner_entity_name='' sp_xpp_partner_object_type='Entity' sp_sprof_path_build_mode='rfc4158'

Listing all TLS Profile names: secdsp select_sprof

Syntax	**`secdsp select_sprof`**
Description	Use this command to list the names of all existing TLS Profiles.

Syntax	**secdsp select_sprof**
Parameters	None
Example	secdsp select_sprof

Listing all TLS Profile abstracts: secdsp status_sprof

Syntax	**secdsp status_sprof**
Description	Use this command to display an abstract of each TLS Profile.
Parameters	None
Example	Enter the command: secdsp status_sprof Result: Name Type Exit Cache Client_auth Server _auth SP Server N N Optional Mandatory

Exporting TLS Security Profile objects: secbase export_sprof

Syntax	**secbase export_sprof** `{-encryption_key_file} [-format] [-user_fmt] [-output] [-append]`
Description	Use this command to export the fields and field values of one or more TLS Security Profiles (SProfs). Gateway either displays the results on the screen or copies the results to an ASCII file. Use SProf attributes as selection criteria, as described below.
Parameters	*Mandatory* -encryption_key_file (-ekf): Enter the path to the encryption key file to be used to decrypt passwords contained by the object to be imported *Optional* -format (-of): Output format. Enter one of the following values: • S = Shell (example: field = *value*). (Default) • C = C-Shell (example: set field = *value*) • U = User format

Syntax	**secbase export_sprof** `{-encryption_key_file} [-format] [-user_fmt] [-output] [-append]`
	-user_fmt (-uf): User format definition. Enter a character string including the keywords %N and %V, where: • %N: name of field • %V: value of field
	-output (-f): Name of the output file. Maximum: 127 alphanumeric characters. If you do not specify this parameter, the standard output file is used.
	-append (-ap): Concatenation at the end of the file (if it exists). This parameter takes no value.
Example	Enter the command: secbase export_sprof -ekf <encryption_key.dat> -n SProf1 -f ExpSProf Gateway creates the ExpSProf export file. From the SProf object named SProf1, it extracts all fields and their corresponding values, and writes them to the ExpSProf file. The encryption key file <encryption_key.dat> will be used to encrypt passwords contained by the SSH Security Profile.

Importing a TLS Profile: secbase import

Syntax	secbase import {-input(-if)} {-encryption_key_file(-ekf)} {-import_pwd_cleartext} {-sprof(-sp)} [-error_free]
Description	Use this command to import an TLS Profile from a text file into the local database.
Parameters	*Mandatory*
	-input (-if): Enter the name of the file that contains the TLS Security Profile to import.
	-encryption_key_file (-ekf): Enter the path to the encryption key file which will be used to decrypt passwords contained by the object to be imported. OR

Syntax	secbase import {-input(-if)} {-encryption_key_file(-ekf)} {-import_pwd_cleartext} {-sprof(-sp)} [-error_free]	
	-import_pwd_cleartext: Specifies a legacy import file which contains unencrypted passwords. **Note** Only one of the -encryption_key_file or -import_pwd_cleartext options should be used. Using one of the parameters is mandatory.	
	-sprof (sp): Enter this parameter without entering a value to import all TLS Security Profile objects contained in the file specified in the input parameter.	
	Optional	
	-error_free: Enter this parameter to create the different objects while ignoring the new parameters and without stopping the process at each error.	
	Note To get the complete list of parameters, execute the command **secbase import ?**	
	Example	To import a TLS Security Profile contained in the file tlsprof_desc_file.txt, enter: secbase import –if <tlsprof_desc_file.txt> -ekf <decryption_key.dat> -sp The Profile is described in the text file provided in the -if parameter, in this case: <tlsprof_desc_file.txt>. All of the Profile parameters are in this file. The decryption key is specified in the -ekf parameter. If you have to import a TLS Security Profile export file containing cleartext passwords, generated with an older version, use the -import_pwd_cleartext parameter instead.

TLS Profile file format

Import TLS Profiles using a file that contains all the fields described in the Security Profile section. All unknown fields are ignored. If a field is absent or if the value is not correct, an error message is displayed and the import is

aborted.

Example of a TLS Profile file

sp_name='SPROF1'
sp_type='SERVER'
sp_tls_versions = 'TLSv12'
sp_cache_enabled='N'
sp_exit_scheduling='N'
sp_un-upgraded_partner_refused='N'
sp_renegotiate_enabled='disabled'
sp_close_notify_disabled='N'
sp_client_auth_policy='TLS_AUT_MANDATORY'
sp_server_auth_policy='TLS_AUT_MANDATORY'
sp_path_depth='7'
sp_user_param=''
sp_cipher_suites = 'ECDHE_RSA_WITH_AES_256_GCM_SHA384' 'ECDHE_RSA_WITH_AES_128_G
CM_SHA256' 'DHE_RSA_WITH_AES_256_GCM_SHA384' 'DHE_RSA_WITH_AES_128_GCM_SHA256' '
RSA_WITH_AES_256_GCM_SHA384' 'RSA_WITH_AES_128_GCM_SHA256'
sp_ft_sec_option='implicit'
sp_ft_session_policy='optional'
sp_ft_data_policy='optional'
sp_protocol_header_policy='unused'
sp_trust_hosted_certificates='Y'
sp_remote_certificate_import='Y'
sp_xpp_entity_name='PassPort_Entity'
sp_xpp_entity_password='vUvLAyN0qg+XMM//1jgotgIGG9bNs2yX7MH/shiKM6ZuAa72jDwtVAcg
vzi7uTuVAAIBAgIKCwUgCg=='
sp_xpp_object_type='Entity'
sp_xpp_partner_entity_name=''
sp_xpp_partner_object_type='Entity'

[UserLocalCertificate]
sp_certificate_name='AxwayHTTP_Server'

[AcceptedAuthorities]
[RemoteUserCertificate]
[RemoteCACertificate]
[RemoteRootCertificate]

Related topics

Managing TLS Security Profiles

Managing TLS

TLS cipher suites

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

FIPS

About FIPS

Application-level protocols

The application-level protocols subject to FIPS compliance are SFTP, FTPS, HTTPS, AS1, AS2, AS3, OFTP, POP3, SMTP, RosettaNet, PeSIT.

The restrictions apply both in the configuration of these protocols and in the data transfer execution.

Transport-level protocols

The transport-level protocols subject to FIPS compliance are: TLS, SSH.

The restrictions apply both in the configuration of these protocols and in the data transfer execution.

Keys and certificates management

In FIPS mode, it is forbidden to import in Gateway keys or certificates which do not comply with FIPS 140-2 standard.

The non-compliant keys and certificates imported prior to the activation of the FIPS mode are marked as FIPS restricted and cannot be referenced by other security-related objects (security profiles, Remote Sites, etc).

If the security-related objects created prior to the activation of the FIPS mode and referencing non-compliant keys or certificates are being used in data transfers, those transfers will fail.

Interoperability with other Axway products

When FIPS mode is enabled, the secure communication between Gateway and PassPort (AM, PM, PS) is FIPS compliant.

Note: when FIPS mode is enabled, it is recommended to use PassPort PS as a PKI.

For more information...

For more information about Gateway and FIPS, refer to Gateway User Guide.

For more information, read SECURITY REQUIREMENTS FOR CRYPTOGRAPHIC MODULES, a publication of Information Technology Laboratory - National Institute of Standards and Technology.

Related topics:

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Using FIPS

Installing FIPS support

FIPS mode cannot be toggled from the Gateway configuration panel. You must provide a FIPS-enabled license key which will enable FIPS mode on Gateway.

On an existing Gateway instance, after upgrade to a newer, FIPS compliant release, you must replace the existing license key with a FIPS-enabled one, by running the Configure tool you can find in the installation root.

Configuring security-related objects to be FIPS compliant

Keys

In FIPS mode, the following restrictions apply when importing new keys (public or private) in Gateway:

- the keys of the following types are not accepted by Gateway (the “FIPS restricted” user message is displayed and logged):
    - ED25519 keys
    - DSS and RSA keys with lengths up to 1024
      Note: Gateway supports DSS keys up to 1024 bits only. As a result, DSS keys are no longer usable in Gateway in FIPS mode.
- In case of legacy keys (keys imported before FIPS-enablement), if the key does not meet the FIPS 140-2 standard, it is marked in Navigator GUI with a **FIPS restricted** label and icon.

Certificates

In FIPS mode, the following restrictions apply when importing new certificates (public or private) in Gateway:

- In case of legacy certificates (imported before FIPS-enablement), if the certificate does not meet the FIPS 140-2 standard, it is marked in Navigator GUI with a “FIPS restricted” label and icon.

SSH Profiles

In FIPS mode, the following restrictions apply when creating a new SSH Profile in Navigator GUI:

- the following MAC algorithms are not available: HMAC_MD5, HMAC_MD5_96, HMAC_SHA_96

In command line, if you will not be able to

- create an SSH Profile referring one of the restricted security functions listed above
- modify an SSH Profile to refer one of the restricted security functions listed above

In case of legacy SSH Profiles (profiles created before FIPS-enablement), if they contain one or more FIPS restricted security functions from the ones listed above, they are not removed from the Profile, they are greyed out in Navigator GUI, but they are not used in data transfer.

If an SSH Profile contains only FIPS-restricted security functions, or if it refers a FIPS restricted public or private key, the execution of any data transfer involving that Profile object will fail.

To remove the non-FIPS settings from an SSH Profile, modify it using the online commands. It is not possible to modify the non-FIPS settings of a legacy SSH Profile from Navigator GUI.

TLS Profiles

In FIPS mode, the following restrictions apply when creating a new TLS Profile in Navigator GUI:

- some cipher suites are not available in the **Accepted cipher suites** list. Refer to the *Available in FIPS mode* column of *TLS cipher suites*

In command line, you will not be able to

- create a TLS Profile referring one of the restricted security functions listed above
- modify a TLS Profile to refer one of the restricted security functions listed above

In case of legacy TLS Profiles (profiles created before FIPS-enablement), if they contain one or more FIPS restricted security functions from the ones listed above, they are not removed from the Profile, they are greyed out in Navigator GUI, but they are not used in data transfer.

If a TLS Profile contains only FIPS-restricted security functions, the execution of any data transfer involving that Profile object will fail.

To remove the non-FIPS settings from a TLS Profile, modify it using the online commands. It is not possible to modify the non-FIPS settings of a legacy SSH Profile from Navigator GUI.

CGates

In FIPS mode, the following restrictions apply when creating a new CGate object in Navigator GUI:

- on the **Security** tab, in the **SSH remote authentication** section, the Public key alias list does not contain the FIPS restricted keys
- on the **Security** tab, in the **TLS and SSH remote certificate filter** section the Subject certificate alias list does not contain the FIPS restricted certificates

In case of legacy CGate objects that refer to FIPS restricted keys or certificates, these are not automatically removed from the CGate object, but any data transfer involving that CGate object will fail.

Remote Sites

All Remote Site objects having a “Net security” tab to configure are subject to the following restrictions when creating or modifying the Remote Site:

• if the TLS option is selected, the Subject certificate alias list does not contain the FIPS restricted certificates.

In case of legacy Remote Sites referring FIPS-restricted certificates, these are not automatically removed from the Remote Site object, but any data transfer involving that Remote Site object will fail.

Additional restrictions apply to Remote Site objects for specific protocols when creating or modifying the Remote Site, as follows:

SFTP

- in the **Parameters to connect to Remote Site** section:
  - the Private key alias list does not contain the FIPS-restricted keys
  - the Certificate alias list does not contain the FIPS-restricted certificates
- in the **Remote Site authentication** section:
  - the Public key alias list does not contain the FIPS-restricted keys
  - the Subject certificate alias list does not contain the FIPS-restricted certificates

If a legacy SFTP Remote Site refers FIPS-restricted keys or certificates, these are not automatically removed from the Remote Site object, but any data transfer involving that Remote Site object will fail.

POP3, AS1_POP3, RosettaNet POP3

- on the POP3 tab, the “Apop” and “Cram” authentication methods are not available.

If a legacy Remote Site refers one of the “Apop” or “Cram” authentication methods, this setting is not automatically removed from the Remote Site object, but any data transfer involving that Remote Site object will fail.

Trading Partners

All Trading Partners (except for the SYSTEM_MESSAGES type) are subject to the following restrictions in the FIPS mode:

- the Certificate Name list(s) do not contain the FIPS-restricted certificates

If a legacy Trading Partner object refers a FIPS-restricted certificate(s), these are not automatically removed from the Trading Partner object, but any data transfer involving that Trading Partner object will fail.

Related topics
FIPS standard
TLS cipher suites

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

DMZs and Firewalls

Axway Gateway: Managing Security

About DMZs and Firewalls

This section contains descriptions of the Gateway processes and functions that you can use to deploy the product in security configurations.

When you link your enterprise network to the Internet, you can enhance local isolation and security by deploying Gateway in a DMZ configuration.

Axway offers two solutions for deploying Gateway in DMZ architectures:

- *Axway Secure Relay*
- *Paired Axway Gateways*

Related topics

Overview: DMZ deployment

About Secure Relay

Working with Secure Relay

Paired Axway Gateways in a DMZ

TCP port management with Firewalls

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

About Secure Relay

Axway Gateway: Managing Security

About Secure Relay

Global architecture

Secure Relay components

About Secure Relay (formerly XSR)

To provide secure links between Gateway and file-exchange partners over the Internet, you can deploy Gateway in a DMZ protected architecture.

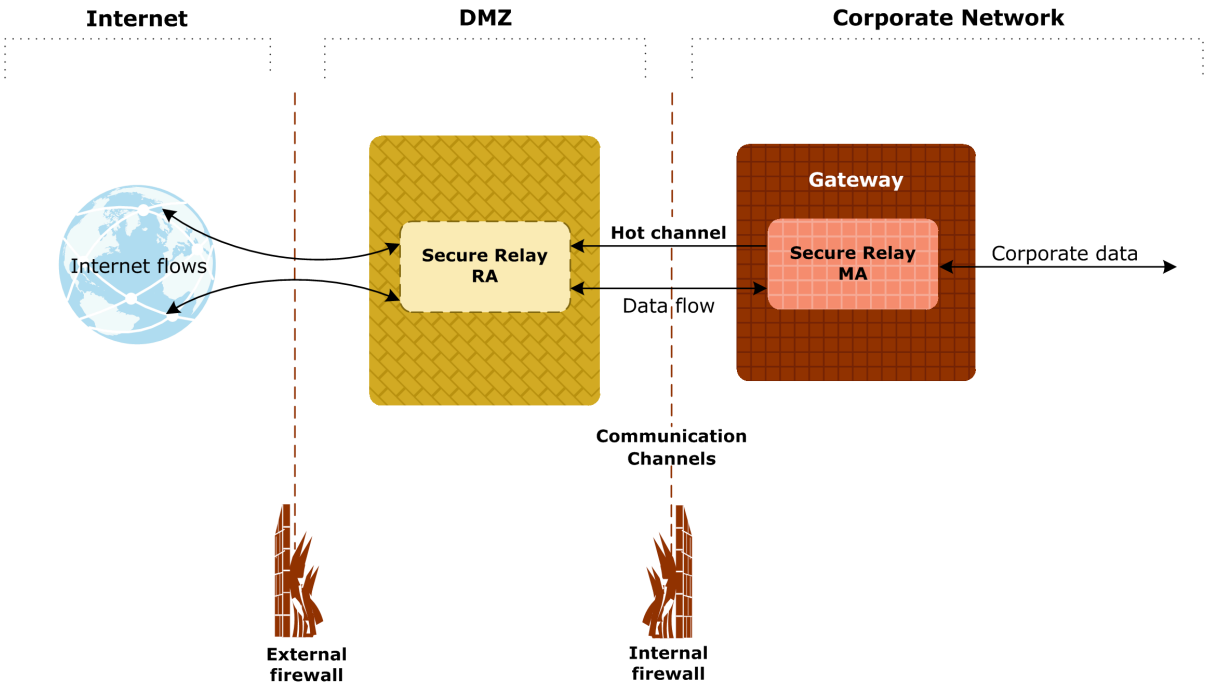
Gateway includes the Secure Relay components. These components enable you to securely exchange transfers with remote partners via a DMZ.

Secure Relay is a secure, multi-protocol, front-end component that prevents external Internet partners from directly accessing Gateway for file exchanges.

For a general description of the role of Secure Relay in Gateway, refer to Overview: DMZ deployment.

Global architecture

The following illustration shows the general principles of Secure Relay deployment.



Secure Relay components

Secure Relay is composed of two main components:

- Secure Relay Master Agent

- Secure Relay Router Agent

Secure Relay Master Agent

The Secure Relay Master Agent (Secure Relay MA) is a Secure Relay component residing with, and fully integrated in, Gateway. All of the processes that support the functioning of Secure Relay MA are automatically included in Gateway on installation.

During installation, the installation program installs Secure Relay related files in the directory <Gateway installation directory>/run_time/xsr.

After installation, you need to configure and activate the Secure Relay Master Agent.

Secure Relay Router Agent

The Secure Relay Router Agent (Secure Relay RA) is a stand-alone component of Secure Relay. You install Secure Relay RA on a machine located in the DMZ, independent of Gateway.

Communication interfaces

Communication between Gateway and the Secure Relay components relies on the following two interfaces:

- Hot channel interface
- Communication channel interface

Hot channel interface

The Hot channel (or Admin channel) interface provides secure connections for dynamic exchanges of orders/responses. Gateway uses these exchanges to drive the physical connections as follows:

- For incoming calls, Secure Relay accepts and streams connections to the protected area, or alternatively, denies the connections
- For outgoing calls, Secure Relay opens connections from the DMZ to external partners

The Hot channel is a single secure TCP/IP connection between a Secure Relay MA and a Secure Relay RA. The Secure Relay MA always opens this connection towards the Secure Relay RA.

The following types of data travel back and forth over the Hot channel:

- Secure Relay configuration settings that Gateway sends on startup (including the static ports to which the Secure Relay RAs must listen for incoming connections)
- Request/Response exchanges that enable Gateway to dynamically:
    - Manage data connections for FTP (either in passive or active mode)
    - Open new connections for outgoing calls
    - Analyze incoming connection indications to indicate that a partner is opening a new TCP/IP connection and that a new logical data streaming channel (through one of the Communication channels) must be started
    - Allow or deny incoming connections based on SSL parameters or protocol authentication

Hot channel security

SSL

The Hot channel TCP/IP connection is secured using TLS transport security.

By using X.509 certificates (strong authentication) on both Gateway and Secure Relay RA sides, TLS ensures that:

- Only Gateway (via Secure Relay MA) and the Secure Relay RA can communicate with each other, avoiding communication hijacking.
- No one can interfere with the orders sent by Secure Relay MA to the Secure Relay RA.
- No one can even sniff the network, or understand what operation is being processed and try to take advantage of it

Firewall-friendly features

You define TCP/IP connections via address/port couples on each communicating site.

You can fully configure the address/port couple for the Hot channel either in the Gateway configuration menu or the Secure Relay configuration file. This facility guarantees that the file-exchange solution (comprising both Gateway and Secure Relay in a distributed system) can be deployed in environments with the most restrictive Firewall management rule policy.

Communication channel interface

Communication channels are a predefined set of TCP/IP physical connections opened between Secure Relay MA and Secure Relay RA to handle the streaming of incoming and outgoing data flows from the DMZ to the protected network and vice versa. Communication channels are always initiated from the Gateway side (Secure Relay MA).

The Communication channel (or Data channel) provides an Axway proprietary protocol that enables you to convert any number of physical multiplexed stream connections established between the DMZ and the outside world into a fixed set of physical connections.

Communication channel security

Communication channel security relies on an administration-oriented dialog that Gateway and Secure Relay establish via the Hot channel.

SSL

The Communication channel TCP/IP connection can be secured using TLS transport security.

By using X.509 certificates (strong authentication) on both Gateway and Secure Relay RA sides, TLS ensures that:

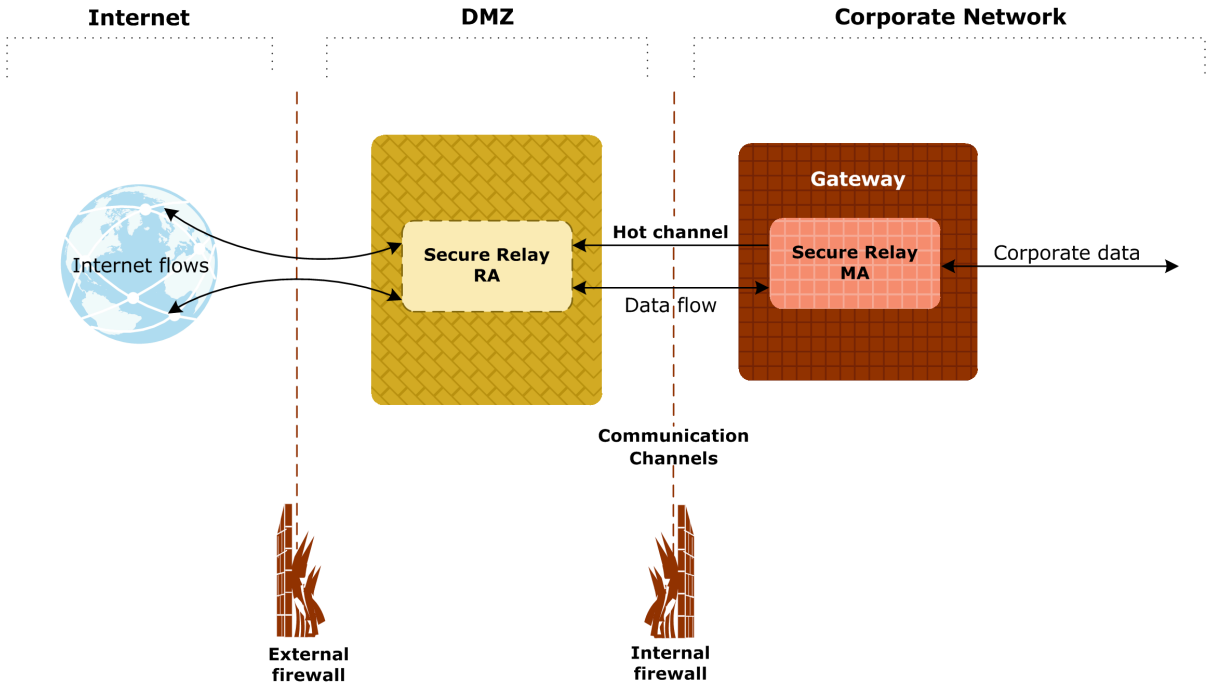
- Only Gateway (via Secure Relay MA) and the Secure Relay RA can communicate with each other, avoiding communication hijacking.
- No one can interfere with the data exchanged between Secure Relay MA and the Secure Relay RA.
- No one can even sniff the network, or understand what operation is being processed and try to take advantage of it.

Because you can configure the number of Communication channels in the Gateway configuration menu and can also define (and restrict) the local port range used on the Gateway host machine, Communication channels are also firewall-friendly.

It should be noted that securing the Communication channels between Secure Relay and Gateway using mutual authentication and encryption may have an impact on the performance of the solution.

Physical connections

The following figure illustrates the physical connections established between the DMZ and the protected zone using the Secure Relay solution.



Security features

TLS termination in DMZ

The Secure Relay Router Agent is able to handle the whole TLS dialog and security for incoming and outgoing connections for the FTP, HTTP and PeSIT E protocols. It authenticates the remote partner by using PassPort PKI services (located in the intranet) via the Secure Relay Master Agent (Secure Relay MA). Only then, is the incoming connection forwarded to the corporate network. It also uses PassPort PKI services to authenticate itself to the remote partner.

In the case of FTP, both implicit and explicit termination are now supported by the Secure Relay Router Agent (Secure Relay RA).

- In the case of FTP, both implicit and explicit termination are supported by the Secure Relay Router Agent (Secure Relay RA).
- for the SecureRelay security termination, the communication with the PKI server (PassPort) is done

through the Master Agent

To activate this feature:

- Set up the TLS profile to use in Secure Relay when *configuring your listening ports* in the Gateway configuration menu, and
- Select **Transport security in SecureRelay for outgoing connection** in the *Remote Site object (Net security tab)*

Delegating the TLS termination to the Secure Relay component may have an impact on the performance of the solution.

Connection to PassPort

The Master Agent will read the configuration for the connection to PassPort from the advanced.properties file located inside the run_time/xsr folder.

A sample advanced.properties file is available in samples/xsr folder

You can copy this file to the run_time/xsr folder and edit it with the relevant configuration information.

Login termination in DMZ (protocol login)

As with SSL termination in the DMZ, it can be useful for protocols that have a built-in login mechanism to verify the credentials associated with an incoming connection in the DMZ. The Secure Relay Router Agent (Secure Relay RA) is then able to delay the establishment of the connection with its associated Master Agent until the credentials have been accepted by the Gateway server, using its standard security checking procedure. By using this feature, you ensure that the only connections accessing the Gateway server in your corporate network through Secure Relay RA are those matching the security parameters defined in the server.

The currently-supported protocols and authentication methods are:

Protocol	Authentication methods
HTTP	http basic, CGI user/password, CGI sessionId, cookie user/password, cookie sessionId
FTP	user/password

Activate this feature by checking the **Protocol login in SecureRelay** option in the Secure Relay SAP configuration for HTTP and FTP.

Warning: If DMZ login termination is active, you can only activate TLS on the Router Agent side (not on the Gateway server).

Delegating the login termination to the Secure Relay component may have an impact on the performance of the solution.

SFTP termination in DMZ

The Secure Relay Router Agent (Secure Relay RA) is able to perform SFTP connection authentication in the DMZ. It can handle the whole SSH dialog and security for incoming and outgoing connections for SFTP. The authentication is done at both transport level (ciphers and certificates) and application level (user names and

passwords). It authenticates the remote partner by using PassPort PKI services (located in the intranet) via the Secure Relay Master Agent. Only then, is the incoming connection forwarded to the corporate network. It also uses PassPort PKI services to authenticate itself to the remote partner.

By using this feature, you ensure that the only connections accessing the Gateway server in your corporate network through Secure Relay RA are those matching the security parameters defined in the server.

Secure Relay RA supports password and public key authentication, in both server and client modes.

The following subset of SSH parameters is supported by Secure Relay RA:

Key exchange algorithms:

- DH_GROUP1_SHA1

Cipher algorithms:

- AES128_CBC
- 3DES_CBC
- NONE

Message authentication code (MAC) algorithms:

- HMAC_SHA1
- HMAC_MD5
- HMAC_SHA1_96
- HMAC_MD5_96
- NONE

Public key algorithms:

- SSH_RSA
- SSH-DSS

To activate this feature:

- Set up the SSH profile to use in Secure Relay when *configuring your listening ports* in the Gateway configuration menu, and
- Select **Transport security in SecureRelay for outgoing connection** in the *Remote Site object (SFTP tab)*

Warning: The selected SSH profile must provide PassPort entities to properly set up Secure Relay RA termination. Using the Gateway internal security server (SECS) is not supported.

Delegating the SFTP termination to the Secure Relay component may have an impact on the performance of the solution.

PKI exit

You can override Secure Relay's default PKI behavior for secure Communication channels and Hot channel. To do this, declare your own implementations of JSSE X509ExtendedKeyManager and X509TrustManager (javax.net.ssl package) in Secure Relay. Secure Relay then uses your implementations instead of its own to manage X.509 certificates and private keys.

Refer to the samples in <Secure Relay RA directory>/exit.

Refer to the Javadoc index.html in <Secure Relay RA directory>/exit/doc.

Password exit

You can override the Secure Relay Router Agent's default method of managing passwords. Declare your implementation of com.axway.xsr.security.pki.PasswordManager in Secure Relay. Secure Relay then calls your interface instead of reading or writing its standard password file.

Refer to the samples in <Secure Relay RA directory>/exit.

Refer to the Javadoc index.html in <Secure Relay RA directory>/exit/doc.

Management of JCE providers

Secure Relay does not explicitly use a particular security provider. It uses the standard Java JCE infrastructure. Therefore, you can set up your JRE to use any of your security providers, and Secure Relay will use it as well.

Management of multiple Secure Relay Router Agents

You can set up multiple instances of Secure Relay RA (eight maximum) running simultaneously on several machines. For high availability, the Secure Relay Master Agent automatically recovers errors that occur on any instance of Secure Relay RA. It also balances outgoing connections between available instances of Secure Relay RA.

You can not set up more than one Secure Relay RA per machine: the Secure Relay Master Agent sends the same configuration to every Secure Relay RA, so that every instance of Secure Relay RA listens on every TCP port you configured for Gateway's protocol connectors.

To configure Secure Relay RAs, refer to Configuration: Connectivity parameters.

For outbound transfers you can select the Secure Relay Router Agents to use for each partner. You make the selection in the Remote Site:

- On the **Network address** tab, use the **Allowed Router Agents** parameter, or
- With online commands, use the peladm update_site command and the *-xsr_allowed_ra_list (-xsr_arl)* parameter.

Logging in the DMZ

You can carry out logging in the DMZ. Accessing log files can provide valuable information when troubleshooting a given setup.

For security reasons, this feature is disabled by default (the logs can contain data that can be useful for an attacker). To enable logging in the DMZ, edit the Secure Relay RA configuration file as described in the *Axway Secure Relay Router Agent Administrator's Guide*.

Related topics

Overview: DMZ deployment

Working with Secure Relay

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Working with Secure Relay

Axway Gateway: Managing Security

Overview

This topic describes the steps you need to perform to set up Secure Relay for communications over the Internet.

1. Check the Prerequisites.
2. Install Gateway (which includes the Secure Relay Master Agent components) on your corporate network.
3. Install the Secure Relay Router Agent in the DMZ.
4. Configure and activate Secure Relay.
5. Confirm that the Secure Relay Router Agent is running.
6. Start the Secure Relay Master Agent.

Prerequisites

You need Java 1.5.0 or higher to operate Secure Relay. This is installed automatically when you install the Secure Relay Router Agent.

For more information, refer to the:

- *Axway Secure Relay Router Agent Administrator's Guide*

Installing Secure Relay Master Agent

The Secure Relay Master Agent is automatically installed when you install Gateway.

Installing Secure Relay Router Agent

The Secure Relay Router Agent is delivered on the Axway Platform installation DVD. Use Axway Installer to install the Secure Relay Router Agent on a machine located in the DMZ, independent of Gateway.

Configuring and activating Secure Relay

Configure and activate Secure Relay in the Gateway configuration menu:

1. In the left pane of the Gateway GUI, right-click the Gateway server that you want to configure and select **Configure**.
2. Select **Connectivity > SecureRelay**.
3. Complete the Secure Relay configuration parameters according to your network requirements.
4. Click **Activate** to enable Secure Relay.
5. Click **OK**.

Note: The Secure Relay modifications you make in the configuration menu only take effect after you restart Gateway.

Using a secure HTTP proxy

In a configuration that has Gateway, Secure Relay and a HTTP proxy, Secure Relay must be bypassed to avoid any issues.

To do so, add a mask of the IP address of the Remote Site in the Secure Relay bypass mask and *not* a mask of the IP address of the proxy.

Using Secure Relay Router Agent with privileged port numbers

If you want to use the Secure Relay Router Agent on privileged ports, in other words port numbers below 1024, you must do one of the following:

OS	Solution
UNIX	Change the owner of the Java executable used to run Secure Relay Router Agent and change the mode to -s. By default this file is the JVM installed with Axway Platform. Use the commands: chown root java chmod -s java
UNIX	Start Secure Relay Router Agent as root.
Windows	Use an Administrator account when installing Secure Relay Router Agent.

Encrypting the Secure Relay Master Agent certificate password

In Gateway, the password used to access the Secure Relay Master Agent certificate must be encrypted using the pelencpass command, while the master_cert_password_dk file, master_cert_password_salt file and master_cert_password_data file configuration parameters must be set accordingly. See the Secure Password Storage section for details.

Confirming that the Secure Relay Router Agent is running

When you start the Secure Relay Master Agent, it connects to the Secure Relay Router Agent(s) and communicates certain configuration parameters. Before you start the Secure Relay Master Agent, confirm that the Secure Relay Router Agent specified in the Gateway configuration menu is running.

Refer to the *Axway Secure Relay Router Agent Administrator Guide*.

Starting the Secure Relay Master Agent

The Secure Relay Master Agent starts automatically at the same time as Gateway (as long as the option **Starting/stopping of SecureRelay Master Agent is controlled by Gateway** is selected in the Secure Relay advanced options).

The *Master Agent process* is monitored by Gatecontroller, on condition that:

- Gatecontroller is activated and
- the Master Agent lifecycle is managed by the product.
  (if you start the Master Agent manually, it will not be monitored by Gatecontroller, and no action will be taken in case it becomes unavailable, crashes or gets stuck in some random processing).

This means that periodically, the Secure Relay Master Agent sends heartbeats to Gatecontroller, just like any other Gateway process. The Master Agent is considered a vital process of the product, meaning that if Gatecontroller detects a problem with it (if it is blocked for some time, or crashed), it restarts the product along with a new instance of Master Agent.

All actions regarding the monitoring of the Master Agent process are logged . Note that for all log entries that refer to Secure Relay Master Agent, the string XSR-MA is used instead of the longer alternative.

Related topics

About Secure Relay

Overview: DMZ deployment

Configuration: Connectivity parameters

Configuration: Secure Relay advanced options

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Paired Gateways in a DMZ

Axway Gateway: Managing Security

About paired Axway Gateway DMZs

To provide secure links between your corporate network applications and file-exchange partners located on the Internet, you can deploy Gateways in a DMZ protected architecture.

For an overview of Gateway DMZ solutions, refer to Overview: DMZ deployment.

You have the option of deploying paired Gateways in a secure DMZ bridge configuration. Alternatively, you can use the Axway Secure Relay option for DMZ configurations.

Using paired Axway Gateways in DMZs

General considerations

To create a secure and manageable DMZ for your network, keep in mind the following default rules for DMZ configuration:

- Do not enable default polling from either the public or private DMZ. Configure polling mechanisms only when strictly necessary.
- Do not enable FTP transfers between the intranet and the Private Gateway component of the DMZ.
- Do not enable NFS mounting. The file systems of applications within the DMZ should be strictly local.
- Rigorously limit the number of external systems that you permit to exchange transfers with the public Gateway. This enables you to open a limited and manageable number of ports on the firewall.

Roles of Gateways in the DMZ

When you create a DMZ by deploying paired Gateways, you configure the Gateways to operate together as public and private DMZ partners:

Private Gateway

A *private* Gateway contains definitions that enable the applications situated in your corporate network to act as servers and/or clients for exchanges with Internet partners.

You control the identity of intranet applications that connect to the private Gateway via Remote Site objects.

You control the protocol-level connections to the private Gateway via CGate objects.

On your private Gateway you decrypt incoming files. You can scan decrypted files for viruses before forwarding files to the target destination on your corporate intranet.

Public Gateway

A *public* Gateway contains definitions of all external Internet partners.

In your public Gateway, to control the identities and possible connection attempts of your external Internet partners, you create:

- Remote Site object definitions for Internet partners acting as servers
- CGate object definitions for Internet partners acting as clients

The temporary files that the public Gateway stores and all Gateway internal files (administration and Mailbox files) are encrypted to protect against malicious intrusions.

Exchanges between public and private Gateways

The public and private Gateways communicate via the PeSIT protocol (secured by SSL/TLS) or via SFTP.

Gateway DMZ transfers using the PeSIT protocol

The PeSIT protocol provides two basic methods that you can use to exchange files between the private and the public Gateways and between the private Gateway and Axway Transfer products located in the corporate network:

Method 1: The receiving Site is the sender/requester

To enable the public Gateway to initiate transfers towards the private Gateway, or to enable the private Gateway to initiate transfers towards the corporate network:

- Configure the PeSIT protocol in the configuration menu, specifying the access ports for both the public and private Gateway
- Configure PeSIT CGates for both the public and private Gateway
- Open a port for Gateway to Gateway PeSIT exchanges in the DMZ firewall application

Method 2: The receiving Site is the sender/server

In this case, using the GUI you set the following attributes of the Remote Site object that defines the receiving Site:

- Clear all *initiator* role check boxes in the **General** tab
- Select the *Change direction support* option of the **PeSIT** tab
- Enter a polling interval in the **PeSIT** tab

Example exchange scenario:

The private Gateway (acting in client mode) calls the public Gateway (acting in server mode). The private

Gateway then either sends files to, or asks to receive files from the server. The server application waits for client connection requests to receive files from the client, or to transmit files to the client upon a selection request.

Enhancing error protection

To enhance error detection when using PeSIT (for example, during Store-and-Forward operations) you can select the *Negative acknowledgment option* in the PeSIT tab of the Remote Site objects between the two Gateways. This option activates PeSIT Hors SIT Negative EERP Support.

Rerouting (Store-and-Forward) mechanisms

After each change in the transfer state (Ended, Acked, Canceled, ToAck), a Gateway process triggers a Store-and-Forward mechanism.

For Store-and-Forward, the following process events occur in this order:

1. Gateway executes the ExitXferMonitor C exit, which enables you to set variables and specify routing behavior
2. Gateway executes the XferMonitor Perl exit, which enables you to set user variables and specify routing behavior
3. Gateway evaluates Decision Rules in order to apply one of the following actions:
    - Default routing
    - Application of a model and deposit of the corresponding routed transfer
    - Execution of a Perl script
    - Execution of a batch command
    - Routing to Axway Integrator or AMTrix
    - No action

Managing user exits

Note: Although the C-code and Perl user exits provide you with the possibility of customizing processing results, we recommend that you use the customization features of Store-and-Forward mechanisms that are available via Decision Rule management.

The user exits can return a result that blocks the progression to the next processing step. When the three processing events are sequentially dependent, and you customize an exit, it is important be sure that the structures modified by an exit are not overwritten by a subsequent processing step.

As illustrated in the step list above, the user exits ExitXferMonitor (in C) and XferMonitor (in Perl) are executed just before the Decision Rule mechanism.

The Perl exit is the most flexible and easy to implement. It requires no compilation, and does not require a stop/restart of Gateway to take into account any changes.

You can use the Perl exit to set user variables that are then available to define conditions in the Decision Rule objects, or to be used in associated Models. Some of these variables can be seen in the Gateway GUI as *integer user param (1 and 2)* and *string user param (1 and 2)*.

This enables you to apply specific processing via Decision Rules, based on the incoming file Transfer attributes.

Managing Store-and-Forward via Decision Rules and Models

Gateway provides two specific objects that you can use to reroute incoming files:

- Decision Rules
- Models

About Decision Rules

You automate routing by creating and managing Decision Rule objects. In a Decision Rule object, you define a set of conditions to apply to the current transfer. When all the Decision Rule conditions match those of the current transfer, Gateway applies an *execution type*, also specified in the Decision Rule, to the transfer.

About Models

For file Store-and-Forward (rerouting), you typically select *Model* as the resulting *execution type* for a Decision Rule. You can create Models that provide sets of attributes for submitting new Transfer Requests. When, as a result of the analysis of a Decision Rule, Gateway generates and submits a new Transfer Request, it extracts attributes from the specified Model. The main advantage of Models is that they enable you to automatically re-use the parameters values of an incoming transfer as values for an outgoing transfer via the application of symbolic variables.

Applying Decision Rules

In general, you should try to manage the least possible number of Decision Rules for your DMZs. Theoretically, you require only two:

- A Decision Rule on the private Gateway manages all the transfers coming from the corporate intranet. This Decision Rule applies a Model that reroutes incoming transfers towards the public Gateway.

  Note: You must include the identity of the final destination partner of the transfer in the rerouted transfer to the public Gateway, This enables the public Gateway to complete file routing or, when acting as a server, to indicate the availability of the file to the targeted client.
- A Decision Rule of the public Gateway manages all the transfers coming from the Internet. This Decision Rule applies another Model, (or alternatively, a script) that reroutes the incoming transfer towards the an application on your corporate intranet.

Naming conventions for DMZ transfers

Overview

When you establish file naming conventions for transfers via a DMZ, it is important that you allow yourself the capability to extend data exchanges for new applications you may add as exchange partners.

Generally it is advised that you replace partner applications names and transferred file names with logical names. From a production point of view, this will enable you to more easily identify which partner deals which kind of exchange.

Applying conventions

For physical file names, you can establish naming convention that uses symbolic variables to create names that correspond your production procedures.

For instance, you can establish a naming convention for your file transfer flow that includes sender and receiver partner in the transfer identity. For example, a flow will be identified by the name *XYnn* in one transfer direction and *YXnn* in the other.

Where:

- *X* (or in the other flow direction, Y) represents the sender application
- *Y* (or in the other flow direction X) represents the receiver application
- *nn* represents the flow type

FTP and HTTP file transfer issues

If a file sending partner application situated on the Internet is an FTP or HTTP client, the Gateway application_name attribute has no meaning. In this case, either:

- Use the FTP or HTTP quote site command to specify the application name before you upload files
- Specify the application name in place of the physical file name in the FTP or HTTP put command
- Configure Gateway to deduce the application name from the transfer

Temporary and database file storage

Protecting data on the public Gateway

To protect the data on your public Gateway from malicious intrusions, you must activate encryption of the temporary files stored on the public Gateway and all Gateway internal files (administration and Mailbox files).

Virus scanning and file storage

It is important to complete the Store-and-Forward mechanism by scanning received Internet files for virus, before sending them towards your corporate network applications.

Virus scanning is not possible on the encrypted files of the public Gateway. For incoming Internet files, you execute virus scans on the *private* Gateway, after the decryption process.

To set up virus scanning on the *private* Gateway, you can do either of the following:

- Use a Decision Rule
  Define a Decision Rule that calls a script (batch/shell or Perl) at the end of the file transfer reception. This script executes a virus scan on the file, and if the file is *clean,* it then executes the forwarding Transfer Request via the peltrans line command.
- Use C-code or Perl exits
  Customize an exit so that the file will be scanned for viruses. If the file is *clean*, the exit returns a positive answer, permitting the continuation of Store-and-Forward mechanism. The forwarding Transfer Request is deposited as defined in the Decision Rules (for example, via a Model).

Gateway file systems

In order to preserve the existing environment with FTP, HTTP or SFTP clients, use RFD (Real File Directory) to store the client files.

Example

Define a Virtual File Directory, /ftp/user1, that is mounted on a physical directory, for instance /data/user1.

In this case the name of each file must be unique. You can use a symbolic variables to generate file names. For example you could use x_local_ident, which represents the transfer identifier of the Gateway Mailbox record.

Reliability management

Use the time slot defined by the application for exchanges to trigger retry processes. You can set maximum retry limits and time delays between each retry in the Remote Site object PeSIT tab.

You can use synchronization points during the transfers to optimize the flow control on network and minimize the data to re-transfer in case of failure. Set synchronization in the Remote Site object PeSIT tab.

In addition to the transfer level control, you can activate an application control level using acknowledgment messages. Set acknowledgment options in the Remote Site object PeSIT tab.

When you activate this feature, public Gateway generates a message containing the process status (correct/incorrect), and sends the message back to the transfer originating application.

Monitoring file transfers

You can use Axway Sentinel to provide end-to-end monitoring of Axway MFT activities. Sentinel enables you to view the status of all network application activities from a single point, in an Internet browser.

All information related to Axway Transfer and Axway Gateway file transfer activity is sent to the Sentinel Server and stored in a database (ORACLE, SQL server).

Sentinel can provide graphic views of transfer Cycle phases. These displays shows links between successive related file transfers. You can configure Sentinel to generate Cycle graph displays in which the color of displayed nodes (transfer events) depends on the last transfer state (Ended, in Alert, Available, Acked…).

Transfer scenarios using paired Axway Gateways in the DMZ

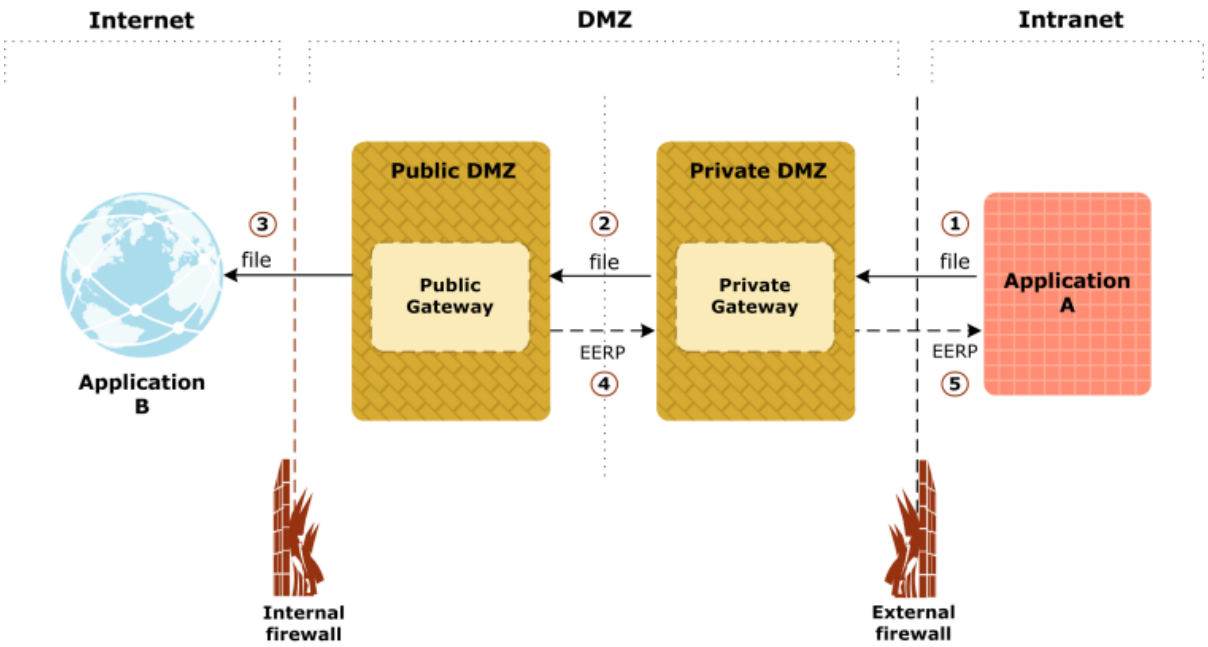
This section contains examples of file transfers between Internet and intranet partners in which paired Gateways provide DMZ security for corporate applications.

This section describes the events that constitute a transfer:

- *From an intranet application to an Internet partner*
- *From an Internet partner to an Axway Integrator application on the corporate intranet*

Scenario 1: Intranet partner to Internet partner

This scenario describes the typical events that occur when transferring a file from an intranet application (such as Axway Transfer) to an Internet partner application.

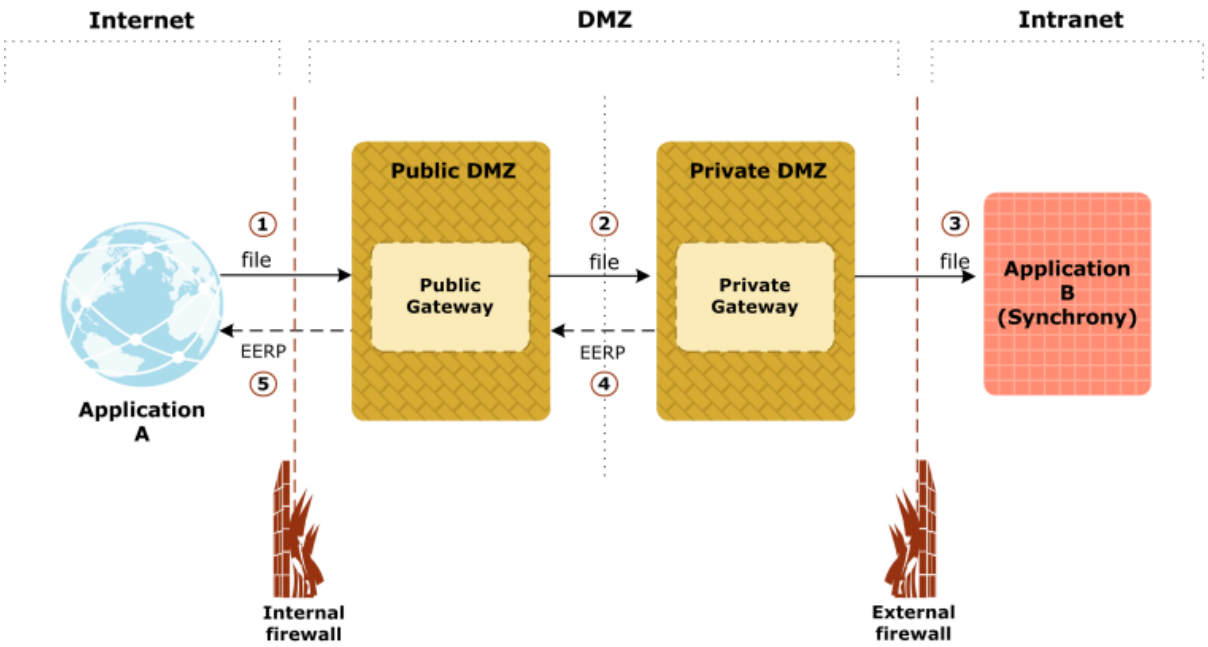


The transfer illustrated above is composed of the following steps:

1. Intranet application A (for example, an Axway Transfer product) sends a file in initiator mode to the private Gateway via the PeSIT HS E protocol. The final destination is specified in the Transfer Request. The private Gateway accepts the transfer via the generic CGate for PeSIT partners. The partner parameters match a Remote Site defined on the private Gateway.
   The file is received correctly on the private Gateway.
2. The private Gateway analyzes the incoming transfer using a Decision Rule. It applies a Model specified in the Decision Rule to forward the file to the public Gateway via the PeSIT HS E protocol.
   The file is received correctly on the public Gateway.
3. The public Gateway analyzes the incoming transfer using a Decision Rule. It applies a Model specified in the Decision Rule. If the destination is a server, the public Gateway sends the file directly to the final destination. Otherwise, the public Gateway makes the file available to the client in the specifies home directory, for example /ftp/Y/download/.
   The file is sent correctly to the FTP server, or alternatively, the FTP client downloads the available file.
4. The public Gateway automatically generates and sends back an acknowledgment to the private Gateway. The acknowledgment includes the final destination (the original file sender).
5. The private Gateway automatically routes the acknowledgment to the final destination. The transfer state then changes to *Acknowledged*.

Scenario 2: Internet partner to intranet partner

This scenario describes the typical events that occur when receiving a file transfer from an application located on the Internet to an Axway Integrator application situated on the intranet.



The transfer illustrated above is composed of the following steps:

1. Internet client application A in initiator mode sends a file to the public Gateway. The transfer could, for example be executed using FTP.
   The public Gateway can accept the transfer only through a CGate specified for this client. The public Gateway dynamically generates a Remote Site object for the transfer connection by using a Template Site.
   The public Gateway correctly receives the file (encrypted).
2. The public Gateway analyzes the incoming transfer using a Decision Rule. It applies a Model specified in the Decision Rule to forward the file to the private Gateway via the PeSIT HS E protocol.
   At the end of the reception on the public Gateway, the handling process calls an exit that executes a virus scan. If the received is infected, the file is deleted and an error procedure is started. In this case, a message can be sent to the destination application specifying that an infected file has been received and deleted.
   In this example, the file is virus free, and is received correctly on the private Gateway.
3. The private Gateway analyzes the incoming transfer using a Decision Rule with execution_type = XIB. It applies a Model specified in the Decision Rule. The private Gateway sends the file directly to the destination Integrator application.
   The file is sent correctly to Integrator.
4. The private Gateway automatically generates and sends back an acknowledgment to the public Gateway.
5. The transfer state of the file transfer issued in step 1 now changes to *Acknowledged*. If the destination application accepts acknowledgment, the public Gateway applies a Model via a Decision Rule to route the acknowledgment to the final destination.

Managing DMZ-related transfer failures

Using negative acknowledgment between paired Gateways

An easy way to track Gateway DMZ errors is to implement the negative acknowledgment option between the paired Gateways.

When you have selected this option and a transfer is canceled, a negative acknowledgment is automatically sent from one Gateway to the other one.

The following examples show selected Mailbox records from paired DMZ Gateway partners of a failed file transfer.

Example – Public Gateway Mailbox


TEST GW_PRV Ils DEFAULT_A PeSIT E TRANS Responder Receive nack by USER (U)?


TEST Ils Ils DEFAULT_A FTP TRANS Initiator Send Cancelled (C) A


TEST GW_PRV MONITOR DEFAULT_A PeSIT EERP Initiator Send Ended (E) 5

Example – Private Gateway Mailbox


TEST MONITOR GW_PUB DEFAULT_A PeSIT TRANS Initiator 8311080 nack by USER (U)?


TEST Ils GW_PUB DEFAULT_A PeSIT EERP Responder 8311080 Ended (E) 7

Mailbox record description

104: The private Gateway sends a transfer to the public Gateway

185: The file is received and the transfer state changes to *Ended* in the Public Gateway

186: The public Gateway sends the file to an IIS server, but the transfer fails and the transfer state changes to *Cancelled*

187: The public Gateway sends a negative acknowledgment back to the private Gateway

105: The private Gateway receives the negative acknowledgment

104: The transfer state on the private Gateway changes from *Ended* to *nack by User*

File availability management : date_to_end

On the public Gateway, in order to simplify error tracking, you can define a date_to_end for each Transfer Request.

Then, if the client does not pick up the file, or if the file is not sent to the server before the specified date_to_end, the corresponding transfer state changes to *Cancelled*.

To specify the date_to_end, you define a Model object on the public Gateway.

You can specify a relative date_to_end. For example: **+000010** means after 10 minutes the file transfer will be canceled if nothing happens in the meantime.

In this case the following message code is returned to the public Gateway:

Last end reason	Latest transfer date reached

Route state value

Gateway automatically sets a route state value for each Transfer Request. The possible values for this route state are:

- Acked (A)
- Routed (R)
- Failed (F)
- To Route (T)
- Not Routed (N)

You can use the route state value to check for instances in which a routing fails. For example, if the final destination for a transfer does not exist in the Public Gateway, the detailed Mailbox view of the corresponding incoming transfer request displays a *Failed* route state value.

Traces in the Rule Table properties

By default, Gateway generates traces of network activities and records these traces to your log file.

You can activate an additional set of tracing mechanisms by setting the Log Level option in the **General** tab of the Decision Rules object creation window.

The following examples compare the results of the two types of tracing, for two different transfer failure cases.

Case 1: The Remote Site does not exist on the public Gateway

- Default tracing results on the public Gateway:

  Gateway sets the route state to *Failed* but nothing is logged in the Log file.

  Note that if you have defined acknowledgment by Monitor in the remote site GW_PRV of the public Gateway settings, a *positive* acknowledgment is sent automatically to the originator.

- Decision Rule *short trace* results in the public Gateway:

  The public Gateway log contains the following trace entries:

  NOT442I 11.02.2005 12:07:21 Examining rule table [DEFAULT].
  NOT421I 11.02.2005 12:07:21 [31138] Decision rule [FROM_GTW_SRV] selected, execution type: "Model".
  NOT423I 11.02.2005 12:07:21 [31138] Model TO_CLI_SRV will be applied.
  NOT434E 11.02.2005 12:07:21 [31138] For General Model TO_CLI_SRV, dest_alias FOUNDRY1 is not defined.

If you are also using Axway Sentinel to monitor transfer activities, in Sentinel you have the corresponding alert message:

2005.02.11 12:31:26 NOT434E	[31141] For General Model TO_CLI_SRV, dest_alias FOUNDRY1 is not defined	GTW-UNIX	GW_PUB 3 EC 1

Case 2: The directory for the client does not exist on the public Gateway

- Default tracing results on the public Gateway:

  The route state is set to *Failed.* Nothing is logged in the Log file.

  If you have defined acknowledgment by Monitor in the GW_PRV Remote Site object of the public Gateway, a *positive* acknowledgment is automatically sent to the originator.

- Decision Rule *short trace* results in the public Gateway:

  If you have defined acknowledgment by Monitor in the GW_PRV Remote Site object of the public Gateway, a *positive* acknowledgment is automatically sent to the originator.

  In the public Gateway log:

  NOT421I 11.02.2005 12:52:47 [31149] Decision rule [FROM_GTW] selected, execution type: "Model".
  NOT423I 11.02.2005 12:52:47 [31149] Model TO_CLI will be applied.
  NOT409E 11.02.2005 12:52:47 [31149] Routing failure: XferRequest error (ErrCod = VFD directory not found).

  If you are also using Axway Sentinel to monitor transfer activities, in Sentinel you have the corresponding alert message:

2005.02.11 12:53:38 NOT409E	[31150] Routing failure: XferRequest error (ErrCod = VFD directory not found)	GTW-UNIX	GW_PUB 3 EC 1

Related topics

Overview: DMZ deployment

About Secure Relay

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

TCP port management with Firewalls

Axway Gateway: Managing Security

About TCP port management

Support for multiple addresses

About TCP port management

Gateway provides advanced port management features that ensure its compatibility with firewalls.

To use the TCP/IP network with a transfer protocol, you define the TCP/IP listener ports. You can declare or define single or multiple listening ports.

Define multiple listening ports as follows:

- Assign different routing rules to a single user
- Define access to specific clients on a specific port

Support for multiple addresses

If the Gateway host machine is connected to multiple TCP/IP networks, and is assigned more than one IP address, it may need to listen on multiple TCP/IP addresses (that correspond to a list of port numbers).

Gateway enables you to define the list of ports to use for a given TCP/IP address. When you define a Remote Site, you can attribute a main address and three alternative addresses in the Network address tab.

Assigning TCP origin port

To force the use of specific origin ports for outgoing TCP connections, you define those ports in the file tcpoport.ini. This file is located in <Gateway installation directory> \runtime\etc.

Gateway bases its selection of the origin port on the destination address for the connection. If a connection request matches a pattern defined in tcpoport.ini, Gateway assigns the corresponding origin port to the connection. If Gateway does not find a match, the operating system assigns the origin port.

Each line of tcpoport.ini has the form:

dest_ip dest_ports: org_ports

or:

! dest_ip dest_ports: org_ports

where:

dest_ip = IP address (10.1.1.1), IP pattern (10.*.*.*) or DNS name (machine.domain)

dest_ports = port number (21), port range (2200-2300) or any port (*)

org_ports = port number (65001) or port range (65000-65100). Special port 0 is used to let the system choose the origin port.

The destination address is matched based on the parameters dest_ip and dest_ports. If both parameters match, the corresponding origin port is used. If one of them does not match, the rule is ignored.

If a line begins with an exclamation mark (!), the rule matches any outgoing TCP connection that does not match the destination address or destination port. If you use a destination port of "*" that is not preceded by the "!" character, the result could be unexpected behavior.

Examples

The following examples use the local IP network 10.*.*.* .

Example 1: If you only want to use origin ports between 42000 and 43000 for connections going out of your network, add the following line to the tcpoport.ini file:

! 10.*.*.* *: 42000-43000

Example 2: If you want to use origin ports between 64000 and 64100 for connection toward your local network (like in a DMZ), add the following line to the tcpoport.ini:

10.*.*.* *: 64000-64100

DNS names are resolved when needed and only if they have not been resolved in the previous five minutes.

Origin port assignment does not override FTP protocol specifications. Data connections triggered by the server (active mode) use origin port L-1 (with L the listening port).

Related topics

Working with Remote Sites

Working with Remote Sites (command line)

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Performance cost of encryption

Performance impact on various mailbox operation when data files are encrypted, N=1000 transfers:

Operation	Average times	Performance decrease
Create one item	2.33ms/2.39ms	2.74%
Update one item	1.85ms/1.97ms	6.15%
Select one item with criteria	2.125ms/2.144ms	0.91%
Select one item	945ms/1.968ms	1.19%

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Configuring Secure Relay Router Agent parameters

This topic lists the Secure Relay Router Agent parameters that you can modify in the *Router Agent parameters* window. Gateway opens this window if you click the **Modify** button when configuring Secure Relay.

1. Complete the fields of the *Router Agent parameters* window as follows:

Parameter	Description
General	
Active	Select this option to activate the Router Agent.
Name	Enter the name you want to assign to the Router Agent.
Address	Enter the network address of the Router Agent.
Admin port	Enter the Router Agent connection port to use for administrative exchanges with the Master Agent. Default = **6810**
Comm port	Define the Router Agent connection port to use for file exchanges with the Master Agent. Default = **6811**
Advanced	
Number of multiplexed connections between Master and Router	Specify the number of simultaneous connections that can be created between the Master Agent and Router Agent. Integer: 1 - 100 Default = **5**
Data channel ciphering	Select this option to cipher the Communication channels between this Router Agent and the Master Agent in TLS. This internal ciphering is completely independent of whether the transported data is ciphered or not.
Incall network interface	Network interface to bind to in order to accept incoming connections.
Incall data port range	*FTP and AS3 only* Port range for data connections in client active and server passive mode.

Parameter	Description
	Enter the lower limit in the **Min** field and the upper limit in the **Max** field.
Outcall network interface	Network interface to bind to when connecting to a partner (for example, 123.123.123.123).
Outcall data port range	Port range for data connections in client passive mode. Enter the lower limit in the **Min** field and the upper limit in the **Max** field.

2. Click **OK** to confirm the *Router Agent parameters.*

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Configuring Secure Relay - advanced options

Axway Gateway: Configuration

This topic lists the Secure Relay parameters that you can modify in the *Secure Relay advanced options* window. Gateway opens this window if you click the **Advanced...** button when configuring Secure Relay.

Options in the Secure Relay advanced options window

To set the options below:

1. Complete the fields of the *Secure Relay advanced options* window as follows:
2. Click **OK** to confirm the *Secure Relay advanced options*.

Parameter	Description
Startup	• *Starting/stopping of SecureRelay Master Agent is controlled by Gateway.* Select this option if you want Gateway to control starting and stopping of the Master Agent. Default = *selected* • *Startup delay max (in seconds) when SecureRelay is started by Gateway* In this field, enter the maximum limit in seconds during which Gateway will accept Secure Relay connection attempts during startup. After the limit has expired, Gateway generates a Secure Relay connection failure message.

Parameter	Description
	Default = **30**
Master Agent settings	This section includes the following fields: • **Address**: Specify the Master Agent address for communication with Gateway. Default = 127.0.0.1 (localhost) • **Comm port** Specify the local port you want to use for data links between Gateway and the Master Agent. The port you use must be unique for the machine. Default = 6801 If the local machine already uses port 6801, select another unused port • **Outgoing admin port set** *Optional* Specify one or more ports for outgoing administrative messages towards the Router Agent. If you do not define this parameter, the Master Agent automatically assigns an available port. • **Outgoing data port set** *Optional* Specify one or more ports for outgoing data transfer towards the Router Agent. If you specify outgoing ports, you must specify at least as many ports as the number of connections specified in the *Number of multiplexed connections between Master and Router* field. If you do not define these ports, the Master Agent automatically assigns available ports.
PKI Server	This section refers to the connection between SecureRelay and PassPort. • **Use PassPort PKI server** Use this option to enable SecureRelay to use PassPort as PKI server. • **Connect using TLS** Use this option to enable TLS connection between XSR MA and Passport.Default is *enabled*. • **User Login** Username used for connecting to PassPort. • **Address** PassPort server address. • **Port** PassPort listening port. Default = 7101 • **Server CA file** PassPort's CA certificate. This option is needed only when enabling TLS.

Parameter	Description
	• **Password salt** Location of the password salt file. • **Password dk** Location of the password derived key file. • **Password data** Location of the file containing the encrypted key. See *Security Guide* > *Password management* for further information.
Protocol login server	This section includes the following fields: • **Address** The address of the protocol login server used when protocol login in DMZ is enabled (usually the address of the machine hosting Gateway). • **Port** The port to use for listening to login requests when protocol login in DMZ is enabled

IP filtering in Secure Relay	This section includes the following field:
	• **IP filtering in Secure Relay enable IP filtering in SecureRelay based on**: Select this option to enable IP filtering in Secure Relay. This is a global setting which applies for all Secure Relay Router Agents and all service access points. The IP filtering can be disabled individually per service access point (Sap) – [*Configuring protocols*: *SAP parameters*]. The following filtering methods are available:
	  • *White list*: Select this option and provide the location (absolute path) and file name of the static white list that Secure Relay will use to filter the incoming connections.*
	  • *Black list*: Select this option and provide the location (absolute path) and file name of the static black list that Secure Relay will use to filter the incoming connections.* **
	  • *Dynamic white list*: Select this option to enable Secure Relay to build and use dynamic white lists
	  • Note: When the *dynamic white list* is enabled, if in Black list mode, Secure Relay populates the dynamic white list (builds the list) with the accepted IP addresses. Then, when switching to White list mode, Secure Relay uses the dynamic list in addition to the static white list
	
	* The white list or the black list will be provided in files. The lists must comply with the following rules:
	• the list separator is `";"`

Parameter	Description
	• wildcards are supported for the IP address • specifying an IP range is allowed • IPv4 / IPv6 standards are supported • FQDN format for host names is allowed ** It is possible to reload the blacklist without restarting Gateway, using the command pelctl reload_xsr_ipfilter_file

Options in Secure Relay relating to Gateway

The Secure Relay options below are not directly set in Gateway, but they have an effect on the working of Gateway. The table below provides a short introduction, with a link to the relevant documentation for Secure Relay.

Parameter	Description
Connection Limiter	Connection Limiter is a feature in Secure Relay that can limit the number of concurrent connections, and the number of incoming connections from an IP address or from a group of IP addresses. By default, the connections that exceed the configured limits are dropped; but it is also possible to create a temporary blacklist to reject further incoming connections for a configurable period of time. Connection Limiter security feature was added in Secure Relay 2.7.4 (available September 2021). To activate Connection Limiter with Gateway there are 2 possibilities: • using the `p_xsr_connlimit_conf` environment variable, which must be set to the full path to a file containing connection limits configuration, or : • by placing the configuration file in the `run_time/` directory with the filename `connlimit.properties`. A sample for Connection Limiter is available in the `run_time/xsr/samples/` directory. For more information about Secure Relay's Connection Limiter configuration file option: Connection Limiter configuration file .

Related topics

Configuration: Connectivity parameters

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Payload integrity

Introduction

Inbound transfers

When activated, this feature enables Gateway to attach signatures to all received files, and to use that additional information when the file is routed to a third party. This approach helps to detect any eventual tampering of the files stored by Gateway. When this functionality is enabled, Gateway computes the signature of the payload received from incoming transfers and in case of routing, payload signature validation is performed. When Integrator is used, Gateway forwards it the signature computed for incoming data and validates the signature received from Integrator before routing the payload further.

Enabling this feature makes Axway Integrator perform a signature verification with Gateway's public key before processing the file. This causes failure for any file submitted by Integrator that is not accompanied by the signature information.

The key used by Gateway for signature validation at routing:

• is the Gateway public key if the signature was manually set into peltrans, in the sigfile option

• is the Integrator public key if the signature was manually set into the model used to route the file, in the sigfile option

In more detail: if the following conditions are met:

- the global configuration is activated (if payload_integrity_option is set to yes)
- the necessary parameters are set: Gateway private key, optionally the password of the key, Integrator public key

then after Gateway has received data on SWIFTNet, PeSIT or JMS, it does the following:

- computes the signature of the payload using SHA256 hash algorithm,
- signs in with Gateway private key
- stores it in the incoming transfer parameters (x_sigalgo, x_signature).
- if the incoming transfer is routed to Integrator, the signature of the transfer is also sent.

Requests submitted locally

Note that data integrity is not enforced on requests submitted locally in Gateway. However there are mechanisms that can enable payload integrity on transfers initiated via command-line tools, or Navigator.

Outbound transfers

Payload integrity is also enforced on the outbound transfers, either submitted a Gateway operator or by Integrator. This is achieved by attaching a signature when a transfer request is submitted to Gateway. In this case, the first operation that is performed when the transfer is about to start is integrity verification. There are two distinct use cases:

- when a local operator submits a transfer on which data integrity should be performed, the signature that is attached must be generated with the private key of Gateway, so that when the transfer is actually started the verification is performed with the pair of this key.
- for outbound requests submitted by Integrator, the signature must be generated with the private key of Integrator. In the case, when the transfer starts, verification is performed with the pair of that key.

Requirements

Using the payload security mechanism described in this chapter supposes you have the following:

- a pair of keys to build/verify signatures, with a public key to perform signature verification on requests submitted by others
- local operations, including requests submitted directly in Gateway (via command line or GUI), are performed only with the Gateway key or keys.
- The option is activated only when payload_integrity_option is set to yes. See *Working with Payload integrity*

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Working with Payload integrity

Openssl example

Local submissions that are subject to payload integrity verification can have their signature generated using the openssl utility, for example:

openssl dgst -sha256 -sign <gateway private key>.pem -out signature.sign <payload file>

peltrans -fc <payloadfile> -sigfile signature.sign -sigalgo SHA256withRSA ...

Note that the two parameters are exposed also in the model. The only difference in this case is that the signature attached via model is verified using the public key from Integrator. This is done on purpose to support the reference mode in which Integrator submits transfers via JMS. The assumption in this case is that the JMS message payload is not the business payload (what ends up in file component parameter) and can be used to store the signature generated by Integrator. This means that when routing the inbound JMS transfer in another transfer, the sigfile parameter would refer the file_component of the JMS transfer, and the rest of the parameters of the resulting transfer would be filled from metadata (JMS properties included) attached to the JMS transfer.

The file referred by the sigfile parameter must contain only the bytes of the signature in binary format.

These parameters are also displayed using peldsp command as x_sigalgo and x_sigdata, containing the signature algorithm and the signature bytes encoded in hex format.

Configuring payload integrity from the command line

To activate this feature, set the payload_integrity_option to *yes* in the monitor section of the configuration.

The rest of the parameters are stored in dedicated section named integrity:

The private key of Gateway (and also its public key) can be configured with the private_key parameter. The value is the path to a file containing a RSA private key in pem format.

The private key file can be password-protected, and the password can be configured via the private_key_dk and private_key_pf parameters. The values for these parameters are a path to a derived key file and a path to a password file. The two files must be generated with the pelencpass utility (See the Security Guide for further information).

The public key of Integrator can be configured with the public_key parameter. The value is a path to a file containing a RSA public key in pem format.

Configuring payload integrity from Axway Navigator

All the parameters, except the one referring to the Integrator public key are in Gateway/Database/Protection section of the Configuration window.

The public key of Integrator can be configured in the Connectivity section dedicated to Integrator.

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*

Setting Data Integrity

Enabling data integrity

By default, the data integrity check functionality is disabled (no). To enable it.

peluconf set -s monitor payload_integrity_option yes

If you enable the data integrity check, you must set the next parameters:

peluconf set -s integrity private_key %location to Gateway private key (RSA pem file) %

peluconf set -s integrity public_key %location to Integrator public key (RSA pem file) %

If you enable the data integrity check, you can optionally set the next parameters that stores the password for the Gateway private key:

peluconf set -s integrity private_key_dk %location to the derived key file%

peluconf set -s integrity private_key_pf %location to the encrypted password file%

The derived key file and the encrypted password file are generated using pelencpass tool.

Transfer parameters for data integrity

sigfile and sigalgo parameters can be used to manually set the signature of the payload data (necessary mostly for JMS reference mode part). The signature must be generated using Gateway private key. The sigfile parameter

references a file containing the signature in binary format.

New transfer parameters will be displayed for the incoming transfers, for which the signature of the payload in computed by Gateway on reception:

x_sigalgo='SHA256withRSA'

x_signature (256 bytes)= '65336230633434323938666331633134396166626634633839393 6

66623932343237616534316534363439623933346361343935393931623738353262383535…'

New parameters have been added to the model:

peladm create_model

[-sigalgo(-sga) signature and hash algorithm used in payload

signature computation { (SHA256withRSA) }]

[-sigfile(-sgf) file containing payload signature in binary format]

Note: if you create a transfer using a model, you must encrypt the payload with Integrator private key. If you route a transfer using a model, the validation of the signature is made using the Integrator private key.

Links to documentation set for Axway Gateway 6.17.3:

- *Installation* -- *User* -- *Unix Configuration* -- *Upgrade* -- *Interoperability* -- *Security*, requires login -- *Release Notes*