

# Security

**Note:**

This chapter applies only when the HTTP server is running in the SMARTS server environment.

Security on the Internet is a major concern of installations publishing IBM mainframe data.

The SMARTS server environment is fully integrated with the Security Authorization Facility (SAF) on OS/390 and MVS/ESA systems and can thus work with CA-ACF/2, RACF or CA-Top Secret. This integration involves

- verifying with the security system any provided user ID and password; and
- building an ACEE for each user that logs on to the system.

The SMARTS server environment then ensures that when the user issues a request to access any resource, the ACEE associated with that request is the one built for the user logging on. In this way, the security system can control access from multiple users with different security profiles from the same address space.

This chapter covers the following topics:

- The Default User
- HTTP Server Security Integration
- Natural Security Considerations
- Implementing SAF Security

---

## The Default User

When the HTTP server initially starts processing a request, it knows nothing about the user until it reads various HTTP header areas. Even then, there may be no information about the user.

For this reason, each request is assigned a 'default' user ID using the HTTPUSER configuration parameter. The SMARTS server sees the 'default' user ID for the duration of the request unless the user has provided some authorization information in the HTTP headers.

When security is active, the 'default' user ID receives the CA-ACF/2, RACF or CA-Top Secret authorization of the SMARTS server address space.

To better identify the default user, Software AG recommends that you specify DEFACEE=YES in the configuration parameters. The HTTP server then builds a default ACEE for the user ID specified by HTTPUSER and ensures that each user running with the default user ID runs with an ACEE built for that default user. Note that when DEFACEE is specified, the user ID specified in HTTPUSER must be defined to the security system; otherwise, the HTTP server initialization fails.

## HTTP Server Security Integration

The HTTP server integrates with the SMARTS server security facilities fully by passing on to the security system for verification any user ID and password information provided with a HTTP request. If the user ID and password is verified, from that moment on, any request initiated on behalf of that HTTP request is verified based on the user ID provided.

However, because different HTTP servers have different security requirements, it is possible to run the HTTP server in a number of modes as defined by the LOGON configuration parameter.

### **Note:**

SAF security processing must be active in the underlying SMARTS server system before the HTTP server security processing functions correctly.

### **Logon Allowed (LOGON=ALLOWED)**

This is the default mode in which the HTTP server runs. In this mode, all HTTP requests are accepted and dispatched without any logon requirements from the user. They are dispatched with the authorization of the default user as discussed earlier.

If the HTTP request attempts to access a resource to which the default user does not have access, a response is sent to the browser requesting that the user provide authorization information; i.e., a user ID and password. When this is submitted, it is verified with the underlying system and the request may then be repeated using the user's security profile. If the access attempt also fails for security reasons, the user's request is rejected.

This mode is intended for servers where unsecured and secured resources are available. Secured resources are only available to users that provide a valid user ID and password that has access to the secured data. Unsecured resources are available to all users that can connect to the server.

### **Logon Required (LOGON=REQUIRED)**

When this mode is specified, the HTTP server requires a user to provide valid authorization criteria (that is, a valid user ID and password) with each request to the HTTP server. The first time a user attempts an access without authorization information, the HTTP server requests this information from the browser, which in turn requests it from the client.

This information is verified with the underlying security system: access to the server is only permitted if the user ID and password are successfully validated. From that point on, anything that the user does within the server is checked against the security profile for the user ID provided. If a resource is requested to which the user has no access, the request is simply rejected as the user already provided authorization criteria.

This mode is intended for servers where only secure resources are available or servers that should only be used by authorized personnel (that is, by people defined to the security system).

### **Logon Disallowed (LOGON=DISALLOWED)**

When this mode is specified, the HTTP server ignores any authorization information provided as part of the HTTP request. All users connecting to the server run with the authority of the default user. In this way, the default user is only allowed access to that data that should be publicly available on the Internet.

This mode is intended for servers that are available publicly and where the installation does not want user IDs and passwords to be submitted over the net to the server.

## HTTP User ID and Password Encryption

When authorization information is provided to an HTTP request, it is encrypted using a simple and publicly available encryption mechanism. It is more secure than TELNET and FTP, which submit passwords in clear text over the net.

The HTTP server security logic fully secures an installation where the network itself is a "trusted" network. Generally, only Intranets may be considered trusted.

Where a server is available publicly on the Internet, a number of additional measures can be used to improve the security offered by this mechanism:

- Force people to change user IDs and passwords on a regular basis.
- Use conversational CGIs. Once the user ID and password are provided on the first CGI request of a conversation, it is no longer necessary to send them again until the conversation terminates.
- Allocate only user IDs and passwords for short periods of time to allow access to the system over a restricted time frame.

The encryption mechanism may also be useful for controlling access to resources that a server provides. While the data may not be sensitive, perhaps only users who have paid for a given service should be able to access the data. The encryption mechanism can ensure that only those who have been supplied with a valid user ID and password can access the system.

## Natural Security Considerations

When Natural acquires control with AUTO=ON, the user ID that is active in the SMARTS server environment is supplied to Natural Security. Where no logon has occurred, this is the user ID defined using the HTTPUSER configuration parameter. If a user provides configuration information and this is accepted and verified by the HTTP server, the user ID provided for the logon is passed to Natural Security.

Once again, Natural Security could be set up to restrict the public HTTPUSER user ID while allowing increased access based on a user ID for which a valid security system user ID and password is supplied.

## Implementing SAF Security

Security must first be implemented in the SMARTS server environment system by specifying the configuration parameter SECSYS. This parameter is used to inform the SMARTS server environment whether RACF, CA-ACF2 or CA-Top Secret is in place.

Refer to the SMARTS Installation and Operations Manual for more information about configuration parameters.

Once the security system is active, you must determine the appropriate level of access for each HTTP server and set the required LOGON configuration parameter in the server's configuration.

