USP SECURE ENTRY SERVER®

Documentation Series

UNITED SECURITY PROVIDERS

HSP

Migration Guide

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

Contents

1 Introduction

The USP Secure Entry Server® 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® to the most recent one.

1.2 Target Audience

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

2 HSP 4.27

2.1 Summary

This migration guide provides a set of guidelines to migrate the USP Secure Entry Server® Software Suite from prior version 4.26 to the latest version 4.27.

However, if you are updating from a Secure Entry Server® Software Suite older than version 4.26, please consult the migration guides for all intermediate versions.

2.2 Change in Header Entry Blacklist Handling for Value Entries

A bug in the header value parser has been fixed to correctly recognize parameters - such as the quality value (q) - attached to individual entries. For example, given the following Accept-Encoding header:

```
Accept-Encoding: br;q=1.0, gzip;q=0.8, identity;q=0.1
```

The parser now correctly tokenizes this into three distinct entries, each preserving its associated parameter:

- br;q=1.0
- gzip;q=0.8,
- identity;q=0.1

To maintain compatibility with existing header entry filters - such as the commonly used blacklist for *Accept-Encoding* - the filtering logic has been updated to use prefix matching. This means that if a header entry begins with a token listed in the filter, it will be removed.

Example filter directive:

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

Applying this filter to the example above results in the following modified request header:

```
Accept-Encoding: identity;q=0.1
```

While prefix matching ensures backward compatibility, it introduces a potential risk of over-filtering. A single filter token may match and remove multiple header values that share the same prefix. However, this behavior is not expected to affect standard HTTP headers.

The following directives are impacted by this change:

- RF_ServerRequestHeaderEntryDeny
- RF_ServerResponseHeaderEntryDeny
- RF_LocationRequestHeaderEntryDeny
- RF_LocationResponseHeaderEntryDeny
- HGW_ServerRequestHeaderEntryDeny
- HGW_LocationRequestHeaderEntryDeny

Customers who use custom commands with any of the affected directives are advised to review their header filter lists. This is especially important if custom header value entries are involved. Adjustments may be necessary to avoid negative side effects and ensure expected behavior. <<<

3 HSP 4.26

3.1 Summary

This migration guide provides a set of guidelines to migrate the USP Secure Entry Server® Software Suite from prior version 4.25 to the latest version 4.26.

If you are updating from a Secure Entry Server® Software Suite older than version 4.25, please consult also the migration guides for all intermediate versions.

3.2 Removal of Secure Handshake (SHS)

The secure handshake module (mod_shs) has been removed from SRM entirely. SRM will ignore SHS headers coming from the Login Service.

4 HSP 4.25

4.1 Summary

This migration guide provides a set of guidelines to migrate the USP Secure Entry Server® Software Suite from prior version 4.24 to the latest version 4.25.

If you are updating from a Secure Entry Server® Software Suite older than version 4.24, please consult also the migration guides for all intermediate versions.

4.2 Constraint on updates of HA Active/Passive setups

If HSP is used in an HA active/passive setup, it is not possible to transfer active sessions from the active (not yet updated) instance to the passive, updated instance. The reason for this is an internal change to the session data structure in the client data store.

Therefore, when updating from HSP 4.24 (or older) to 4.25, the update MUST be done in a maintenance window, as a restart of all HSP instances is necessary.

The transfer of incompatible session data would cause the SRM component of HSP 4.25 to crash.

4.3 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 HSP 4.24

5.1 Summary

This migration guide provides a set of guidelines to migrate the USP Secure Entry Server® Software Suite from prior version 4.23 to the latest version 4.24.

If you are updating from a Secure Entry Server® Software Suite older than version 4.23, please consult also the migration guides for all intermediate versions.

5.2 Version number of mod_security removed

With this release 4.24 the newest engine version 2.9.6 of mod_security is include. In the past we delivered also the 2.8.0 version of the engine. However, this version is out-dated and no more supported. Thus, version 2.8.0 is no more included in the delivery package.

For sake of simplicity the version number in the filename of the shared object file has been removed. As a result, the LoadModule directive need to be adjusted:

```
LoadModule security2_module libexec/mod_security.so
```

5.3 RF_Icap* directives removed

The directives starting with RF_Icap have been deprecated a long time ago. Now these directives have been removed from HSP. If you still have RF_Icap directives in your configuration it is necessary to replace them with the corresponding ICAP_ alternatives.

5.4 mod_analyze as shared module

The mod_analyze module is no more statically linked into HSP binaries but provided as shared module for both - HTS and SRM.

If *DBG_* settings are in your configuration, you should remove (or comment out) them, otherwise SRM or HTS will no more start. In case a traffic analysis is necessary, the mod_analyze module can be loaded via *LoadModule*.

Please note that mod_analyze should not be used on productive systems for performance and stability reasons. <<<

6 HSP 4.23

6.1 Summary

There are no steps needed to migrate the USP Secure Entry Server® Software Suite from prior version 4.22 to the latest version 4.23.

However, if you are updating from a Secure Entry Server® Software Suite older than version 4.22, please consult the migration guides for all intermediate versions.

7 HSP 4.22

7.1 Summary

This migration guide provides a set of guidelines to migrate the USP Secure Entry Server® Software Suite from prior version 4.21 to the latest version 4.22.

If you are updating from a Secure Entry Server® Software Suite older than version 4.21, please consult also the migration guides for all intermediate versions.

7.2 DBG_TraceMaxSize removed

The directive *DBG_TraceMaxSize* has been removed.

We recommend using a log file rotation tool (e.g. rotatelogs) instead which is setup by using a pipe output "|" on *DBG_TraceLog* settings. For example,

DBG_TraceLog "|rotatelogs /path/to/logfile 86400"

7.3 RF_SF_WSDLSchemaFile and RF_SF_TrustedNamespace removed

Due to clean-up work, the long outdated directives *RF_SF_WSDLSchemaFile* and *RF_SF_TrustedNamespace* have been removed.

If *RF_SF_WSDLSchemaFile* is present in configurations, customers are asked to simply remove corresponding settings. No additional migration steps are required.

If *RF_SF_TrustedNamespace* is present in configurations, customers are asked to use *RF_SF_TrustedSchema* instead, whereby it must be noted that the "namespace" parameter can be omitted.

7.4 SOAP filtering "log" mode

Due to clean-up work, the long outdated "log" mode of the SOAP validation module has been removed.

If SOAP validation is enabled, customers are asked to review configurations and to ensure that "log" is not used for the directives *RF_SF_SoapValidator* and *RF_SF_SoapValidatorResponse*.

In case "log" mode is used, it is necessary to switch to one of the alternative modes.

8 HSP 4.21

8.1 Summary

This migration guide provides a set of guidelines to migrate the USP Secure Entry Server® Software Suite from prior version 4.20 to the latest version 4.21.

If you are updating from a Secure Entry Server® Software Suite older than version 4.20, please consult also the migration guides for all intermediate versions.

8.2 TLSv1.3 support

Starting with version 4.21, HSP supports TLSv1.3 throughout. That is, both incoming connections from clients as well as outgoing connections to backend servers can be secured using TLSv1.3.

With TLSv1.3 some fundamental changes have been made in comparison to its predecessor TLSv1.2. Some of these changes result in limitations that are rather general, whereas other limitations are relevant to the use of TLSv1.3 within HSP only.

Before going in to detail of these limitations it should be noted that a separate document "TLSv1.3 in HSP" is available which provides general information about improvements in TSL 1.3 as well as an full overview of necessary changes for its use in HSP. Readers are invited to consult this document as a primer into the usage of the new TLS protocol in HSP.

For detailed information on TLS-specific directives the reader is referred to the HSP Administration Guide.

8.2.1 Post Handshake Authentication

TLSv1.3 does no more support renegotiation. In TLSv1.2 (and previous versions), renegotiation is often used to enable mutual/client authentication at a "Location based" level. TLSv1.3 instead provides the "post handshake authentication" (PHA) extension, which allows a server to request a client's certificate at any time after the handshake. However, this extension is optional and not required for the basic functionality of TLSv1.3 to work. In fact, most browsers do not yet support this extension (status as of August 2020), not least because of some uncertainties in specifications.

Therefore, it should be noted that the following two functionalities are based on PHA and therefore can not be used with TLSv1.3 in conjunction with client software not supporting the PHA extension.

- Secure Handshake (SHS)
- Mutual/Client authentication on a location level, it does work however at virtual host level (SSLVerifyClient require)

Please note that when using SHS with TLSv1.3 at least version 5.12 of SLS must be used to ensure the correct interaction with HSP.

8.2.2 SSL ID locking

The documentation of OpenSSL 1.1.1 states the following regarding session tickets and session IDs.

"The TLSv1.3 protocol only supports tickets and does not directly support session ids. However, OpenSSL allows two modes of ticket operation in TLSv1.3: stateful and stateless. Stateless tickets work the same way as in TLSv1.2 and below. Stateful tickets mimic the session ID behavior available in TLSv1.2 and below. The session information is cached on the server and the session id is wrapped up in a ticket and sent back to the client. When the client wishes to resume, it presents a ticket in the same way as for stateless tickets. The server can then extract the session id from the ticket and retrieve the session information from its cache."

In HSP the behavior can be controlled by the *SSLSessionTickets* directive. If set to "off", the stateful mode of TLSv1.3 is enabled. For security reasons, OpenSSL 1.1.1 may hand out multiple stateful SessionTickets per connection so that clients are motivated to use each SessionTicket only once (even though it is not strictly necessary). Research has shown that well-known browsers follow this best practice and use SessionTickets only once.

The consequence is that the TLS session ID changes on each request, which in turn leads to "false positives" when using HSP's Session ID Lock and Advanced Session ID Lock Rules features together with TLSv1.3. In other words: When using TLSv1.3 for secure communication, HSP version 4.21 does not yet provide the ability to correlate session IDs and thus guarantee the functionality of the mentioned features. However, we work on providing a more sophisticated solution for this in future versions. In the meantime, customers must adapt Session ID Lock Rules accordingly or fall back to the usage of TLSv1.2.

Please remind that with TLSv1.2, session tickets must be disabled in order to support Session ID Lock and Advanced Session ID Lock Rules.

8.3 Renegotiation issue with TLS 1.1/1.2

Tests with plain Apache httpd version 2.4.46 and OpenSSL 1.1.1 have revealed the existence of an issue with TLS renegotiation on proxy connections: when backend servers initiate a TLS renegotiation, httpd's TLS proxy engine behaves incorrectly leading to a "handshake failure" and finally to the abortion of the connection. As HSP uses httpd's TLS proxy engine, the issue can also affect outgoing connections on SRM to backend servers. At the time of writing it is still uncertain whether the behaviour of backend software plays a role in the error. Thus, it can not be concluded that the error occurs with any backend software.

Pleae note that the issue has been reported to Apache httpd's development community and as soon as a solution is found, a new version of HSP will be made available.

For most customers this issue has no impact as TLS renegotiaion is not used by the majority of backend applications due to security reasons. Or as renegotiation is simply not necessary for most of applications.

However, customers that encounter the issue we recommend to try to circumvent it by preventing the usage of TLS renegotiation on backend server side. For example, by preventing client authentication after the initial handshake, by increasing the TLS session timeout, or by setting suitable cipher suites on SRM that are accepted by the backend without any objections.

8.4 SSLv3 no more supported

Starting with version 4.21 SSLv3 is no more supported in HSP and thus, one of its successors (at best TLSv1.3) must be used to secure connections.

For this reason customers are asked to review settings of following directives and to change protocol specifications accordingly in case of obsolete SSLv3 usage.

- *HGW_SslClientProtocol*
- *HGW_SslClientProtocolDefault*
- *SSLProtocol*
- *SSLProxyProtocol*

9 HSP 4.20

9.1 Summary

This migration guide provides a set of guidelines to migrate the USP Secure Entry Server® Software Suite from prior version 4.19 to the latest version 4.20.

If you are updating from a Secure Entry Server® Software Suite older than version 4.19, please consult also the migration guides for all intermediate versions.

9.2 New default flag for regular expressions core component of httpd

Please note that this migration step must only be applied when updating to version 4.20.0.1 or higher.

With the update of httpd to version 2.4.43, a new default flag for PCRE (Perl Compatible Regular Expressions) was introduced to mitigate security vulnerability CVE-2020-1927. A bug in the core component could lead to unpredictable matches and substitutions, which then could be abused to fool the redirect component with encoded newline characters.

To avoid this, the PCRE flag *PCRE_DOTALL* is set by default. The flag causes the . character in regular expressions to also match the newline *\n* and return *\r* characters. Note that this affects all regular expressions and therefore has the possibility to break existing configurations.

Customers have two possiblitiies to migrate the configuration:

1. Restore old behavior:

The default PCRE flags can be controlled via the *RegexDefaultOptions* directive. With *RegexDefaultOptions -DOTALL* the above change can be reverted. Note that by doing so, the vulnerability becomes possible again.

2. Check the regular expressions

Regular expressions must be checked for possible logical errors due to the introduced change. The tricky thing is, that the configuration will still be valid i.e. all regular expressions will still compile. It completely depends on the use case and the matched/validated content, whether above described behavior is a problem or not. Nevertheless some rule of thumbs can help to ease the work:

- In general, whenever only one line is processed (e.g. a request header or the URL), the regular expression can be considered as safe.
- Often *(.*)* is used to "match the rest of the line" this can easily improved by adding the beginning of the next line to the expression: *(.*)\n<beginning_of_next_line>*
- Finally *(.*)* will match newlines, but with ? we can make the . lazy so that it only matches as many characters as needed for the rest of the pattern to succeed. Note that this could also have an impact on other whitespace characters. By adding a newline (or whatever should be the "end" of the match) this risk can be reduced: *(.*?)\n* <<<

10 HSP 4.19

10.1 Summary

This migration guide provides a set of guidelines to migrate the USP Secure Entry Server® Software Suite from prior version 4.18 to the latest version 4.19.

If you are updating from a Secure Entry Server® Software Suite older than version 4.18, please consult also the migration guides for all intermediate versions.

10.2 Dynamically loading of *mod_http2*

Please note that this migration step must only be applied when updating to version 4.19.0.6 or higher.

In HTS the linking of the HTTP/2 module *mod_http2* has been changed from static to dynamic. That is, the module is no more loaded automatically when HTS is started.

Customers that have HTTP/2 enabled must add following setting to the global section of the configuration in order the *mod_http2* module gets loaded on start-up.

```
LoadModule http2_module /path/to/mod_http2.so
```

The standard path of the shared object file is /opt/usp/hsp/hts/libexec/mod_http2.so.

The HTTP/2 protocol is enabled by adding "h2" to the *Protocols* directive. If "h2" should not be enabled but directives of *mod_http2* be present in the configuration, please remove all corresponding H2 settings. Otherwise, a start-up of HTS would fail.

10.3 New default flag for regular expressions core component of httpd

Please note that this migration step must only be applied when updating to version 4.19.0.10 or higher.

With the update of httpd to version 2.4.43, a new default flag for PCRE (Perl Compatible Regular Expressions) was introduced to mitigate security vulnerability CVE-2020-1927. A bug in the core component could lead to unpredictable matches and substitutions, which then could be abused to fool the redirect component with encoded newline characters.

To avoid this, the PCRE flag *PCRE_DOTALL* is set by default. The flag causes the *.* character in regular expressions to also match the newline *\n* and return *\r* characters. Note that this affects all regular expressions and therefore has the possibility to break existing configurations.

Customers have two possiblitiies to migrate the configuration:

1. Restore old behavior:

The default PCRE flags can be controlled via the *RegexDefaultOptions* directive. With *RegexDefaultOptions -DOTALL* the above change can be reverted. Note that by doing so, the vulnerability becomes possible again.

2. Check the regular expressions

Regular expressions must be checked for possible logical errors due to the introduced change. The tricky thing is, that the configuration will still be valid i.e. all regular expressions will still compile. It completely depends on the use case and the matched/validated content, whether above described behavior is a problem or not. Nevertheless some rule of thumbs can help to ease the work:

- In general, whenever only one line is processed (e.g. a request header or the URL), the regular expression can be considered as safe.
- Often *(.*)* is used to "match the rest of the line" this can easily improved by adding the beginning of the next line to the expression: *(.*)\n<beginning_of_next_line>*
- Finally *(.*)* will match newlines, but with ? we can make the *.* lazy so that it only matches as many characters as needed for the rest of the pattern to succeed. Note that this could also have an impact on other whitespace characters. By adding a newline (or whatever should be the "end" of the match) this risk can be reduced: *(.*?)\n* <<<

11 HSP 4.18

11.1 Summary

This migration guide provides a set of guidelines to migrate the USP Secure Entry Server® Software Suite from prior version 4.17 to the latest version 4.18.

If you are updating from a Secure Entry Server® Software Suite older than version 4.17, please consult also the migration guides for all intermediate versions.

11.2 Update of header definitions for *mod_header_validator*

The HTTP header syntax definitions used by *mod_header_validator* have been updated according to the RFC 723x series (RFC 7230, RFC 7231, etc.). In addition, new definitions and validator rules for following request headers have been added

- Origin
- Access-Control-Request-Method
- Access-Control-Request-Headers
- HTTP2-Settings

Customers that have *mod_header_validator* and the delivered configuration files *hsp_hv_rfc_compliance.conf* and *hsp_hv_srv_all_headers.conf* in use are asked to test setups before going live in order to ensure that syntax changes may not cause the removal of essential request headers (e.g. when non-RFC conform client software is in use).

11.3 New default flag for regular expressions core component of httpd

Please note that this migration step must only be applied when updating to version 4.18.0.6 or higher.

With the update of httpd to version 2.4.43, a new default flag for PCRE (Perl Compatible Regular Expressions) was introduced to mitigate security vulnerability CVE-2020-1927. A bug in the core component could lead to unpredictable matches and substitutions, which then could be abused to fool the redirect component with encoded newline characters.

To avoid this, the PCRE flag *PCRE_DOTALL* is set by default. The flag causes the *.* character in regular expressions to also match the newline *\n* and return *\r* characters. Note that this affects all regular expressions and therefore has the possibility to break existing configurations.

Customers have two possiblitiies to migrate the configuration:

1. Restore old behavior:

The default PCRE flags can be controlled via the *RegexDefaultOptions* directive. With *RegexDefaultOptions -DOTALL* the above change can be reverted. Note that by doing so, the vulnerability becomes possible again.

2. Check the regular expressions

Regular expressions must be checked for possible logical errors due to the introduced change. The tricky thing is, that the configuration will still be valid i.e. all regular expressions will still compile. It completely depends on the use case and the matched/validated content, whether above described behavior is a problem or not. Nevertheless some rule of thumbs can help to ease the work:

- In general, whenever only one line is processed (e.g. a request header or the URL), the regular expression can be considered as safe.
- Often *(.*)* is used to "match the rest of the line" this can easily improved by adding the beginning of the next line to the expression: *(.*)\n<beginning_of_next_line>*
- Finally *(.*)* will match newlines, but with ? we can make the *.* lazy so that it only matches as many characters as needed for the rest of the pattern to succeed. Note that this could also have an impact on other whitespace characters. By adding a newline (or whatever should be the "end" of the match) this risk can be reduced: *(.*?)\n*

12 HSP 4.17

12.1 Summary

This migration guide provides a set of guidelines to migrate the USP Secure Entry Server® Software Suite from prior version 4.16 to the latest version 4.17.

If you are updating from a Secure Entry Server® Software Suite older than version 4.16, please consult also the migration guides for all intermediate versions.

12.2 Removal of *hsp_tcp* script

The outdated script *hsp_tcp*, which was intended to set different TCP kernel parameters, has been removed from the delivery package. As a replacement, customers are asked to add such parameters to */etc/sysctl.conf* or any file in */etc/sysctl.d/*.conf* (on most standard Linux systems).

12.3 Information about port collision on loopback device

On heavy-loaded systems sporadic "address already in use" errors during graceful restarts have been reported when new virtual hosts have been added to SRM.

The reason were port collisions of sockets in time-wait state - used by HTS to connect to SRM on loopback device - with custom port numbers used for new virtual hosts on SRM.

To avoid such collisions it has to be ensured that the start of the ephemeral port range is always higher than the highest custom port number of any virtual host on SRM.

Customers are asked to review the kernel parameter *net.ipv4.ip_local_port_range* and - in case of an overlapping - to adjust the range accordingly in */etc/sysctl.conf*.

12.4 New load balancing module

The new load balancing module *mod_hgw_balancer* is based on *mod_proxy_balancer* and thus provides access to numerous new functions, such as changeable scheduling algorithms or configurable health checks.

mod_hgw_balancer can not be used together with the outdated module *mod_hgw_fo*, so migration is necessary when switching. However, using *mod_hgw_balancer and _mod_hgw_fo* on different locations (or even sub-locations) is entirely possible.

Please refer to the HSP Administration Guide, which provides a separate chapter on using *mod_hgw_balancer*.

For all information about *mod_proxy_balanacer* and its submodules we refer to the official websites:

https://httpd.apache.org/docs/2.4/mod/mod_proxy_balancer.html
https://httpd.apache.org/docs/2.4/mod/mod_proxy_hcheck.html

https://httpd.apache.org/docs/2.4/mod/mod_lbmethod_byrequests.html
https://httpd.apache.org/docs/2.4/mod/mod_lbmethod_bytraffic.html
https://httpd.apache.org/docs/2.4/mod/mod_lbmethod_bybusyness.html

12.5 Removal of `AC_AcceptedCiphers`

The obsolete directive `AC_AcceptedCiphers` has been removed from SRM.

Customers that use this directive are asked to remove it from any location in SRM and instead to use `SSLCipherSuite` in corresponding locations in HTS.

All details about `SSLCipherSuite` can be found on the official website of *mod_ssl*:
https://httpd.apache.org/docs/2.4/mod/mod_ssl.html#sslciphersuite

13 HSP 4.16

13.1 Summary

This migration guide provides a set of guidelines to migrate the USP Secure Entry Server® Software Suite from prior version 4.15 to the latest version 4.16.

If you are updating from a Secure Entry Server® Software Suite older than version 4.15, please consult also the migration guides for all intermediate versions.

13.2 HSP binary compatibility / requirement changes

From version 4.16 onwards until further notice new software requirements and compatibilties are in place. The HSP binary is binary compatible with Red Hat® Enterprise Linux® 6 Update 8 (or newer) and CentOS 6.8 (or newer).

Although only Red Hat® Enterprise Linux® 6 Update 8 (or newer) and CentOS 6.8 (or newer) will be supported, the HSP package may also install and run on platforms that fulfil the following software requirements:

- linux 2.6.32 (or newer)
- glibc 2.12 (or newer)
- libgcc 4.4.7 (or newer)
- libstdc++ 4.4.7 (or newer)

13.3 Processing of internal cookies

In version 4.16 some fundamental changes have been made regarding the processing of internal cookies.

13.3.1 Domain Matching

Internal cookies are now checked for Domain Matching in accordance with RFC 6265; the consequence is that if the hostname of the selected backend does not domain-match the value of the domain-attribute, the corresponding cookie is filtered. We call this newly default behaviour the "strict" Domain Matching Policy.

The default behaviour can be changed to "loose" by using following new directive

```
SE_IntCookie_DomainMatchingPolicy loose
```

"loose" disables any Domain Matching checks so that any cookie of any domain is accepted.

13.3.2 Persistent cookies "host-only" enforcement

Persistent cookies now are always enforced to be set to "host-only" on the corresponding client by removing the domain-attribute (if it exists). Thus, a client sends cookies only to the front-end host on which they have been set - and not to the whole domain, as in earlier versions.

If persistent cookies should be made available to a certain domain namespace, the new directive `SE_IntCookie_PersistentCookiesMapping` can be used to map cookies (by regular expression on cookie name) to a specific domain.

```
SE_IntCookie_PersistentCookiesMapping <regular-expression> <domain>
```

Please refer to the HSP Administration Guide for any further information about the new introduced directives.

13.3.3 Cookies from SLS

The Secure Login Service SLS up to version 5.9.x sends the "Version" attribute in the Set-Cookie header if the cookie is defined with "cookie.set" in the sls_instance.properties file. Also, some cookies had the "expire" attribute set instead of the "max-age" attribute. These attributes were deprecated by RFC 6265 and lead to parsing problems in HSP, or even complete removement of the cookie in worst case.

Note
It is therefore recommended to update SLS to a version 5.10.0.0 or later to guarantee compatibility with HSP 4.16.x and later on.

If you can't update SLS for some reason, you have to set cookies in a different way in the sls_instance.properties file. By using "hsp.header" instead of "cookie.set" it is possible to prevent the Version attribute in the Set-Cookie header:

```
# Create the Set-Cookie header by hand
hsp.header.Set-Cookie=my-login-cookie=my-cookie-value; Max-Age=28800; Path=/app

# instead of using cookie.<...>
#cookie.set.successful.name.1=my-login-cookie
#cookie.LoginTicket.value=my-cookie-value
#cookie.LoginTicket.max-age=28800
#cookie.LoginTicket.path=/app
```

13.4 Removing of AC_HspCredentialDomain

`AC_HspCredentialDomain` has been removed from the set of SRM's directives. Customers that have set this directive are asked to replace settings as following

```
AC_HspCredentialDomain <domain>
```

by

```
SE_IntCookie_PersistentCookiesMapping SessionCredential <domain>
```

As a side-effect of the removing, the SessionInfo header sent to Login Services during authentication steps has lost the `SrvCookieDomain` attribute. Customers who depend on this attribute in Login Services are asked to add following setting in corresponding login locations

```
RequestHeader edit SessionInfo (.*) "SrvCookieDomain=<domain>; $1"
```

13.5 Changes in translation and substitution functionalities

Because of a conflict in response header rewriting following changes have been made to path translation and response substitution functionalities.

13.5.1 Path translation

Up to version 4.15, when setting `HGW_TranslatePath` (or `HGW_TranslatePathMatch`), this also added an implicit reverse rewriting rule to modify the path attribute of Set-Cookie headers in responses. However, it did not create such implicit reverse translations for other headers with path related content, like the Location header.

As of version 4.16, when setting `HGW_TranslatePath` (or `HGW_TranslatePathMatch`), implicit reverse translation rules for path related content of the response headers *Set-Cookie* and *Location* are created. During response processing, these headers are automaticaly changed to fit the respective request path rewriting.

13.5.2 Response substitution

Up to version 4.15, when setting `HGW_ResFltSubstitute` (or `HGW_ResFltSubstituteMatch`) substitution rules were applied to the Location header of redirection responses and to the path attribute of Set-Cookie headers when present.

As of version 4.16, response substitution rules are only applied to the response body and no longer to any header. As a consequence, the directive `HGW_ResFlt_BodyOnly` became obsolete and thus, has been removed.

Customers that have `HGW_ResFlt_BodyOnly` in use are asked to remove it from configurations. If substitutions in Location headers or Set-Cookie headers are required, customers are asked to use the directive `Header` from the module mod_headers.

For details on `Header` the reader is kindly asked to refer to the official documentation that can be found at http://httpd.apache.org/docs/current/mod/mod_headers.html.

14 HSP 4.15

14.1 Summary

This migration guide provides a set of guidelines to migrate the USP Secure Entry Server® from prior version 4.14 to the latest version 4.15.

If you are updating from a Secure Entry Server® Software Suite older than version 4.14, 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.

14.2 Updating to USP Secure Entry Server 4.15

Due to a bug in `AC_StartQueryString`, the HSP-specific query string parameters `RequestedPage`, `HSPClientDataKey` and `PostDataStorageInfo` were not verified against the freely configurable regular expression of this directive.

The bug has been fixed in version 4.15 so that all three parameters are now considered upon verification.

Since the directive's default is to not allow any query string parameter for login requests, customers that have not set AC_StartQueryString in the configuration of the SRM must now add following setting

```
AC_StartQueryString "^([&]?(RequestedPage=[a-zA-Z0-9_\-\$]+)? \
                                  (HSPClientDataKey=[a-zA-Z0-9_\$]+)? \
                                  (PostDataStorageInfo=DataSize=[0-9]+)?)*$"
```

in order to match said HSP-specific parameters.

Customers that have set `AC_StartQueryString` must adjust their setting by adding the stated regular expression groups accordingly.

Note that, without adjustment, login requests (containing one of the parameters) may be blocked erroneously.

14.3 HSP_SSLAllowLegacyRenegotiation

The legacy renegotiation support of SSL has been removed so that the directive `HSP_SSLAllowLegacyRenegotiation` is no more available. However, if legacy SSL renegotiation is still needed, mod_ssl's equivalent directive `SSLInsecureRenegotiation` should be used.

A complete documentation of SSLInsecureRenegotiation can be found on the Web Page of Apache HTTPD [1]

14.3.1 Session ID lock rule variable LOGIN_IP

The variable `LOGIN_IP` can be used in Session ID lock rules to specify conditions based on the client's login IP address. The default unset value of this variable has been changed.

Before version 4.15, the variable resolved to `"0.0.0.0"`, if a client has not performed a login so far. Since version 4.15, for unauthenticated sessions, the variable resolves to the empty string ("").

Session ID lock rule directives (`SE_LockRule` or `SE_LockDefineEvent`) using the default unset value of `LOGIN_IP` in expressions must be adjusted accordingly.

For instance, after updating to version 4.15, the following event

```
SE_LockDefineEvent IPhasChangedAfterLogin : (%{LOGIN_IP} != "0.0.0.0") && \
                                                                    (%{LOGIN_IP} != %{ ↩
                                                                       CURRENT_IP})
```

must be changed to

```
SE_LockDefineEvent IPhasChangedAfterLogin : (%{LOGIN_IP} != "") && \
                                                                     (%{LOGIN_IP} != %{CURRENT_IP})
```

in order to still evaluate to true whenever a client has changed its IP after login. In other words: If the client is authenticated (`LOGIN_IP` != ""), and the client's current IP address is different to its login IP address, increase the value of session ID lock rule event variable `IPhasChangedAfterLogin` by one.

14.4 Web Links

More information can be found in the following web resources:

[1] Documentation of the directive `SSLInsecureRenegotiation`

https://httpd.apache.org/docs/current/mod/mod_ssl.html#sslinsecurerenegotiation

15 HSP 4.14

15.1 Summary

This migration guide provides a set of guidelines to migrate the USP Secure Entry Server® from prior version 4.14 to the latest version 4.14.

If you are updating from a Secure Entry Server® Software Suite older than version 4.12, 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.

15.2 Updating to USP Secure Entry Server 4.14

15.2.1 HSP_InitialRequestTimeout

The `HSP_InitialRequestTimeout` directive from the `mod_dos` module has been removed as its functionality is completely covered by `mod_reqtimeout's` directive `RequestReadTimeout`.

The server-wide setting of `HSP_InitialRequestTimeout` in the HSP Listener of following format

```
HSP_InitialRequestTimeout <n>
```

should be be replaced by

```
RequestReadTimeout header=<n>
```

"Important"

As the default setting of `HSP_InitialRequestTimeout` was 3 seconds (n=3), whereas the header timeout of `ReadRequestTimeout` is (at the maximum) 40 seconds, configurations with no customized setting should be completed with

```
RequestReadTimeout header=3
```

in order to achieve same header timeout behavior after the update.

`RequestReadTimeout` additionally provides the possibility to set a body timeout as well as a minimum data rate. A complete documentation of all functionalities can be found on the official Web Page of Apache under https://httpd.apache.org/docs/2.4/mod/mod_reqtimeout.html#requestreadtimeout.s

Please note that non-migrated configurations would lead to start-up errors of the HSP Listener after updating to newest version 4.14.

16 HSP 4.12

16.1 Summary

This migration guide provides a set of guidelines to migrate the USP Secure Entry Server® Software Suite from prior versions 4.11 to the latest version 4.12.

The latest version of USP Secure Entry Server® Software Suite introduces Apache Server 2.4 for the Secure Request Manager (SRM). There have been made changes in many different modules that **could require changes** to your configuration files before using them with the latest version.

We **strongly recommend to carefully study these Apache changes**. We also recommend to have a closer look into the upgrade documentation [1] provided by the Apache community which is presented in the Web Links section below.

With the Apache upgrade, the way how **SSI (Server Side Include) documents** are parsed has changed. Such documents are mainly used for error pages. If this functionality is used, migration is most likely required.

With this new version, USP Secure Entry Server® Software Suite supports authentication enforcement on **web socket connections**. This involves a configuration migration for web socket setups.

The **ModSecurity engine** is delivery now in multiple version including the latest engine version 2.9.1. The configuration need to be changed to select the explicit version.

16.2 Updating to USP Secure Entry Server 4.12

16.2.1 Apache 2.4 changes

The following list of directives have been replaced or removed in Apache 2.4. As they are relevant to the latest USP Secure Entry Server® Software Suite, Linux Version they must be adjusted accordingly for the successful operation.

Changed Modules

Table 2: Changes in directives

Core/Module	Directive	Description / Action required
mod_log_config.c	ErrorLog, TransferLog, CustomLog	Piped logging commands were invoked using `/bin/sh -c` in 2.2 and earlier. In 2.4 and later, piped logging commands are executed directly. To restore the old behavior, use `"|$"` instead of `"|"`. This may be required if you use logrotate in combination with syslog (`/usr/bin/logger`)
various	`Lockfile RewriteLock SSLMutex AcceptMutex SSLStaplingMutex WatchdogMutexPath`	Have been replaced with a single Mutex directive. You will need to evaluate any use of these removed directives in your 2.2 configuration to determine if they can just be deleted or will need to be replaced using Mutex. Recommended setting: `Mutex sysvsem default`
mod_include.c		mod_include has switched to the new `ap_expr` syntax for conditional expressions in `#if` flow control elements used e.g. in SSI files. Use the directive `SSILegacyExprParser` to switch to the old syntax which is compatible with previous Apache version or migrate corresponding server side includes (like error documents) accordingly.
mod_ssl.c	SSLSessionCache	This setting uses a URL alike syntax to define what shared memory implementation to use. If you used a shmht:// setting in here, replace it by shmcb://
	`SSLRequire`	`SSLRequire` is deprecated and should in general be replaced by Require expr. There are some differences described here.
	*_DN environment variables	The format of the *_DN variables has changed following RFC2253 e.g. new comma is used as delimiter. If such variables are used, check for conformity.

New directives

Apache Server 2.4 (in comparison to 2.2) introduced new directives which may be used in the SRM custom commands or in your configuration files:

Table 3: New directives in Apache Server 2.4 and 3rd-party modules

Core/Module	Directive	Remark / Web Resource
core.c	AllowOverrideList UnDefine DefaultRuntimeDir Define Error ErrorLogFormat ExtendedStatus IncludeOptional MaxRangeOverlaps MaxRangeReversals Mutex	
mod_authz_core.c	AuthzSendForbiddenOnFailure AuthMerging	For more information about mod_authz_core, visit http://httpd.apache.org/docs/2.4/mod/-mod_authz_core.html
mod_proxy.c	BalancerGrowth BalancerInherit BalancerPersist ProxyPassInherit ProxySourceAddress ProxyAddHeaders	For more information about mod_proxy, visit http://httpd.apache.org/docs/2.4/mod/-mod_proxy.html
mod_include.c	SSILegacyExprParser	For more information about mod_include, visit http://httpd.apache.org/docs/2.4/mod/-mod_include.html
mod_ssl.c	SSLCARevocationCheck SSLOCSPDefaultResponder SSLOCSPEnable SSLOCSPOverrideResponder SSLOCSPResponderTimeout SSLOCSPResponseMaxAge SSLOCSPResponseTimeSkew SSLOCSPUseRequestNonce SSLOpenSSLConfCmd SSLProxyCARevocationCheck SSLProxyCheckPeerName SSLSRPUnknownUserSeed SSLSRPVerifierFile SSLSessionTicketKeyFile SSLStaplingCache SSLStaplingErrorCacheTimeout SSLStaplingFakeTryLater SSLStaplingForceURL SSLStaplingResponderTimeout SSLStaplingResponseMaxAge SSLStaplingResponseTimeSkew SSLStaplingReturnResponderErrors SSLStaplingStandardCacheTimeout SSLUseStapling	For SSLOpenSSLConfCmd, at least OpenSSL 1.0.2 is required (which is available in HSP 4.8.0.0 and later) For more information about mod_ssl, visit http://httpd.apache.org/docs/2.4/mod/-mod_ssl.html
mod_setenvif.c	SetEnvIfExpr	For more information about mod_setenvif, visit http://httpd.apache.org/docs/2.4/mod/-mod_setenvif.html

Table 3: (continued)

Core/Module	Directive	Remark / Web Resource
mod_qos.c	QS_ClientEventBlockExcludeIP QS_CondClientEventLimitCount QS_RedirectIf QS_SupportIPv6	3rd-party module For more information about mod_qos, visit http://opensource.adnovum.ch/mod_qos/
mod_setenvifplus.c	AddOutputFilterPlus CookieEncPlus SetStatusPlus	3rd-party module For more information about mod_setenvifplus, visit http://opensource.adnovum.ch/-mod_setenvifplus/

New Modules

Below is a list of new modules which have been introduced in the latest USP Secure Entry Server® Software Suite, Linux Version due to the update to Apache 2.4.

Table 4: New modules introduced with USP Secure Entry Server® Software Suite, Linux Version

Module	Description	Web Resource
`mod_proxy_html.c`	Formerly a third-party module, this supports fixing of HTML links in a reverse proxy situation, where the backend generates URLs that are not valid for the proxy's clients.	http://httpd.apache.org/docs/2.4/mod/-mod_proxy_html.html

`mod_proxy_wstunnel.c`
`mod_proxy_balancer.c`
`mod_lbmethod_byrequests.c`
`mod_lbmethod_bytraffic.c`
`mod_lbmethod_bybusyness.c`
	Modules providing support for tunneling web socket connections to a backend web socket server and load balancing.	http://httpd.apache.org/docs/2.4/mod/-mod_proxy_wstunnel.html
		http://httpd.apache.org/docs/2.4/mod/-mod_proxy_balancer.html

`mod_slotmem_shm.c`	`mod_slotmem_shm` is a memory provider for creation and access to a shared memory segment in which the datasets are organized in "slots."	http://httpd.apache.org/docs/2.4/mod/-mod_slotmem_shm.html
`mod_xml2enc.c`	Formerly a third-party module, this supports internationalizations in libxml2-based (markup-aware) filter modules.	http://httpd.apache.org/docs/2.4/mod/-mod_xml2enc.html

Removed directives

Some directives are no longer supported in Apache 2.4 and therefore have been removed i.e. are no longer available in USP Secure Entry Server® Software Suite, Linux Version. These directives require special attention as the HSP processes will not start up unless the directives have been manually replaced by their equivalents (if any are available) or have been manually removed entirely from the configuration.

Table 5: Removed directives from USP Secure Entry Server® Software Suite, Linux Version

Module	Directive	Equivalent or action required
`mod_rewrite.c`	`RewriteLog` `RewriteLogLevel`	This functionality has been completely replaced by the new per-module logging configuration using the `LogLevel` directive. Example: `LogLevel info rewrite:trace3` See also: [3]

"Server Side Include (SSI) in Error Documents"

If SSI is used in the SRM Error Documents, usually indicated by the file extensions ".shtml", it might require manual changes.

The syntax for flow control `#if has changed.` The condition expression is using now ap_expr syntax [4] with some differences described below. Old conditions are might evaluated differently after the update and can show the wrong content. The Error Log will show a corresponding error message, that the expression couldn't be parsed.

USP Secure Entry Server® Software Suite customer can enable the legacy expression syntax by setting the directive `SSILegacyExprParser on` in the Error Documents locations.

Note: The new condition expression will not work anymore if this directive is set.

Migration of existing SSI documents

When using the ap_expr syntax in SSI documents the variable references is not documented correctly. Should a variable be evaluated the syntax is `v("<VARIABLE_NAME>")` and not as documented `%{<VARIABLE_NAME>}`.

```
<!--#if expr='v("REDIRECT_ERROR_NOTES") =~ /^HSP150.*/'-->
```

Example if clause with variable and regex match with the new syntax.

Web socket tunnels

Since version 4.10.0.0, USP Secure Entry Server® supports tunneling of web socket connections by use of the service of mod_proxy and its additional web socket module mod_proxy_wstunnel. Latter has been introduced in Apache 2.4 and thus – in former version of USP Secure Entry Server® – was only available in HTTP/HTTPS Listener but not in Secure Request Manager. For that reason, until the latest version the Secure Gateway module (mod_sgw) has been used in order to tunnel web socket connections through SRM.

Since the update to Apache 2.4, the mod_proxy_wstunnel service can also be used in SRM and customers should find in the following the needed configuration changes to migrate the web socket support from mod_sgw to mod_proxy.

In the Secure Request Manager, the migration can be made in two different ways.

Either the corresponding SGW virtual hosts remain in use by migrating them to mod_proxy or these virtual hosts are removed and the configuration is moved to another (already existing) virtual host by adding a new location. The latter is advised, but note, this also leads to adjustments in the HTTP/HTTPS Listener because of changes of port numbers and of end-point URLs.

Further information to the integration of web socket back-ends including how to enable WSS to those, can be found in the HSP Administration guide.

Add web socket location to an existing virtual host

If the tunneling of web socket connections should be moved to an existing virtual host, following steps are necessary:

- The SRM SGW virtual host with the web socket integration should be removed completely.
- Add web socket location with the mod_proxy settings within the scope of an appropriate virtual host.

```
<Location ##SRM-LOCATION_NAME##>
   SetHandler default
   HGW_Disable

   ProxyPass ws://##NAME-OR-IP-TARGET-WS-SERVER##:##PORT-OF-TARGET-WS-SERVER##/##PATH-TO- ↩
       SERVICE-TRANSLATED##
</Location>
```

New Secure Request Manager configuration example of a web socket application in the latest version of USP Secure Entry Server®

- In the HTTP/HTTPS Listener the port number of the virtual host has to be changed within the ProxyPass directive to the corresponding SRM virtual host port.
- In the HTTP/HTTPS Listener the location path of the SRM must be added.

```
<Location ##LOCATION-NAME## >
   SetHandler default
   HGW_Disable

   ProxyPass ws://127.0.0.1:##PORT-OF-SRM##/##SRM-LOCATION-NAME##
</Location>
```

Migrated HTTP/HTTPS Listener configuration example of a web socket application in the latest version of USP Secure Entry Server®

Migration by keeping existing web socket virtual host

All SGW directives should be removed. All other directives (as for instance ErrorLog) can stay in effect. As mod_proxy allows to integrate a web socket back-end server at location level, a Location is added. In the new location the directive ProxyPass is used to state the IP (or name) and port of the web socket target server. In addition, the settings "SetHandler default" and HGW_Disable must be set.

```
Listen 127.0.0.1:##PORT##

<VirtualHost 127.0.0.1:##PORT##>
  ErrorLog ##params##

  SGW_Host ##NAME-OR-IP-TARGET-WS-SERVER##:##PORT-OF-TARGET-WS-SERVER##
  SGW_ConnectTimeOut ##param##
  SGW_FinalTimeOut ##param##
  SGW_TimeOut ##param##
</VirtualHost>
```

Obsolete SRM integration example of a web socket applications in prior version of USP Secure Entry Server®

```
Listen 127.0.0.1:##PORT##

<VirtualHost 127.0.0.1:##PORT##>
   ErrorLog ##params##
```

```
   <Location />
      SetHandler default
      HGW_Disable
      ProxyPass ws://##NAME-OR-IP-TARGET-WS-SERVER##:##PORT-OF-TARGET-WS-SERVER##/
   </Location>
</VirtualHost>
```

Migrated SRM configuration example of a web socket application in the latest version of USP Secure Entry Server®

Migrating web socket setups with multiple back-ends

Should there be several back-ends be defined in the SGW_Host directive, the configuration must be migrated as such:

- The list of load-balanced backends must be added in virtual host scope

```
<Proxy balancer://##BALANCER-REFERENCE##>
  BalancerMember ws://##TARGET-WS-SERVER-1##:##PORT-OF-TARGET-WS-SERVER-1## loadfactor=1
  BalancerMember ws://##TARGET-WS-SERVER-2##:##PORT-OF-TARGET-WS-SERVER-2## loadfactor=1
  ProxySet lbmethod=byrequests
</Proxy>
```

New load-balanced web socket backend list. For further information about load-balancing method see the documentation.

- Set the reference to the load-balance group in the ProxyPass directive

```
<Location ##SRM-LOCATION_NAME##>
   SetHandler default
   HGW_Disable

   ProxyPass balancer://##BALANCER-REFERENCE##/##PATH-TO-SERVICE-TRANSLATED##
</Location>
```

Web socket location using load-balanced backend reference.

Multiversion ModSecurity engine

ModSecurity is new delivered in multiple version. With this release 4.12 the newest engine version 2.9.1 and the previous delivered version 2.8.0 is include. Due this new delivery the configuration needs to be adjusted.

The modul library name contains new the version number, therefore the LoadModule directive need to be adjusted:

```
LoadModule security2_module libexec/mod_security-2.9.1.so
```

16.3 Web Links

More information about upgrading Apache Server to 2.4 can be found at the following web resources:

[1] Upgrading to 2.4 from 2.2: http://httpd.apache.org/docs/2.4/upgrading.html [2] Run-Time Configuration Changes: http://httpd.apache.org/docs/2.4/upgrading.html#run-time [3] mod_rewrite: Logging: http://httpd.apache.org/docs/2.4/mod/mod_rewrite.html#logging [4] SSI: new condition syntax: http://httpd.apache.org/docs/current/expr.html

17 HSP 4.11

17.1 Summary

This migration guide provides a set of guidelines to migrate the USP Secure Entry Server® from prior version 4.14 to the latest version 4.11.

The Web Application Firewall (HSP) removed support for the SSLv2 protocol. Additionally the custom command directive `SSLSessionClientTicket` was replaced by the directive `SSLSessionTickets`. Please see Section 17.2 below for more details on how to handle existing configurations.

If you are updating from a Secure Entry Server® Software Suite older than version 4.10, 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.

17.2 Updating to USP Secure Entry Server 4.11

The Web Application Firewall (HSP) ceased support for the SSLv2 protocol on backend connections. The section 2.1 below contains more details about this change. The custom command directive SSLSessionClientTicket was replaced by the directive SSLSessionTickets, which is described in more detail in Section 17.2.2 below.

17.2.1 Removed support for SSLv2 on Backend Connections

As announced in the SES security bulletin SES-2016-0002 released by USP on 4th of March 2016, the Web Application Firewall (HSP) removed support for the SSLv2 protocol on connections to backend servers. This affects the directives `HGW_SslClientProtocol` and `HGW_SslClientProtocolDefault`, which don't accept the parameter SSLv2 anymore.

Existing configurations that are still explicitly configured to use the SSLv2 protocol on backend connections must be updated.

17.2.2 Directive `SSLSessionClientTicket` replaced by `SSLSessionTickets`

The directive `SSLSessionClientTicket` was replaced by the directive `SSLSessionTickets`. This change only affects the directive name, the provided functionality is exactly the same.

On update, the directive `SSLSessionClientTicket` must be replaced by `SSLSessionTickets` without changing its parameter value.