

**NAME**

**rapolicy** – compare a **argus(8)** data file/stream against a Cisco Access Control List.

**COPYRIGHT**

Copyright (c) 2000-2011 QoSient. All rights reserved.

**SYNOPSIS**

**rapolicy -r** *argus-file* [*ra options*]

**DESCRIPTION**

**Rapolicy** reads **argus** data from an *argus-file* list, and tests the argus data stream against a Cisco access control list configuration file, printing out records that represent activity that would violate the policy. **Rapolicy** can be used to indicate access control violations, as well as test new access control definitions prior to installing them in a router.

**OPTIONS**

**Rapolicy**, like all **ra** based clients, supports a large number of options. Options that have specific meaning to **rapolicy** are:

- f <Cisco ACL file> Print records that violate the policy.
- D 0 (default) Print records that violate the policy.
- D 1 Print records and the violated ruleset.
- D 2 Print all records and the ruleset that matched.

See **ra(1)** for a complete description of **ra options**.

**EXAMPLE INVOCATION**

**rapolicy -r** argus.file

**CISCO ACL SYNTAX**

There does not seem to be authoritative Cisco-ACL-Documentation, nor ACL syntax standardization. Because Cisco has been know to improve its ACL rules syntax, **rapolicy** is known to work with Cisco ACL router defintions up to July, 2002.

A Cisco ACL configuration file consists of a collection of any number of ACL statements, each on a separte line. The syntax of an ACL statement is:

ACL = "access-list" ID ACTION PROTOCOL SRC DST NOTIFICATION

ID = Number

ACTION = permit | deny

PROTO = protocol name | protocol number

SRC | DST = ADDRESS [PORTMATCH]

ADDRESS = any | host HOSTADDR | HOSTADDR HOSTMASK

HOSTADDR = IPv4 address

HOSTMASK = matching-mask

PORTMATCH = PORTTOP PORTNUM | range PORTRANGE

PORTOP = eq | lt | gt | neq | established

PORTRANGE = PORTNUM PORTNUM

PORTNUM = TCP or UDP port value (unsigned decimal from 0 to 65535)

## EXAMPLE CONFIGURATION

This example Cisco Access Control List configuration is provided as an example only. No effort has been made to verify that this example Access Control List enforces a useful access control policy of any kind.

```
#allow www-traffic to webserver
access-list 102 permit tcp any 193.174.13.99 0.0.0.0 eq 80

#allow ftp control connection to server
access-list 102 permit tcp any 193.174.13.99 0.0.0.0 eq 21

#allow normal ftp
access-list 102 permit tcp any 193.174.13.99 0.0.0.0 eq 20

#allow ftp passive connctions in portrange 10000 to 10500
access-list 102 permit tcp any host 193.174.13.99 range 10000 10500

#dummy example
access-list 102 permit tcp host 193.174.13.1 eq 12345 host 193.174.13.2 range 12345 23456

#deny the rest
access-list 102 deny tcp any any

#same thing in other words:
access-list 102 deny tcp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
```

## AUTHORS

Carter Bullard (carter@qosient.com).  
Olaf Gellert (gellert@pca.dfn.de).

## SEE ALSO

**ra**(1), **rarc**(5), **argus**(8)