

The SPOCP protocol

<!-- --> <!-- -->

Table of contents

1 The SPOCP protocol.....	2
1.1 Content.....	2
1.2 0. Introduction.....	2
1.3 1. Basic operation.....	2
1.4 1.1 Input grammar.....	3
1.5 1.2 Output grammar.....	4
1.6	4
1.7 2. Client commands.....	4

1. The SPOCP protocol

1.1. Content

- 0. Introduction
- 1. [Basic operation](#)
 - 1.1 [Input grammar](#)
 - 1.2 [Output grammar](#)
 - 1.3 [Response codes](#)
- 2. [Client Commands](#)
 - 2.1 [STARTTLS](#)
 - 2.2 [QUERY](#)
 - 2.3 [ADD](#)
 - 2.4 [CAPA](#)
 - 2.5 [DELETE](#)
 - 2.6 [LIST](#)
 - 2.7 [LOGOUT](#)
 - 2.8 [BEGIN](#)
 - 2.9 [COMMIT](#)
 - 2.10 [ROLLBACK](#)
 - 2.11 [SUBJECT](#)
 - 2.12 [SUMMARY](#)
 - 2.13 [AUTH](#)
 - 2.14 [SHOW](#)
- 3. [Example](#)

1.2. 0. Introduction

1.3. 1. Basic operation

An SPOCP connection consists of the establishment of a client/server network connection, and subsequent client/server interactions. These client/server interactions consist of a client command and a server result response.

All interactions transmitted by client and server are in the form of sequences of bytes in a format described below.

Commands in SPOCP consists of a keyword, possibly followed by one or more argument. Keywords consists of printable ASCII characters.

Server data are only sent as a result of a client command.

The SPOCP protocol

The server completion result response indicates the success or failure of the operation.

Every Spocp reply consists of a three digit number possibly followed by some text. More about that [below](#).

The traffic between a client and a server can be protected by the use of TLS/SSL. *STARTTLS* is used to upgrad the connection from unprotected to protected state.

The Spocp protocol are almost correct S-expression, the thing making them not correct is the usage of a leading length definition and the abandonment of the outer parentheses. The leading length definition is there to let the server and the client know how much data to expect.

1.4. 1.1 Input grammar

1.4.1. !!! The Spocp server is working in case exact mode !!!

spocp-command = length ":" command ; the length value represents the length of the command in number of bytes
command = starttls / query / add / delete / list / aci
logout / begin / commit / rollback / subjectcom
starttls = "8:STARTTLS" query = "5:QUERY" [l-path] l-s-expr
add = "3:ADD" [l-path] l-s-expr [return-info] delete = "6:DELETE" [l-path] length ":" ruleid
list = "4:LIST" [l-path] *(length ":" "+" / "-" s-expr)
aci = "3:ACI" [l-path] length ":" aci-expr
logout = "6:LOGOUT" begin = "5:BEGIN" commit = "6:COMMIT" rollback = "8:ROLLBACK" subjectcom = "7:SUBJECT" [l-s-exp]
aci-expr = "(3:aci(8:resource [s-expr])(6:action [aci-ops-or])(7:subject [s-expr]))"
aci-ops = ("3:ADD" / "6:DELETE" / "4:LIST")
aci-ops-or = aci-ops / "(2:or" 2*3aci-ops)"
; the or construct only with reasonable combinations of aci-ops ; (2:or3:ADD3:ADD) is for instance not reasonable
l-s-expr = length ":" s-expr s-expr = "(" bytestring *s-part ")"
s-part = bytestring / s-expr / starforms
return-info = bytestring bytestring = length ":" 1*bytes ; The number of bytes in the bytestring must be equal to the ;
length specification length = nzdigit *digit
nzdigit = %x31-39 digit = "0" / nzdigit
byte = %x00-FF starforms = "(1:*" (range / prefix / suffix / or / bcond))"
or = "2:or" 1*s-expr range = "5:range" rangespec prefix = "6:prefix" bytestring suffix = "6:suffix" bytestring
bcond = "5:bcond" 1*extref rangespec = alpha / numeric / date / time / ipv4 / ipv6
alpha = "5:alpha" [lole utf8string [goge utf8string]] / [goge utf8string [lole utf8string]]
numeric = "7:numeric" [lole number [goge number]] / [goge number [lole number]]
date = "4:date" [goge dat [lole dat]] / [lole dat [goge dat]]
time = "4:time" [lole hms [goge hms]] / [goge hms [lole hms]]
ipv4 = "4:ipv4" [lole ip4num [goge ip4num]] / [goge ip4num [lole ip4num]]
ipv6 = "4:ipv6" [lole ip6num [goge ip6num]] / [goge ip6num [lole ip6num]]
lole = "1:l" / "2:le" goge = "1:g" / "2:ge"
number = length ":" 1*digit dat = length ":" date hms = length ":" timeofday
ip4num = length ":" IPv4addr ip6num = length ":" IPv6addr utf8string = length ":" 1*UTF8
bytestring = length ":" 1*byte ruleid = length ":" 1*(lc / digit)
date = nzdigit 3digit "-" month "-" day "-" timeofday timeofday = hour ":" sixty ":" sixty month / "0"nzdigit) / "1" ("0" / "1" / "2") day = (("0" / "1" / "2") digit) / "3" ("0" / "1")

```
hour = ( ("0" / "1" ) digit ) / "2" %x30-34 sixty = %x30-35 %x30-39 UTF8 = %x01-09 /
%x0B-0C / %x0E-7F / UTF8-2 / UTF8-3 / UTF8-4 / UTF8-5 / UTF8-6 UTF8-1 =
%x80-BF UTF8-2 = %xC0-DF UTF8-1 UTF8-3 = %xE0-EF 2UTF8-1 UTF8-4 =
%xF0-F7 3UTF8-1 UTF8-5 = %xF8-FB 4UTF8-1 UTF8-6 = %xFC-FD 5UTF8-1
extref = length ":" type ":" typespecific type = 1*lc lc = %x61-7A typespecific = *UTF8
l-path = length ":" path path = '/' [ 1*pathpart ] pathpart = 1*dirchar '/' dirchar =
%x30-39 / %x41-5A / %x61-7A / '-' / '_' ; ipv6 ABNF from RFC 2373 IPv6addr =
hexpart [ ":" IPv4addr ] IPv4addr = 1*3DIGIT "." 1*3DIGIT "." 1*3DIGIT
IPv6prefix = hexpart "/" 1*2DIGIT hexpart = hexseq | hexseq ":" [ hexseq ] | "::" [
hexseq ] hexseq = hex4 *( ":" hex4 ) hex4 = 1*4HEXDIG
```

1.5. 1.2 Output grammar

The number is intended for use by automata to determine what state to enter next; the text, except in the case of multiline responses, is meant for the human user. The intention is that the three digits should contain enough encoded information that the SPOCP-client need not examine the text to find out what has happened.

```
spocp-response = *(multiline) singleline multiline = length ":" multi-response
multi-response = "3:201" m-info singleline = length ":" response response = "3:"
replycode [ bytestring ] ; replycode 201 can not appear in singleline responses
replycode = ( "2" / "3" / "4" / "5" ) digit digit [ bytestring ]
```

In some cases a shorthand for the rule can be used, a ID. This ID is presently the SHA1 hash of the rule itself. Possible return-info connected to the rule is not included when the hash is calculated.

There are only two cases where multiline responses can appear, one is if the client poses a query and the server wants to pass back som return-info together with the positive answer. Then that return-info will appear in a separate line before the line containing the 200 code. The other case is when the clients performs a list operation and there are rules matching the list criteria. each rule will then be returned on a separate line.

The ABNF for the two cases are

```
m-info = return-info / list-info list-info = l-path ruleid l-s-expr [ return-info ]
```

1.6.

1.7. 2. Client commands

1.7.1. 2.1 STARTTLS command

Usage:

The SPOCP protocol

Starttls is used when the connection between the client and the server must be protected by SSL or TLS. The client must identify itself using a certificate, it is not only the server that has to have its certificate verified

Arguments:

As of now, none

Result:

9:3:2002:Ok - starttls recognized and TLS/SSL negotiation will start

20:3:50113:Not supported - Starttls not implemented in this server

20:3:50320Already in operation - TLS/SSL already in operation

The client is also expected to authenticate. If the authentication fails the connection is immediately closed. Downgrading to not use TLS/SSL is not possible,

Any commands sent by the client after the server has received this command and before the TLS/SSL negotiation is completed will be silently ignored.

The client as well as the server MUST check its understanding of the opponent's hostname against the its identity as presented in the opponent's Certificate message, in order to prevent man-in-the-middle attacks.

Matching from the clients point of view is performed according to these rules:

- The client MUST use the server hostname it used to open the SPOCP connection as the value to compare against the server name as expressed in the server's certificate. The client MUST NOT use the server's canonical DNS name or any other derived form of name.
- If a subjectAltName extension of type dNSName is present in the certificate, it SHOULD be used as the source of the server's identity. If more than one identity of a given type is present in the certificate (e.g. more than one dNSName name), a match in any one of the set is considered acceptable.
- Matching is case-insensitive.

If the hostname does not match the dNSName-based identity in the certificate per the above check, user-oriented clients SHOULD either notify the user (clients MAY give the user the opportunity to continue with the connection in any case) or terminate the connection and indicate that the server's identity is suspect.

Automated clients SHOULD close the connection, returning and/or logging an error indicating that the server's identity is suspect.

Beyond the server identity checks described in this section, clients SHOULD be prepared to do further checking to ensure that the server is authorized to provide the service it is observed to provide. The client MAY need to make use of local policy information.

Example:

C: 10:8:STARTTLS

S: 9:3:2002:OK

1.7.2. 2.2 QUERY command

Arguments:

One S-expression representing a authorisation query

Possible Results:

9:3:2002:OK - The given S-expression matches some rule stored in the SPOCP server

13:3:2026:Denied - No rule matched

22:3:50012:Syntax error - If the S-expression is not a wellformed S-expression

If a rule in the server matched the given S-expression and there is return information coupled to that rule, the return information will be returned together with the positive response. If there are more than one rule that matches only one of them will be used to construct the response. Under these conditions sending several queries with the same S-expression in a sequence might not necessarily result in the return of the same answer to each one of them. That is you can not assume that there exists any well defined and consistent order between the rules.

Overlapping rules should therefore if possible be avoided, especially if there are return information coupled to them.

Example:

C:

91:5:QUERY81:(5:spocp(8:resource(4:file3:etc6:passwd))(6:action4:read)(7:subject(3:uid

S: 37:3:20129:ftp://ftp.spocp.org/cert0.pem

S: 9:3:2002:OK

1.7.3. 2.3 ADD command

Arguments:

S-expression representing a rule to be stored, possibly accompanied by some return information. See the ABNF grammar defined in [input](#) section.

Possible Result:

9:3:2002:OK - The rule was successfully stored in the rule database

11:3:202:6:Denied - Adding was not permitted

20:3:50012:Syntax error - Error in the S-expression syntax

Example:

C:

88:3:ADD80:(5:spocp(8:resource(4:file3:etc6:groups))(6:action4:read)(7:subject(3:uid3:10

S: 9:3:2002:OK

1.7.4. 2.4 ACI command

Usage:

Controlling the access to the ruledatabase. Format: aci = "3:aci" length ":"
ras-desc ras-desc = "(3:aci(8:resource" s-expr *args ")(6:action" action
")(7:subject" subject ")))" action = length ":" ("ADD" / "DELETE" / "LIST" / "ACI")
args = bytestring subject = s-expr

Arguments:

Only one argument and it is a Access control rule for some set of rules in the database

Results:

Example:

C:

107:3:ACI99:(3:aci(8:resource(5:spocp(8:resource3:etc)(6:action4:read)(7:subject)))(6:act
3:2002:Ok - The rule/rules was successfully added

1.7.5. 2.5 DELETE command

Arguments:

Index of the rule/rules to be deleted from the rule database

Result:

3:2002:Ok - The rule/rules was successfully deleted

3:50012:Syntax error

3:50418:Too many arguments

3:50510:Unknown ID - The rule to be deleted was not in the rule database.

Example:

C: 94:6:DELETE40:a51b9a500c15d773e7e504f7344d89790bb9dd1b

40:a51b9a500c15d773e7e504f7344d89790bb9dd1c

S: 26:3:50418:Too many arguments

C: 94:6:DELETE45:/bar/a51b9a500c15d773e7e504f7344d89790bb9dd1b

3:2002:Ok

1.7.6. 2.6 LIST command

Arguments:

List of subelements

Result:

3:2002:Ok - zero or more rules matched the subelement specification

3:50012:Syntax Error - syntax error in subelement specification

The less permissive comparison, as specified in [Introduction to S-expressions as used by](#)

[SPOCP](#), is used here as well as when evaluating the query command. The difference is that you are able to define different directions for the comparison with this command. If you use '+' as order specifier the order is the same as used in the query command, that is you search for a rule that are more permissive than the S-expression specified in the query. If on the other hand you specify the '-' as order specified, you will look for rules that are less permissive than the specification. The syntax for this command allows you to specify different directions for each subelement, hence you can specify:

```
LIST +spocp -(8:resource) +(6:action4:read) -(7:subject(3:uid))
```

Which then will match rules like:

```
(5:spocp(8:resource(4:file3:etc6:groups))(6:action4:read)(7:subject(3:uid3:100)))  
(5:spocp(8:resource(4:file3:etc6:passwd))(6:action4:read)(7:subject(3:uid3:100)))
```

Since every rule in the SPOCP server MUST be a list S-expression, the subelements that we are talking about here are the first level elements of that list.

Note that since the less permissive comparison is used, you can not easily list all rules that are valid for ranges of values. If you for instance have subelements in rules that look like this:

```
(age (* range numeric le 6))  
(age (* range numeric ge 7 le 18))  
(age (* range numeric gt 18 le 40))  
(age (* range numeric ge 41 lt 65))  
(age (* range numeric ge 65))
```

Then "LIST +3:age -(1:*5:range2:le2:10)" will match rule 1 but not rule 2. One of two partly overlapping ranges can never be regarded as less permissive than the other. If you really want to know every rule that concerns entities whos age are less than or equal to 10, then you have to combine the result of two queries: the one specified above and "LIST +3:age +2:10".

Note when searching for and listing rules that contain boundary conditions. One might want the SPOCP server to disregard some types of such conditions. For instance time constrains can lead to akward constraints as to when testing can be done. Therefore one would like to have the possibily to specify a number of boundary conditions that are disregarded while searching/listing. How this is done is not decided on yet.

Example:(The "201" lines wrapped to make them more readable.)

```
C: 45:4:LIST8:+5:spocp26:-(8:resource(4:file3:etc))
```

```
S: 132:3:2011:/40:a51b9a500c15d773e7e504f7344d89790bb9dd1b
```

```
81:(5:spocp(8:resource(4:file3:etc6:groups))(6:action4:read)(7:subject(3:uid3:100)))
```

```
S: 132:3:2011:/40:d361bd84e7e0deaa56bffa229c61813c59161eef
```

```
81:(5:spocp(8:resource(4:file3:etc6:passwd))(6:action4:read)(7:subject(3:uid3:100)))
```

S: 9:3:2002:Ok

1.7.7. 2.7 LOGOUT command

Arguments:

none

Result:

10:3:2033:Bye -No other response should be expected

Example:

C: 8:6:LOGOUT

S: 10:3:2033:Bye

1.7.8. 2.8 BEGIN command

Arguments:

None

Results:

Example:

1.7.9. 2.9 COMMIT command

Arguments:

None

Results:

Example:

1.7.10. 2.10 ROLLBACK command

Arguments:

None

Results:

Example:

1.7.11. 2.11 SUBJECT command

Arguments:

S-expression representing the subject on which behalf the Spocp client works

Results:

Example:

1.7.12. 3. EXAMPLE

Adding a rule that allows Eva and Roland to 'read' any resource within the picture gallery

C: 64:3:ADD56:(2:pg(3:res)(3:act4:read)(4:subj(1:*2:or3:eva6:roland)))

S: 9:3:2002:Ok

Adding a rule that allows Jeanne to 'read' any picture within the 'turkiet' part of the '2003' part of the picture gallery

C: 65:3:ADD57:(2:pg(3:res4:20037:turkiet)(3:act4:read)(4:subj6:jeanne))

S: 9:3:2002:Ok

Adding a rule that allows Hanne to also view any picture within the 'turkiet' part of the '2003' part of the gallery

C: 65:3:ADD57:(2:pg(3:res4:20037:turkiet)(3:act4:read)(4:subj5:hanne))

S: 9:3:2002:Ok

Jeanne tries to view one of the pictures in the 'turkiet' part of the '2003' part of the gallery. And she is give the permission to do so

C:

82:5:QUERY72:(2:pg(3:res4:20037:turkiet12:dscf0404.jpg)(3:act4:read)(4:subj6:jeanne))

S: 9:3:2002:Ok

Someone, unspecified who, tries to list all the rules. Since no access control on the rules is in place, every rule is listed

C: 6:4:LIST

S:

113:3:201104:40:06caa09539aa0aa59652c9c9e3df3eb46153310b58:/(2:pg(3:res4:20037:turki

S:

112:3:201103:40:694b21327916616ca5a4c08350499472289beb8057:/(2:pg(3:res4:20037:turk

S:

112:3:201103:40:fab37dfe994e15e2f4f7381c0bb4dfd0834bb0b57:/(2:pg(3:res)(3:act4:read)(4

S: 9:3:2002:Ok

Someone, unspecified who, wants to delete a rule. Since the rule exists and no access control on the rules are in place the request is granted

C: 53:6:DELETE40:06caa09539aa0aa59652c9c9e3df3eb46153310b57

S: 9:3:2002:Ok

Someone lists the rules

C: 6:4:LIST

S:

112:3:201103:40:694b21327916616ca5a4c08350499472289beb8057:/(2:pg(3:res4:20037:turk

S:

112:3:201103:40:fab37dfe994e15e2f4f7381c0bb4dfd0834bb0b57:/(2:pg(3:res)(3:act4:read)(4

S: 9:3:2002:Ok

Jeanne tries to look at the picture again, but now as the rule allowing her to watch it has been removed she is denied

C:

82:5:QUERY72:(2:pg(3:res4:20037:turkiet12:dscf0404.jpg)(3:act4:read)(4:subj6:jeanne))

S: 13:3:2026:Denied

Eva wants to look at a picture, and since she can do just about anything on the gallery she's allowed

C:

78:5:QUERY68:(2:pg(3:res4:20036:sommar12:dscf0668.jpg))(3:act4:read)(4:subj3:eva))

S: 9:3:2002:Ok

Access control on the rules are put into place, starting of by given Roland the right to add any type of access control

C: 70:3:ACI62:(3:aci(8:resource(3:aci)))(6:action)(7:subject(3:uid6:roland))

S: 9:3:2002:Ok

The following operations on Spocp are to be performed in the name of Roland

C: 27:7:SUBJECT15:(3:uid6:roland)

S: 9:3:2002:Ok

Roland gives everyone the right to read the rules governing access to the picture gallery

C: 60:3:ACI52:(3:aci(8:resource(2:pg)))(6:action4:LIST)(7:subject))

S: 9:3:2002:Ok

He gives Eva the right to add any type of access control rules

C: 67:3:ACI59:(3:aci(8:resource(3:aci)))(6:action)(7:subject(3:uid3:eva))

S: 9:3:2002:Ok

He gives Eva the right to add new access rules to the picture gallery

C: 66:3:ACI58:(3:aci(8:resource(2:pg)))(6:action)(7:subject(3:uid3:eva))

S: 9:3:2002:Ok

He gives himself the right to add new access rules to the picture gallery

C: 69:3:ACI61:(3:aci(8:resource(2:pg)))(6:action)(7:subject(3:uid6:roland))

S: 9:3:2002:Ok

And finally he lists all the rules to make certain it is as he wanted it to be

C: 6:4:LIST

S:

112:3:201103:40:694b21327916616ca5a4c08350499472289beb8057://(2:pg(3:res4:20037:turk

S:

112:3:201103:40:fab37dfe994e15e2f4f7381c0bb4dfd0834bb0b57://(2:pg(3:res)(3:act4:read)(4

S:

114:3:201105:40:280fe84080388f90a9fb59e529a683805fe2db0359://(3:aci(8:resource(2:pg)))(6

S:

117:3:201108:40:2bab848e4bbbd23acf5c22b0c21c824128921aa462://(3:aci(8:resource(2:pg)))(6

S:

118:3:201109:40:869cf055685e0ecec42d93e8f1ca36ef9fec1c5f63://(3:aci(8:resource(3:aci)))(6:a

S: 9:3:2002:Ok

Roland 'logs out'

C: 9:7:SUBJECT

S: 9:3:2002:Ok

Someone, lists the rules. This person only sees what is explicitly allowed which is the picture gallery rules

C: 6:4:LIST

S:

112:3:201103:40:694b21327916616ca5a4c08350499472289beb8057:/(2:pg(3:res4:20037:turk

S:

112:3:201103:40:fab37dfe994e15e2f4f7381c0bb4dfd0834bb0b57:/(2:pg(3:res)(3:act4:read)(4

S: 9:3:2002:Ok

Eva 'logs in'

C: 24:7:SUBJECT12:(3:uid3:eva)

S: 9:3:2002:Ok

Eva adds a rule giving Jeanne the permission to see the pictures within the 'sommer' of the '2003' part of the gallery

C: 64:3:ADD56:(2:pg(3:res4:20036:sommar)(3:act4:read)(4:subj6:jeanne))

S: 9:3:2002:Ok

Eva 'logs out'

C: 9:7:SUBJECT

S: 9:3:2002:Ok

Jeanne checks to see if she can view a specific 'sommer' '2003' picture and she can

C:

81:5:QUERY71:(2:pg(3:res4:20036:sommar12:dscf0668.jpg)(3:act4:read)(4:subj6:jeanne))

S: 9:3:2002:Ok