USP SECURE ENTRY SERVER®

Documentation Series

UNITED SECURITY PROVIDERS

Appliance

Migration Guide

Version 5.18.0.2

United Security Providers AG
www.united-security-providers.ch
info@united-security-providers.ch

Headquarter Stauffacherstrasse 65/15 CH-3014 Bern Tel. +41 31 959 02 02
Baslerpark Mürtschenstrasse 27 CH-8048 Zürich Tel. +41 44 496 61 11

UNITED SECURITY PROVIDERS

UNITED SECURITY PROVIDERS

Contents

1 Introduction

The USP Secure Entry Server® appliance is a web application firewall acting as a reverse proxy with authentication capability.

1.1 Purpose of Document

This guide provides instructions for migrating a previous version of the USP Secure Entry Server® appliance to the most recent one.

1.2 Target Audience

The document is intended for system administrators of the United Security Providers Secure Entry Server® appliance.

1.3 SES Glossary

Table 1: Glossary

Term	Definition
SES	USP Secure Entry Server®, the entire software suite, including the Web Application Firewall, the Authentication Service and the Administration Web GUI.
WAF	Web Application Firewall, the entire web application firewall part of the SES. It acts as reverse proxy and consists of the three components "HTTP Listener" (HTL), "HTTPS Listener" (HTS) and "SRManager" (SRM). The composition of these components is technically called HSP.
HSP	HTTP Secure Proxy, the technical name of the WAF.
HTL	WAF HTTP Listener, the HTTP listener component of the web application firewall.
HTS	WAF HTTPS Listener, the HTTPS listener component of the web application firewall.
SRM	WAF Request Manager, the session management component of the web application firewall.
SLS	Secure Login Service, the technical name of the Authentication Service component that is responsible for authenticating and authorizing end-users.
Target server	An application backend server which is the actual target of an incoming HTTP request.

1.4 Important General Note

The SES migration mechanism supports only the automated migration of three versions back. So when updating from an older version (e.g. 5.8 to 5.12) then it is necessary to upgrade to the next intermediary release first, possibly also more, depending on how much older the currently used release is. For example, in order to update a 5.8 SES to 5.12, an intermediary update to 5.9 (at least) would be required!

Skipping the intermediary updates as described above will result in a failed update procedure!

Also, it is absolutely necessary to read all the migration guides of all intermediary releases, and check if there are any manual steps / adjustments that need to be done, based on the features and configuration currently in use.

2 SES Appliance 5.18

2.1 Summary

This migration guide provides a set of guidelines to migrate the USP Secure Entry Server® from versions prior to 5.17 to the latest version 5.18.

In most cases, no actions are required and all changes are done automatically. However, some minor manual changes might be required by the update. Consult the migration guide carefully prior the update and apply any steps required before or after the update.

When upgrading existing installations, please follow the recommended upgrade procedure outlined in Section 6.

If you are updating from a Secure Entry Server® Appliance older than version 5.17, please consult also the migration guides for all intermediate versions, as this migration guide only outlines the changes between the new and the previous USP Secure Entry Server® version.

It's no longer possible to upgrade directly to SES 5.18, if the version of the upgrade system is older than SES 5.15. Also is it not possible to import a configuration export or backup created with a version older than SES 5.15!

2.2 Removal of free MaxMind GeoIP database

Due to changes in the licensing terms of the MaxMind GeoIP databases, it is no longer possible for USP to include a free GeoIP country database in the USP SES Appliance. It is now mandatory for our SES customers to register themselves at MaxMind and create an account, and then use that account to retrieve credentials for downloading the free GeoIP databases. These credentials must be configured in the SES GUI.

In cases where, as described above, a setup is migrated from the previously included free USP MaxMind license to a new free license of the SES customer, the checkbox "Provide commercial license" *must remain unchecked*, but the credentials for the free license must still be configured. If the credentials are not configured, the MaxMind GeoIP DB update will not work anymore.

In other words, previously it was only necessary to configure an account and license key if a commercial license was used, but now the credentials must always be configured. The checkbox now only defines if the free (aka "GeoLite") or the commercial (aka "GeoIP") databases will be downloaded.

2.2.1 What to do

These instructions are to be executed after the SES update has been installed:

- If the installed SES already used a commercial license, nothing should change, and no configuration changes should be required.
- If the free DB was used, it is necessary to
  - configure new credentials by registering an account on the MaxMind website and retrieving them there
  - select the desired type of free database (city or country)
  - once saved, run the update manually using the "Run update. . . " button to verify that the configuration is valid

NOTE: *If the GeoIP feature was already enabled before the SES update, but the included free database was never updated (either manually by uploading a new one, or automatically online) so that the SES still contains and used the originally included free database, the database will NOT be available anymore after the SES update! In this case, it is necessary to upload a new free database (can be downloaded from the MaxMind website). If, however, the GeoIP DB has been updated before, it will remain unchanged and still be available after the SES update (but it won't be automatically updated any further until valid credentials are configured).*

2.3 Removal of factory reset

The options to perform a "factory reset" or a rollback to the previous appliance version have been removed from the GRUB boot menu. The reason for this is that these options began to cause issues with updates from older systems due to diskspace problems, caused by the multiple system images needing to be stored on the system, and the fact that the update images have grown larger and larger with newer releases. Additionally, performing a rollback after a system had already been updated usually caused problems anyway due to the configuration already having been migrated to the new release.

The second reason is that in this day and age, with the option of creating and restoring snapshots very easily and quickly with virtual machine platforms, the built-in restore and rollback have become very much obsolete.

3 SES Appliance 5.17

3.1 Summary

This migration guide provides a set of guidelines to migrate the USP Secure Entry Server® from versions prior to 5.16 to the latest version 5.17.

In most cases, no actions are required and all changes are done automatically. However, some minor manual changes might be required by the update. Consult the migration guide carefully prior the update and apply any steps required before or after the update.

When upgrading existing installations, please follow the recommended upgrade procedure outlined in Section 6.

If you are updating from a Secure Entry Server® Appliance older than version 5.16, please consult also the migration guides for all intermediate versions, as this migration guide only outlines the changes between the new and the previous USP Secure Entry Server® version.

It's no longer possible to upgrade directly to SES 5.17, if the version of the upgrade system is older than SES 5.14. Also is it not possible to import a configuration export or backup created with a version older than SES 5.14!

3.2 SLS Tomcat Update

The Tomcat version used by the SES GUI, Identity and SLS has been upgraded to 9.0.109. For the GUI and IDM, this will be active automatically once the system is rebooted after the update has been installed. For the SLS, however, a restart may be necessary, since the init script for the SLS is deployed by the GUI, and the new script using Tomcat 9 will not be active until a restart.

Note
A simple activation in the GUI may not suffice, since it will not restart the SLS if there are not pending SLS configuration changes. Therefore it is recommended to also explicitely restart the SLS manually after the activation.

3.3 SLS Content Security Policy (CSP) Introduction

The SLS is introducing support for CSP / Content Security Policy in this new release. In order to increase security levels in general, such a policy is now pre-configured in the SLS by default.

This new Content Security Policy will break existing SLS JSPs if they contain embedded scripts or styles! Since there is no automatic migration functionality for JSPs, this HAS TO BE addressed manually!

These are the options for preventing the SLS from not working anymore after the update (before activating the configuration again):

- Recommended: Update the JSPs beforehand by externalizing all embedded scripts and styles. This can be done before the update (see instructions below).
- Alternatively, either remove the CPS by clearing the new input field for it in the "General Settings" of the SLS, or adapt it by adding the "unsafe-inline" expression, e.g.:

```
... script-src: 'self' 'unsafe-inline' ...
```

However, both options reduce the security level provided by CSP, so a migration of the JSPs is the best way to go.

3.3.1 Migration / Upgrade considerations

Please note: **If you enable a content security policy in an existing SLS installation, embedded scripts, event handlers and styles will not work anymore!** You will have to update the JSPs as described here to make them work properly with an active CSP.

3.3.1.1 JSP Migration

In existing JSPs, you will usually see an embedded JavaScript which is included server-side by the JSP with the instruction

```
<%@ include file="/WEB-INF/include/js/sls.js"%>
```

In all JSPs, replace this line with this one:

```
<sls:javaScript localFile="include/js/sls-content-only.js" />
```

This new JSP tag "`sls:javaScript`" will generate the HTML "`<script>`" tags, but with the "`nonce`" attribute required for CSP to work properly.

Additionally, in WebAuthn-related JSPs, you have to replace these includes as well:

```
<%@ include file="/staticfiles/includes/base64.js" %>
<%@ include file="/staticfiles/includes/webauthn.js" %>
```

Replace them with

```
<sls:javaScript localFile="include/js/base64.js"></sls:javaScript>
<sls:javaScript>
    <%@ include file="/WEB-INF/include/js/webauthn.js" %>
</sls:javaScript>
```

Finally, if the page contains any includes for styles ("`.css`") files, replace them with external "`href`" references as well (see "Styles").

Note
The "`webauthn.js`" file MUST be included with a server-side include statement because the JavaScript code contains other SLS JSP tags; they have to be processed by the JSP compiler, and that will not be possible if the JavaScript is included in any other way.

Note
Include statements for actual include files (with the suffix "`.inc`") are not relevant for this issue and can be safely left as they are.

3.3.1.2 JavaScripts

If you have customized the JSPs and added any custom embedded JavaScript code, you will have to externalize it or enable it again using the "unsafe-inline" expression - but that kind of renders the whole thing pointless. There are two options for migrating existing embedded scripts:

- SLS "`javaScript`" JSP tag: Replace existing "`<script>`" HTML tags with embedded code by using the new JSP "`javaScript`" tag instead. As described in the paragraph further above, the "`<sls:javaScript>`" tag will generate a "`<script>`" tag with a "`nonce`" attribute.
- Externalize: Put the JavaScript code into a separate file in an external source, e.g. the "`/staticfiles`" folder, and reference it from there using the "`<script url=`" tag, with a URI pointing to the location of the file content.

Example for embedding a local file (path is relative to the SLS "WEB-INF" directory):

```
<sls:javaScript localFile="include/sls.js" />
```

Example for embedding custom JavaScript code directly:

```
<sls:javaScript>
   console.log('hello world');
</sls:javaScript>
```

This way, the SLS JSP tag will generate the HTML "`<script>`" tag with the required "`nonce`" attribute.

3.3.1.3 Styles

If you have customized the JSPs and added any custom embedded styles, you will have to externalize them or allow is again using the "`unsafe-inline`" expression (not recommended). However, there are two options for migrating existing embedded scripts:

- SLS "`style`" JSP tag: Replace existing "`<style>`" HTML tags with embedded styles by using the new JSP "`<sls:style>`" tag instead. As described in the paragraph further above, the "`<sls:style>`" tag will generate a "`<style>`" HTML tag with a "`nonce`" attribute. See the "JSP Tag *sls:style*" chapter in the "SLS Administration Guide".
- Replace inline "`style=`" attributes with "`class=`" attributes (see further below).
- Externalize: Put the styles into a separate file in an external source, e.g. the "`/staticfiles`" folder, and reference it from there using the "`<link rel=stylesheet href=...`" tag, with a URI pointing to the location of the file content, e.g.

```
<link rel="stylesheet" href="staticfiles/includes/acme-custom-style.css">
```

The important thing is that the styles must be loaded separated through a "href" URL, or that they use nonces.

NOTE: *Inline "style=" attributes do not work with CSP, as it is not possible to add a "nonce" attribute to another tag attribute (HTML tags can have attributes, but a tag attribute can not in itself have an attribute)! So all inline "style" attributes must be removed, as they will be blocked with an active CSP!*

Example for an inline "`style`" attribute that must be removed:

```
<span style="color: red; border: none;">Hello World</span>
```

Use the "class" attribute instead, with CSS class(es) defined in the properly included stylesheet file(s), e.g.:

```
<span class="helloWorldStyle">Hello World</span>
```

with a corresponding CSS class defined in an included stylesheet:

```
.helloWorldStyle {
    color: red;
    border: none;
}
```

3.3.1.4 Custom Event Handlers

A bit more complicated are any custom event handlers in HTML tags. Generally speaking, the migration of a simple "onClick" eventhandler for a "buttom" element consists of removing the handler from the tag, and instead adding a separate "<script>" tag with some JavaScript code which registers the intended action for the "click" event. The "<script>" tag must be created using the SLS "javaScript" JSP tag in order to get a proper "nonce" attribute.

Let's look at this example HTML tag:

```
<button type="submit" id="button-number-one" onclick="doThings('hello world');">DO IT</ ↩
    button>
```

This needs to be replaced with a construct like this:

```
<button type="submit" id="button-number-one">DO IT</button>
<script nonce="aosifwuhasdfjh">document.
addEventListener('DOMContentLoaded',
    function () { document.getElementById('button-number-one')
        .addEventListener('click', function callMethod()
            { doThings('hello world');
    });
});
</script>
```

So, if you need to do this for custom HTLM tags not created with SLS JSP tags, you can use the scripting function "`function.getRequestNonce()`" as shown in this example:

```
<script nonce='<sls:getScript expression="#{function.getRequestNonce()}"/>'>
   console.log('hello world');
</script>
```

3.4 Updating Web Application Firewall (HSP)

No migration steps are necessary for the WAF component.

4 SES Appliance 5.16

4.1 Summary

This migration guide provides a set of guidelines to migrate the USP Secure Entry Server® from versions prior to 5.15 to the latest version 5.16.

In most cases, no actions are required and all changes are done automatically. However, some minor manual changes might be required by the update. Consult the migration guide carefully prior the update and apply any steps required before or after the update.

When upgrading existing installations, please follow the recommended upgrade procedure outlined in Section 6.

If you are updating from a Secure Entry Server® Appliance older than version 5.15, please consult also the migration guides for all intermediate versions, as this migration guide only outlines the changes between the new and the previous USP Secure Entry Server® version.

It's no longer possible to upgrade directly to SES 5.16, if the version of the upgrade system is older than SES 5.13. Also is it not possible to import a configuration export or backup created with a version older than SES 5.13!

4.2 SLS Tomcat Update

The Tomcat version used by the SES GUI, Identity and SLS has been upgraded to 9.0.109. For the GUI and IDM, this will be active automatically once the system is rebooted after the update has been installed. For the SLS, however, a restart may be necessary, since the init script for the SLS is deployed by the GUI, and the new script using Tomcat 9 will not be active until a restart.

Note
A simple activation in the GUI may not suffice, since it will not restart the SLS if there are not pending SLS configuration changes. Therefore it is recommended to explicitely restart the SLS manually after the activation.

4.3 SLS ELCARD adapter support removal

The ELCARD authentication system by ELCA has been end-of-life for several years now. Because of that, support for it has been dropped from the SLS and the SES appliance.

As it was a completely proprietary authentication solution, there is no real, specific migration path. The general recommendation is to replace it with a similar standard authentication solution, such as OpenID Connect (OIDC) or SAML.

Note
The ELCARD-related JSPs can now be removed from existing appliances (but it is not neccessary). That would be "Elcard-Wait.jsp" and "ElcardCompleted.jsp".

4.4 Secure HandShake (SHS) Feature Removal / Migration

4.4.1 Introduction

Because newer TLS Versions (1.3) do not support re-negotiation at all anymore, the HSP-proprietary "Secure Handshake" (SHS) feature has been removed now, as it is based on triggering such a re-negotation on an existing TLS connection. This means that existing SLS/HSP setups that use SHS need to be adapted accordingly.

4.4.2 Terminology

- "PKI Login Model": Refers to the default PKI login model, as it is created from the template in the SES appliance GUI. This model contains only the states required to perform a secure handshake (SHS) and then verification of the client certificate, and mapping of the user credential, with the PKI adapter.
- "Mixed Login Model": Refers to a custom (sometimes very complex) login model that contains states for various types of login procedures, usually with conditional branching between them. For example, the model might decide to perform a PKI login if the client IP is from a certain IP range, or an SMS (HTTP adapter) login otherwise. Since every such model is by definition unique and tailored to a specific customer situation and requirements, this article can’t be any more specific about the details of such cases.

4.4.3 Are You Affected?

If you use any SLS login models with the state **"get.cert.cred"**, then it means you are using the SHS functionality. In that case, you will need to migrate your setup according to this description.

4.4.4 General Changes

4.4.4.1 Credential Provider

Without secure handshake, the certificate sent by the client will not be received in the SLS through the proprietary SHS headers sent from the SRM. Instead, after the HTTPS listener has performed a mutual authentication with the client (more about that further below), it will send the client certificate in a regular HTTP header to the SLS. On the SES appliance, this header is NOT being processed by the SLS by default, so in order for this to work at all, the following custom settings must be configured for the SLS:

```
cred.provider.pki=certificate
cred.provider.pki.cred.1.type=certificate
cred.provider.pki.cred.1.source=header
cred.provider.pki.cred.1.name=sslcertificate
```

"`sslcertificate`" is the name of the HTTP header in which the HSP will forward the certificate to the SLS. With this credential provider configuration, the SLS will process it, and a "`do.cert.auth`" model state can verify the certificate with the PKI adapter.

The http header "`sslcertificate`" does not have to be whitelisted in the HSP location configuration if the alias `normal` is configured.

4.4.4.2 PKI Login Model

For most scenarios and options discussed further below (except the SHS-only login case), a new separate PKI login model will be required in the SLS:

```
model.pkilogin.uri=/pkiauth
model.pkilogin.failedState=get.usererror
model.pkilogin.state.2000.name=do.cert.auth
model.pkilogin.state.3000.name=do.mapping
model.pkilogin.state.5000.name=do.success
model.pkilogin.state.9000.name=get.usererror
```

4.4.5 Migration Scenarios

4.4.5.1 Existing PKI login model

If you have an SLS with a pure (SHS-based) PKI login model as shown below. . .

```
model.pkilogin.uri=/auth
model.pkilogin.failedState=get.usererror
model.pkilogin.state.1000.name=get.cert.cred
model.pkilogin.state.3000.name=do.cert.auth
model.pkilogin.state.7000.name=do.mapping
model.pkilogin.state.8000.name=do.success
model.pkilogin.state.9000.name=get.usererror
```

i. the migration is very simple and straightforward:

The state "`get.cert.cred`" is the one which actually triggers the secure handshake. This state must be removed, and mutual authentication must be configured in the HSP (HTTPS listener) directly. So the required steps are:

1. Remove model state "get.cert.cred" - see new login model below
2. Add credential provider as shown above (SLS custom settings)
3. Add CA certificate installed in the PKI adapter (to verify the client certificates) also in the HTTPS listener (HTS)
4. Enable mutual authentication in the HTS on the virtual host

The updated login model:

```
model.pkilogin.uri=/auth
model.pkilogin.failedState=get.usererror
model.pkilogin.state.3000.name=do.cert.auth
model.pkilogin.state.7000.name=do.mapping
model.pkilogin.state.8000.name=do.success
model.pkilogin.state.9000.name=get.usererror
```

Note
Whenever the CA certificate used to verify the client certificates changes, it must be updated both in the SLS PKI adapter and in the HSP HTTPS listener configuration.

4.4.5.2 Existing mixed login model

In a more complex scenario where you have an SLS setup with a model which only runs through the SHS / PKI login part based on certain conditions, it gets more complicated. You have two major options, but one is preferrable over the other in most cases:

- Option 1: Make mutual authentication "optional" on the virtual host and create a new separate login location, then trigger the handshake on this new location (more about this below). For this, the PKI login model will be required on this new location. Then in the old mixed login model, when a condition is fullfilled to switch to PKI login, instead of using the PKI model states perform a redirect to this other login model (using the "cmd" parameter).
- Option 2: Create a new virtual host for the PKI login. On this virtual host, configure mandatory mutual authentication in the HTS and use a PKI-login model as described in the "SHS-only login" chapter. Then in the old mixed login model, when a condition is fullfilled to switch to PKI login, instead of using the PKI model states perform a redirect to this new virtual host.

Option 2 is most likely preferrable, because for Option 1, in order to trigger mutual authentication only for certain locations, using Apache mod_authz "Require*" expressions in the HTTPS listener will be required, making this a more complicated solution.

Also, another drawback of option 1: If the virtual host is set to `optional` client authentication, some locations may still enforce client authentication with an expression, but when the user first accesses any location on the virtual host, they must provide client authentication, as otherwise only locations without enforcement will be accessible (*no renegotiation possible*).

4.4.5.3 Option 1: location-based TLS handshake

- On the virtual host, set mutual authentication to "optional".
- Create a new location for the SLS, "`/login/sls/pkiauth`" in the HSP. Make it a login location by adding at least the following custom commands in the SRM:

```
AC_LoginPage                /login/sls/pkiauth
AC_StartPage                /login/sls/pkiauth
AC_StartQueryString         ^.*$
HGW_RequestHeaders          +%SRM_ls_std
```

- On this location, "/login/sls/pkiauth", also add the necessary Apache "Require*" statements statements as HTS custom commands in order to trigger the mutual authentication once the client is redirected to the SLS PKI login model.
  - see also IP Blocking and Granting in combination with IP-Restriction and Client Authentication (Requirement expression)
- Add a new SLS PKI login model "pkilogin" (see above)
- Register the URI "/pkiauth" in the SES GUI under "Configuration" / "Authentication Service" / "URI Mapping"
- Then in the old, existing virtual host, adapt the mixed login model. Replace the PKI login states with a redirect to the new certificate login. Note that the redirect still also uses the "cmd" parameter, because this is the only way to force the SLS into this login model in case it already has a session and is currently in another model:

```
... old mixed login model start ...

# Check if PKI login must be skipped (to continue with alternative login)
model.login.state.3000.name=do.generic-skipPkiLogin
model.login.state.3000.nextState.1=do.generic-continue

# Otherwise, redirect to new vhost with mandatory mutual authentication
model.login.state.4000.name=do.redirect-startPkiLogin
model.login.state.4000.param.url=/login/sls/pkiauth?cmd=pkiauth

model.login.state.5000.name=do.generic-continue

... continue with alternative login ...
```

4.4.5.4 Option 2: vhost-based TLS handshake

- First create a new separate virtual host for the certificate-based login. On this virtual host, enable mutual authentication as required.
- Add a new SLS PKI login model "pkilogin" (see above)
- Then in the old, existing virtual host, adapt the mixed login model. Replace the PKI login states with a redirect to the new certificate login (must include the host name):

```
... old mixed login model start ...

# Check if PKI login must be skipped (to continue with alternative login)
model.login.state.3000.name=do.generic-skipPkiLogin
model.login.state.3000.nextState.1=do.generic-continue

# Otherwise, redirect to new vhost with mandatory mutual authentication
model.login.state.4000.name=do.redirect-startPkiLogin
model.login.state.4000.param.url.absolute.allow=true
model.login.state.4000.param.url=https://my.new-vhost.com/login/sls/auth?cmd=pkilogin

model.login.state.5000.name=do.generic-continue

... continue with alternative login ...
```

4.4.5.5 Option 2: SAML / OIDC Considerations

In case of a separate new virtual host, it would also be possible to use federation mechanisms to delegate the login from the first login model on vhost "A" to the PKI login model on the new vhost "B". This would require a setup of a SAML IdP or OIDC OP on the new virtual host "B", and than registering the SLS of the old virtual host "A" as a Service Provider (SAML) or Relying Party (OIDC) - and then use the regular mechanisms of these protocols to delegete the login.

4.4.5.6 Additional Information

In case any conditional expressions must be used in the SLS - possibly in the PKI login model trigger - the following variables or expressions might be useful:

- `#{hasNonEmptyVar('header.sslcertificate')}` - Will check if there is a header "sslcertificate", which means the HTTPS listener has performed mutual authentication
- `#{function.getCurrentRequest()}` - Returns a reference to a Java "HttpServletRequest" object, which provides all kinds of getter methods that allow to obtain information about the current incoming request, such as its URI, headers etc.

4.5 Accept-Encoding Migration

The list of acceptable encodings has been extended from

```
gzip x-gzip compress deflate x-deflate identity sdch br
```

to

```
aes128gcm br compress deflate exi gzip identity pack200-gzip peerdist sdch x-compress x- ↩
    deflate x-gzip zstd
```

Accordingly, the default list of denied encodings per location has been automatically migrated from

```
gzip compress deflate x-gzip x-deflate
```

to

```
aes128gcm br compress deflate exi gzip pack200-gzip peerdist sdch x-compress x-deflate x- ↩
    gzip zstd
```

i.e. to all acceptable encodings except `identity`, because compressions make some features of the WAF ineffective.

If you have made different settings on locations to deny accepted encodings than the exact default above, migrate those settings manually — if necessary (per location setting under the tab "Filter" at "Denied request header entries").

5 SES Appliance 5.15

5.1 Summary

This migration guide provides a set of guidelines to migrate the USP Secure Entry Server® from versions prior to 5.14 to the latest version 5.15.

In most cases, no actions are required and all changes are done automatically. However, some minor manual changes might be required by the update. Consult the migration guide carefully prior the update and apply any steps required before or after the update.

When upgrading existing installations, please follow the recommended upgrade procedure outlined in Section 6.

If you are updating from a Secure Entry Server® Appliance older than version 5.14, please consult also the migration guides for all intermediate versions, as this migration guide only outlines the changes between the new and the previous USP Secure Entry Server® version.

It's no longer possible to upgrade directly to SES 5.15, if the version of the upgrade system is older than SES 5.12. Also is it not possible to import a configuration export or backup created with a version older than SES 5.12!

5.2 LDAP/AD User Synchronization with Roles from LDAP/AD

The option to import and synchronize users from LDAP/AD to local GUI users has been improved: GUI roles (Administrator, Configurator, Deployer, Viewer) are now assigned based on AD groups, namely via `memberOf` attributes in LDAP/AD.

For that purpose you can now define role mappings in the GUI. After the upgrade, those mappings will initially be empty. This means that if you had already set up LDAP user synchronization (possibly also with a cron job), nothing will happen, no sync, no users modified or deleted, until you define at least one role mapping.

IMPORTANT: Please make sure that at least one local Administrator GUI user exists and you know its password before defining role mappings, so in case of any issues you can use it to recover.

Then set up role mappings as desired.

If you trigger the synchronization manually, you will, as usual, be shown the users that would be deleted and could cancel if the result was not as desired, but if you set up a cron job and it happened to run, it can - in the worst case - remove all users imported from LDAP, leaving you only with local users, hence the above precaution.

5.3 Trace header filter disabled by default

Starting from this release the function “trace header filter” can only be enabled if the “traffic analyzer” is active. It is recommended to disable the “trace header filter” and “traffic analyzer” on productive system.

5.4 Updating Web Application Firewall (HSP)

5.4.1 HA Active/Passive Upgrade Problem

When a SES HA active/passive setup is updated, a transfer of the active sessions is NOT possible because the internal data structure of sessions in the session store of the HSP has changed.

If a takeover of an updated passive SES is attempted, the HSP will crash due to the incompatible session data!

Therefore, when updating from SES 5.14 or older to 5.15, it is MANDATORY to do it in a maintenance window, and restart both systems after the upgrade.

5.4.2 SE_L1Cds directives renamed

Some SRM directives starting with *SE_L1Cds* have been renamed (using the new the prefix *SE_SessionStore*) as due to the integration of the new *mod_geode_session_store* module they have influence on both - the L1 CD session store and the Geode session store.

The following table shows all renamed directives:

Old name	New name
SE_L1Cds_FinalTimeOut	SE_SessionStore_FinalTimeOut
SE_L1Cds_InactiveTimeOut	SE_SessionStore_InactiveTimeOut
SE_L1Cds_AnonymousInactiveTimeOut	SE_SessionStore_AnonymousInactiveTimeOut
SE_L1Cds_TimeOutRequestCounter	SE_SessionStore_TimeOutRequestCounter
SE_L1Cds_TimeOutRequestCounterTmo	SE_SessionStore_TimeOutRequestCounterTmo
SE_L1Cds_HostLock	SE_SessionStore_HostLock
SE_L1Cds_CClen	SE_SessionStore_CClen

Customers are adviced to review SRM configurations and to migrate existing settings. However, for backwards compatibility the old names will still work until the final removal of the directives is announced.

5.4.3 *Trace header filter* is a sub-feature of the *Traffic analyzer*

The *Trace header filter* feature of virtual host's Logging was moved to be a sub-feature of the *Traffic analyzer*. Hence, in all existing configurations, if *Traffic analyzer* is not enabled, *Trace header filter* will also be disabled.

NB The *Traffic analyzer* feature must not be enabled on a productive server since it captures user data and may generate high data volume which is written to the systems hard disk.

6 Recommended Upgrade Procedure

We strongly recommend following the steps listed below to update the USP Secure Entry Server® Appliance to the new release.

1. Ensure that no open changes exists and the latest configuration is activated.
2. **Backup** your configuration: before continuing with the update procedure,
    a. Create a system backup
    b. Export the current configuration
    Note: Backup and Recovery is described in detail in the **SES Appliance Guide**.
3. Take a snapshot of your current system if running the USP Secure Entry Server® as a virtual machine.
4. Check all the **prerequisites** in this migration guide are met.
5. Perform the **software update** as described in the SES Appliance Guide. For HA clusters in active/passive mode start with the HA node that is currently in passive mode.
6. **Review** the configuration. Settings that are configured in the GUI are updated (but not committed) automatically as needed, except Custom commands.
7. After the update, **check** that the **date and time** are set correctly. We recommend to always use an NTP server for time synchronization.
8. **Commit** and **activate** all the changes that were made due to the update. This should bring all the services back online.
9. For HA clusters in active/passive mode, next perform a **takeover** and update the other HA node.

6.1 Upgrade Fallback

If you decide to go back to the previous installed release, reboot the appliance and wait until the GNU GRUB menu appears during start up on the console. Select the second entry from the menu called **Restore old image**. This will restore the old image (User accounts will be reset too). Login with the default admin account and restore the system backup created before.

In case your running the USP Secure Entry Server® as a virtual machine, just restore the snapshot taken before the update.