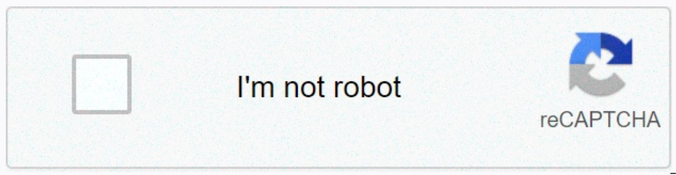

Continue

Cisco asa access- list security- group

You can incorporate Cisco TrustSec policy into many ASA functions. Any feature that uses extended APTs (unless listed in this chapter as unsupported) can take advantage of Cisco TrustSec. You can add security group arguments to extensible AC's, as well as traditional network-based parameters. For example, an access rule allows or denies traffic on an interface using network information. With Cisco TrustSec, you can control access based on security group. For example, you can create an access rule for sample_securitygroup1 10.0.0.0 255.0.0.0, meaning the security group can have any IP address on subnet 10.0.0.0/8. You can configure security policies based on combinations of security group names (servers, users, unmanaged devices, and so on), user-based properties, and traditional IP address-based objects (IP address, Active Directory object, and FQDN). Security group memberships can extend beyond roles to include device and location features and are independent of user group memberships. The following example shows how to create an ACL that uses a locally defined security object group: object group security objgrp-it-admin securitygroupname it-admin-sg-name security group tag 1 object group security objgrp-hr-admin security-group name hr-admin-sg-name // single sg_name group-object this-admin // locally defined object group as nested object-object-group security objg hr-hr-servers security-group name hr-servers-sg-name object-group security objgrp-hr-network security-group tag 2 access-list hr-acl permit ip object-group-group-security objgrp-hr-admin any object group-security objgrp-hr servers The ACL configured in the previous example can be enabled by configuring an access group or the Modular Policy Framework. Additional examples: !match src hr-admin-sg-name of any network to dst host 172.23.59.53 access-list idw-acl permit ip security-group name hr-admin-sg-name any host 172.23.59.53 !match src hr-admin-sg-name of the host 10.1.1.1 to any access list idfw-acl permit ip security-group name hr-admin-sg-name host 10.1.1.1 any !match src tag 22 of any network to dst hr-servers-sg-name any network access-list idfw-acl permit ip security-group tag 22 any security-group name hr-servers-sg-name any !match src user maria from any host to dst hr-servers-sg-name any network access-list idfw-acl permit ip user CSCO\mary any security-group name hr-servers-sg-name any !match src objgrp-hr-admin of any network to dst objgrp-hr-servers any network access-list idfw-acl permit ip object-group-security objgrp-hr-admin any object-group-security objgrp-hr-servers any !match src user Jack from objgrp-hr-network and ip subnet 10.1.1.0/24 ! to objgrp-hr-servers any network access-list idfw-acl permit ip user CSCO\Jack objgrp-hr network 10.1.1.0 255.255.255.0 object group-security objgrp-hr-servers any !match src user Tom of security-group mktg any google.com object network net-Google fqdn google.com access-list sgacl permit ip sec name any object only-Google ! If user Tom or object_group security objgrp-hr-admin must match, ! multiple ACEs can be defined as follows: accesslist idfw-acl2 permit ip user CSCO\Tom 10.1.1.0 255.255.255.0 object group-security objgrp-hr-servers any access-list idfw-acl2 permit object-group security objgrp-hr-admin 10.1.1.0 255.255.255.0 object group-security objgrp-hr-servers any In large networks especially Data Centers, the ACLs may be too large — up to hundreds of lines and hard to configure and manage. Object group-based AJLs provide the solution here – it's smaller, readable, and easier to configure and manage. Not only is the static ACL, but also dynamic ACL deployments for large user environments are befit. Related - ACL on Router vs Firewall The Object Groups feature enables us to classify users, devices, or protocols into groups and apply those groups to access control lists (ACLs). This lets us create access control policies for groups and use object groups instead of IP addresses, protocols, or even port numbers used in conventional APTs. The approach is to use a single ACE to allow an entire

group of users to access a group of user or server groups. Object Groups supports two types of object groups for grouping ACL parameters - Network Object Groups Service Object Groups A network object group is a group of any of the following objects: Any IP address Host IP addresses Hostnames Other network object groups Series Ip addresses Subnets Related - Cisco ASA 5505 Factory Reset A service object group is a group of any of the following objects : Source and destination protocol ports (Such as telnet, SNMP etc) ICMP types (such as echo , echo-reply etc) Top-level protocols (such as TCP, UDP etc)Other service object groups A scenario will help to understand the concept of Object Groups and how with simple commands, we can reduce creating major ACL lines in just a few, HTTP and SMTP. Under table, the requirements include - Related - Cisco ASA Maintenance Questions Configuration of Object Groups - To replicate, the 2 Group of types used here is -- Network -- To specify IP addresses and subnets Service -- To specify port numbers and protocols Now we put it together in an ACL as below -- Let's look at the outcome by showing access list output So in all total 216 line will be created that will encompse all the user and SERVER communication ACLs along with port numbers and protocols. Hence, to summarized a single line (via Object Groups) did the job that 216 lines were originally supposed to perform via A ACLs. Imagine you need to run a Cisco ASA firewall that has hundreds of hosts and dozens of servers behind it, and for each of these devices we need access list rules that allow or refuse traffic. With so many devices you will have a lot of access list statements and can become an administrative nightmare to understand and update the access list. To make our lives a little easier, Cisco introduced the object group on Cisco ASA Firewalls (and also on iOS routers since iOS 12.4.20T). An object group allows you to group objects, it can have a collection of IP addresses, networks, port numbers, etc. Instead of creating an access list with many different statements, we can refer to an object group. This makes the access list smaller and easier to read. When you make changes in the object group, it is also reflected in the access list. There are different types of object groups, let's see what options we have on the ASA: ASA1 (config)# object group ? configure mode commands/options: icmp-type Specify a group of ICMP types, such as echo network Specify a group host or subnet IP addresses protocol Specify a group protocol, such as TCP, etc. security Specify identity features such as security group service Specify a group TCP/UDP ports/services user Specifies single user, local or import user group Let me give a quick explanation of each object group: icmp type can be used to select all the different ICMP types, for example echo, echo reply, tracking, unreachable, etc. networking is used to select IP addresses and/or network addresses. protocol allows you to choose an entire protocol. For example, TCP, UDP, GRE, ESP, AH, OSPF, EIGRP, and many others. security is used for Cisco TrustSec. service is used to select TCP and/or UDP port numbers. user is to select local user groups for Identity Firewall. In this lesson, we will focus on networking (used for IP addresses/network addresses) and service (used for TCP/UDP port numbers). We'll take a look at some examples and you'll see why object groups are very useful. I'll start with a simple example for servers in the DMZ. Let's say we have five web servers in the DMZ. This means we need access to TCP port 80 for their IP addresses. Our access list can look like this: ASA1 (config)# access list HTTP_TO_DMZ permit tcp any host 192.168.3.1 eq 80 ASA1 (config)# access list HTTP_TO_DMZ allow tcp any host 192.168.3.2 eq 80 ASA1 (config)# access list HTTP_TO_DMZ permit tcp any host 192.168.3.3 eq 80 ASA1 (config)# access list HTTP_TO_DMZ permit tcp any host 192.168.3.4 eq 80 ASA1 (config)# access HTTP_TO_DMZ permit tcp any host 192.168.3.5 eq 80 This will work, but we require 5 statements in our access list. Let's see if we can make it smaller by using an object group. First, I'll delete this access list: ASA1 (config)# clear configure access list HTTP_TO_DMZ Now I'll create a network object group where I configure the IP addresses of all my servers in the DMZ: ASA1(config)# object group network WEB_SERVERS ASA1 (config-network-object-group)# network object host 192.168.3.1.1.1.1.1.1.1.1.1.1.1.1.1.11 ASA1 network-object host 192.168.3.2 ASA1 (config-network-object-group)# network-object host 192.168.3.3 ASA1 (config-network-object-group)# network-object host 192.168.3.4 192.168.3.4 network object host 192.168.3.5 The object group is ready, now we will re-create the access list and we will use the object group in it: ASA1(config)# access list HTTP_TO_DMZ permitt tcp any object group WEB_SERVERS eq 80 I have reduced the access list from five statements to just one statement. Instead of specifying each IP address separately, I'm referring to the object group. That's useful right? If you look in the configuration, you'll find this single entry: ASA1 (config)# show run | include HTTP_TO_DMZ access list HTTP_TO_DMZ extended permittop any object group WEB_SERVERS eq www But if you look at the access list, it will show you both the object group and the specific entries: ASA1 (config)# show access list HTTP_TO_DMZ access list HTTP_TO_DMZ; 5 elements; name hash: 0x6ce713ae access list HTTP_TO_DMZ line 1 extended permit tcp any object group WEB_SERVERS eq www (hitcnt=0) 0x0964f55b accesslist HTTP_TO_DMZ line 1 extended permit tcp any host 192.168.3.3.2 eq www (hitcnt=0) 0x461c3d40 accesslist HTTP_TO_DMZ line 1 extended permit tcp any host 192.168.3.2 eq www (hitcnt=0) 0x3413c8db access list HTTP_TO_DMZ line 1 extended allow any host 192.168.3.3 eq www (hitcnt=0) 0x5ee1c727 accesslist HTTP_TO_DMZ line 1 extended permittcp any host 192.168.3.4 eq www (hitcnt=0)x 089dd7 access list HTTP_TO_DMZ line 1 extended permit tcp any host 192.168.3.5 eq www (hitcnt=0) 0x68e87688 The previous example should give you a good idea how you can use object groups to make your access list smaller. Let's continue by adding a few more requirements. Let's say our web servers need access to a few extra TCP ports... besides TCP port 80 we also need access to 22, 23 and 443. We can update our access list to add these ports: ASA1 (config)# access list HTTP_TO_DMZ allow any object group WEB_SERVERS eq 22 ASA1 (config)# access list HTTP_TO_DMZ allow tc any object group WEB_SERVERS eq 23 ASA1(config)# access HTTP_TO_DMZ permitt any object group WEB_SERVERS eq 443 This does the job, but now we have 4 statements... one for each TCP port. Instead of specifying the TCP port in each statement, we'll create a different object group that combines all of our TCP ports. Here's what it will look like: like: