

Simple Policy Control Protocol

```
<!-- --> <!-- --> <!-- --> <!-- --> body { font-family: verdana, charcoal, helvetica, arial,
sans-serif; font-size: small ; color: #000000 ; background-color: #ffffff ; } .title { color:
#990000; font-size: x-large ; font-weight: bold; text-align: right; font-family: helvetica,
monaco, "MS Sans Serif", arial, sans-serif; background-color: transparent; }
.filename { color: #666666; font-size: 18px; line-height: 28px; font-weight: bold;
text-align: right; font-family: helvetica, arial, sans-serif; background-color:
transparent; } td.rfcbug { background-color: #000000 ; width: 30px ; height: 30px ;
text-align: justify; vertical-align: middle ; padding-top: 2px ; } td.rfcbug span.RFC {
color: #666666; font-weight: bold; text-decoration: none; background-color: #000000
; font-family: monaco, charcoal, geneva, "MS Sans Serif", helvetica, verdana,
sans-serif; font-size: x-small ; } td.rfcbug span.hotText { color: #ffffff; font-weight:
normal; text-decoration: none; text-align: center ; font-family: charcoal, monaco,
geneva, "MS Sans Serif", helvetica, verdana, sans-serif; font-size: x-small ;
background-color: #000000; } A { font-weight: bold; } A:link { color: #990000;
background-color: transparent ; } A:visited { color: #333333; background-color:
transparent ; } A:active { color: #333333; background-color: transparent ; } p {
margin-left: 2em; margin-right: 2em; } p.copyright { font-size: x-small ; } p.toc {
font-size: small ; font-weight: bold ; margin-left: 3em ; } span.emph { font-style: italic; }
span.strong { font-weight: bold; } span.verb { font-family: "Courier New", Courier,
monospace ; } ol.text { margin-left: 2em; margin-right: 2em; } ul.text { margin-left:
2em; margin-right: 2em; } li { margin-left: 3em; } pre { margin-left: 3em; color:
#333333; background-color: transparent; font-family: "Courier New", Courier,
monospace ; font-size: small ; } h3 { color: #333333; font-size: medium ; font-family:
helvetica, arial, sans-serif ; background-color: transparent; } h4 { font-size: small;
font-family: helvetica, arial, sans-serif ; } table.bug { width: 30px ; height: 15px ; }
td.bug { color: #ffffff ; background-color: #990000 ; text-align: center ; width: 30px ;
height: 15px ; } td.bug A.link2 { color: #ffffff ; font-weight: bold; text-decoration: none;
font-family: monaco, charcoal, geneva, "MS Sans Serif", helvetica, sans-serif;
font-size: x-small ; background-color: transparent } td.header { color: #ffffff; font-size:
x-small ; font-family: arial, helvetica, sans-serif; vertical-align: top; background-color:
#666666 ; width: 33% ; } td.author { font-weight: bold; margin-left: 4em; font-size:
x-small ; } td.author-text { font-size: x-small; } table.data { vertical-align: top ;
border-collapse: collapse ; border-style: solid solid solid solid ; border-color: black
black black black ; font-size: small ; text-align: center ; } table.data th { font-weight:
bold ; border-style: solid solid solid solid ; border-color: black black black black ; }
table.data td { border-style: solid solid solid solid ; border-color: #333333 #333333
#333333 #333333 ; } hr { height: 1px ; } -->
```

Table of contents

13

Simple Policy Control Protocol

1.

TOC	
Network Working Group	R. Hedberg
Internet-Draft	Stockholm university
Expires: July 1, 2004	January 2004

Simple Policy Control Protocol

draft-hedberg-spocp-00

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on July 1, 2004.

Copyright Notice

Copyright (C) The Internet Society (2004). All Rights Reserved.

Table of Contents

- [1.](#) Abstract
- [2.](#) Introduction
- [3.](#) The SPOCP model
- [4.](#) SPOCP functions
 - [4.1](#) Input grammar
 - [4.2](#) Output grammar
 - [4.3](#) QUERY
 - [4.4](#) STARTTLS

- [4.5](#) ADD
- [4.6](#) DELETE
- [4.7](#) LIST
- [4.8](#) LOGOUT
- [4.9](#) BEGIN
- [4.10](#) COMMIT
- [4.11](#) ROLLBACK
- [4.12](#) SUBJECT
- [4.13](#) AUTH
- [4.14](#) CAPABILITY
- [4.15](#) BCOND
- [5.](#) The theory behind return-info
- [6.](#) Security considerations
- [§](#) References
- [§](#) Author's Address
- [A.](#) SPOCP Reply Codes
- [§](#) Intellectual Property and Copyright Statements

[TOC](#)

1. Abstract

[TOC](#)

2. Introduction

The objective of SPOCP is to support the use of a generalized access control service (as defined in [\[RFC2828\]](#)) by a multitude of applications. The protocol not only supports querying for access control decisions but also for administration of security policies. The significance of the term generalized is that SPOCP is not written to support a specific set of applications but should be possible to use by almost any application. And one SPOCP server should be able to service a set of applications simultaneously.

SPOCP is independent of the particular transmission subsystem and requires only a reliable ordered data stream channel. A companion document [\[SPOCP/TCP\]](#) describes one implementation of SPOCP over TCP.

[TOC](#)

3. The SPOCP model

The SPOCP design is based on the 'PULL sequence' as described in RFC 2904 [\[RFC2904\]](#). That is, a application/some service equipment queries the access control service as to the permission for a specific user to access/use a specific resource.

This has some consequences; one of them being that it is in fact the service equipment that

Simple Policy Control Protocol

makes the decision whether a specific user shall be allowed to perform a specific operation or not. The access control service can only return a recommendation, it can not enforce anything.

One of the design criteria behind Spocp is that it is not uncommon that a simple yes/no answer is sufficient. There are lots of cases where a yes, but.. answer is more appropriate. Hence we have added the concept of return-info, some information that can be returned together with a positive reply. A negative reply is never accompanied by any return information.

Another design criteria was that we did not want to copy lots of information that are already present in different information resources (like a enterprise directory, a whois server, DNS or what have you). This service should be able to use the information where it is and not demand a local copy. Now, we do acknowledge that there are cases where due to security of performance concerns you might have to keep local copies but our aim was to make this a optimization and not the standard. To solve this equation we have added boundary conditions. Boundary conditions, which might be things like; the time has to be between 0800 and 1700 on a working day or the entity that wants to do this has to belong to the group 'admin' as defined in the enterprise directory, are evaluated after a request for a permission has match a policy stored in the authorization server.

And finally, any rule placed in the server represents a permission for someone to use some resource. There are no negative rules, that is rules that says that someone are not allowed to access a resource. The reason for this was that the system compexity increases substantially if negative rules are allowed.

How the service equipment finds the right access control service to ask is outside the scope of this document but will be specified in a separate draft where a access control service location mechanism based on DNS is described.

[TOC](#)

4. SPOCP functions

A 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 response

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

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

If the traffic between a client and a server, happens over TCP/IP, then it can be protected by the use of TLS/SSL [\[RFC2246\]](#) and/or SASL [\[RFC2222\]](#) The STARTTLS and AUTH

commands are used to initialize the TLS and SASL handshakes.

4.1 Input grammar

```
command = starttls / query / add / delete / list / bcondcom logout / begin
/ commit / rollback / subjectcom / capability / saslauthstarttls =
"STARTTLS"query = "QUERY" [path] s-expradd = "ADD" [path] s-expr [ bcond [
return-info ]]delete = "DELETE" [path] ruleidlist = "LIST" [path] *(
"+" / "-" s-expr )logout = "LOGOUT"begin = "BEGIN"commit = "COMMIT"rollback =
"ROLLBACK"subjectcom = "SUBJECT" [ s-exp [ token ]]capability =
"CAPABILITY"saslauth = "AUTH" mech tokenbcondcom = "BCOND" ( "ADD" /
"DELETE" / "REPLACE" ) bcondname [ bcond ]bcond = bcondexpr /
bcondspecbcondexpr = bcondor / bcondand / bcondnot / bcondrefbcondspec =
bcondname ":" typespecificbcondname = 1*dirchartypespecific =
octetstringbcondref = "(" "ref" 1*dirchar ")"bcondand = "(" "and"
1*bcondexpr ")"bcondor = "(" "or" 1*bcondexpr ")"bcondnot = "(" "not"
bcondexpr ")"mech = 1*pcharruleid = 1*pcharpath = "/" / 1*( "/" pathpart
)pathpart = 1*dirchardirchar = pchar / '-' / '_' / '.'pchar = %x30-39 /
%x41-5A / %x61-7Atoken = octet-strings-expr = ; defined in [SPOCP Sexp]
with one restriction that the tag of a list consists solely of dirchar's
```

4.2 Output grammar

```
spocp-response = *(multiline) singlelinemultiline =
multi-responsemulti-response = "201" m-infosingleline = responderesponse =
replycode [ octetstring ]; replycode 201 can not appear in singleline
responsesreplycode = ( "2" / "3" / "4" / "5" ) digit digit [ octetstring
]m-info = return-info / list-infolist-info = path ruleid s-expr [
return-info ]return-info = [ mimecontenttype ] octetstringmimecontenttype =
; as defined in [RFC2045]octet-string = 1*bytebyte = %x00-FF
```

The MIME content-type attribute is an optional attribute which describes the return-info data. This is a string with values defined by [RFC2045](#). This attribute is purely advisory; no validation of the mime type information is required by this specification.

The format of the reply codes follows the common IETF principal as used in SMTP [RFC0821](#) and HTTP [RFC1945](#) to mention a few.

The first digit of the Status-Code defines the class of response. The last two digits do not have any categorization role. There are 5 values for the first digit:

```
1xx: Informational - Not used, but reserved for future use2xx: Success -
The action was successfully received, understood, and accepted.3xx:
Redirection - Further action must be taken in order to complete the request
or an authentication handshake are in progress4xx: Client Error - The
request contains bad syntax or cannot be fulfilled5xx: Server Error - The
server failed to fulfill an apparently valid request
```

A complete list of reply codes are given in [Appendix A](#)

4.3 QUERY

This command takes one argument, a S-expression representing the access permission that is in question.

If a rule in the server matched the given S-expression and there is return information coupled

Simple Policy Control Protocol

to that rule, the return information will be returned together with the positive response, in the multipart. If there are more than one rule that matches, only one of them will be used to construct the response.

The upshot of this is that if there are more than one rule giving a certain permission and all of these return different return-info then a sequence of consecutive queries for this permission, might not necessarily result in the return-info being the same in each reply. 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.

Note: When a permission is checked, it is not checked against the union of all the rules but against each rule separately.

4.4 STARTTLS

This command takes as of now, no arguments

If SSL/TLS is used 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 host name against 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 host name 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 host name 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.

4.5 ADD

The command has at least one argument and at the most three. The first argument is the S-expression that the administrator wants to store in the rule database. The other arguments, if present, is the boundary condition connected to this rules and possible also static return information.

The second argument, if present, can either be a name, which then refers to a boundary condition stored somewhere on the server, or it can be a direct specification of a boundary condition. If there is the need to define some return info but no boundary condition the second argument should be the string "NULL" which then signifies the null boundary condition.

The third argument, if present, really consists of two parts; a optional mime content-type specification and then the information which is treated by the server as a number of bytes and never interpreted in any way. It will be returned to the client exactly as given.

4.6 DELETE

Only one argument and that is the ruleID of the rule to be deleted.

Every rule in the server MUST have a ruleID and if exactly the same rule appears in several servers the ruleID should be the same. To accomplish this the MD5 [\[RFC1321\]](#) message-digest of the S-expression is used as the ruleID.

4.7 LIST

The arguments are a list of sub elements.

The less permissive comparison, as specified in [\[SPOCP Sexp\]](#), is used here. The difference, compared to usage in connection with other commands, 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 is more permissive than the S-expression specified in the query. If on the other hand you specify the '-' as order specifier, 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 sub element, 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
```

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

Simple Policy Control Protocol

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 sub elements 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 who's age are less than or equal to 10, then you have to combine the result of several queries: for instance the one specified above and "LIST +3:age +2:10".

4.8 LOGOUT

No argument. When the server receives this command it will immediately close down the connection.

4.9 BEGIN

No argument.

Marks the start of a transaction, that is a number of commands that should be applied as 'one'

4.10 COMMIT

No argument.

Applies all the commands that has been specified within a transaction.

4.11 ROLLBACK

No argument.

Clears the list of commands that has been added since the 'begin' commands was issued

4.12 SUBJECT

One or two arguments, the first being the 'user' name, the other a token that proves to the server that the user really is who she says she is.

4.13 AUTH

This command takes two arguments: an authentication mechanism and an initial authentication handshake token. This document only specified SASL handshakes but other may be defined in the future. SASL-mechanisms are named using the "SASL:"-prefix; eg SASL:GSSAPI or SASL:SCRAM-MD5. A list of supported authentication mechanisms are listed (among other things) in the output of the CAPABILITY command.

If the server supports and is willing to use the requested mechanism the server sends the token to the underlying mechanism and returns a 301 (Authentication in progress) with a return token for the client mechanism. The client must either continue the handshake with a new AUTH command using the same mechanism or a LOGOUT command. If the handshake fails the server must issue a 508

(Authentication error) and close the connection. If the server does not support the requested mechanism a 405 (Argument error) is returned and the connection is closed. After a success full handshake the client and server uses whatever protection layer has been negotiated (integrity and/or confidentiality) and sends SASL-wrapped tokens instead of the raw bytes. 4.14 CAPABILITY

This command takes no arguments. The return is a 200 with an token containing a space (ascii 32)-separated list of capitalized ascii strings signifying server capabilities. Currently the only capabilities supported are "SASL:"-prefixed SASL [\[RFC2222\]](#) authentication mechanism eg "SASL:GSSAPI".

4.15 BCOND

This command takes two or three arguments: and action to perform a name of a boundary condition and depending on the action a third which would be a specification of a boundary condition.

Boundary conditions are tests that are run when the rule they are connected to is more permissive than the query. Boundary conditions can return TRUE or FALSE, which means that even though the rule matches the final decision is based on the result of the boundary condition evaluation.

[TOC](#)

5. The theory behind return-info

The user is permitted to access the resource, but if she does so special care should be taken as to loggingThe user is permitted to access the resource, but belongs to a set of users that should use a special instance of the resource.The user is permitted to access the resource, and the application can cache this information for a specified amount of timeIf the application is a proxy, and the user has the permission to use the end resource the proxy should use the one-time-password returned in the return-info when accessing the resource

As has been described earlier, an administrator can set the return-info when a access rule is added to the access control service. This kind of return-info is static and will not be changed before it is returned to a application.

To support dynamic construction of return-info in our implementation of a SPOCP server we are allowing back ends that handle boundary conditions to construct and return return-info when condition is evaluated to TRUE. This has meant that we have a couple of back ends connected with boundary conditions that are there more for the side effect that they produce dynamic return-info then for really checking a necessary boundary condition.

[TOC](#)

6. Security considerations

Introducing middleware components have a lot of benefits but also opens new ways of

Simple Policy Control Protocol

attacking applications. Making a application dependent on a middleware component disregarding which service it provides means that the application is not more secure than the least secure of the services it is using. There for the aim of any implementer of or user of a middleware service must be to make it safer than any of the applications it is serving.

If boundary conditions are used the information that they use for their evaluation is as important to protect and verify as the rules that the server is using.

TOC	
References	
[RFC0821]	Postel, J., " Simple Mail Transfer Protocol ", STD 10, RFC 821, August 1982.
[RFC1321]	Rivest, R. , " The MD5 Message-Digest Algorithm ", RFC 1321, April 1992.
[RFC1945]	Berners-Lee, T. , Fielding, R. and H. Nielsen , " Hypertext Transfer Protocol -- HTTP/1.0 ", RFC 1945, May 1996.
[RFC2222]	Myers, J. , " Simple Authentication and Security Layer (SASL) ", RFC 2222, October 1997 (HTML , XML).
[RFC2246]	Dierks, T. and C. Allen , " The TLS Protocol Version 1.0 ", RFC 2246, January 1999.
[RFC2828]	Shirey, R., " Internet Security Glossary ", RFC 2828, May 2000.
[RFC2904]	Vollbrecht, J., Calhoun, P., Farrell, S., Gommans, L., Gross, G., de Bruijn, B., de Laat, C., Holdrege, M. and D. Spence, " AAA Authorization Framework ", RFC 2904, August 2000.
[RFC2045]	Freed, N. and N. Borenstein , " Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies ", RFC 2045, November 1996.
[SPOCP/TCP]	Hedberg, R. , "The Simple Policy Control Protocol over TCP/IP".
[SPOCP Sexp]	Hedberg, R. and O. Bandmann , "Restricted S-expressions for usage in a generalized access control service".

TOC	
Author's Address	
	Roland Hedberg
	Stockholm university
	Kasamark 114
	Umea 90586
	Sweden
Phone:	+46 90 147275
E-Mail:	roland@it.su.se
TOC	

Appendix A. SPOCP Reply Codes

Response code	Response text (might be present)	Description
200	Ok	Command accepted and executed
201		Multi line response
202	Denied	Query did not match any stored policy
203	Bye	Server is closing connection
204	Transaction complete	
301	Authentication in progress	
400	Syntax error	
401	Already in operation	
402	Too many arguments	
403	Line too long	
404	Access denied	
405	Argument error	
406	Not supported	
407	Already exists	Rules or client specifications

Simple Policy Control Protocol

		can not be overwritten
408	Input error	
409	Protocol error	
410	Unknown command	
411	Size limit exceeded	The total number of bytes in the command exceeded the limit set by the server
500	Operations error	Permanent operations error
501	Service not available	
502	Information unavailable	When a remote service is not available, this is when a boundary condition can not be checked.
503	Unknown ID	
504	Already active	
505	Internal error	
506	Time limit exceeded	
507	Other error	For example error encountered while using back ends
509	Authentication error	
510	Not implemented	
TOC		

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the

IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

Full Copyright Statement

Copyright (C) The Internet Society (2004). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assignees.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.