Policy-Based Routing

https://campus.barracuda.com/doc/20644562/

Transparently Routing Web Traffic to the Barracuda Web Security Gateway

This article demonstrates how to route traffic to the Barracuda Web Security Gateway as a proxy without requiring proxy rules to be pushed out to all clients on the network. This method allows the Barracuda Web Security Gateway to forward HTTPS (443) traffic in addition to standard HTTP traffic, which cannot be done using other methods of transparent proxy routing.

Important Notes:

- The example shown in this article assumes a configuration with a Cisco Router with built-in Firewall Security Module (FWSM), but it should work with any routing equipment supporting Policy-Based Routing (PBR).

Run this command on the router

Text in yellow boxes shows commands that need to be run on the router.

Installation of the Barracuda Web Security Gateway

For this configuration, you will need to connect the Barracuda Web Security Gateway LAN interface to its own dedicated port on the router. Give the Barracuda Web Security Gateway an IP address in its own dedicated IP subnet, and assign a gateway IP to the router interface that it is connected to. An example network is shown here:

Barracuda Networks IP Address:
10.100.3.2/30 gateway 10.100.3.1

Internal Ranges:
10.100.1.0/24 (VLAN_1)
10.100.2.0/24 (VLAN_2)

Router Configuration

Step 1. Define 2 access lists

You must define two access lists because you need to create a route-map for both the internal and external interfaces of the router. These rules describe which clients will be routed to the Barracuda Web Security Gateway. Your routing rules will be different based on whether this is outbound or inbound traffic.

Run these commands on the router

[Inbound]
ip access-list extended HTTP(S)_Proxy_Inbound
permit udp any eq domain 10.100.0.0 0.0.255.255
permit tcp any eq 443 10.100.0.0 0.0.255.255

[Outbound]
ip access-list extended HTTP(S)_Proxy_Outbound
permit tcp 10.100.0.0 0.0.255.255 any eq www
permit tcp 10.100.0.0 0.0.255.255 any eq 443

Note that this is routing inbound DNS traffic back through the Barracuda Web Security Gateway. This is the key to making policy-based routing work for HTTPS traffic.

Step 2. Create route maps

Match these route-maps to the access lists you just created. Any traffic matching those lists will have the “match” rule applied to it. In this case, you are modifying the next-hop for the packet to the Barracuda Web Security Gateway's IP address. Note that you need two route-maps—one for inbound traffic, and one for outbound traffic.

Run these commands on the router

[Inbound]
route-map HTTP(S)_Proxy_Inbound permit 10
match ip address HTTP(S)_Proxy_Inbound
set ip next-hop 10.100.3.2

[Outbound]
route-map HTTP(S)_Proxy_Outbound permit 20
match ip address HTTP(S)_Proxy_Outbound

set ip next-hop 10.100.3.2

Step 3. Apply route-maps to the interfaces on your router

The inbound route-map you created is applied to the outside (WAN-side) interface on your router/firewall. The outbound route-maps are applied to any internal interfaces on your router/firewall. This includes any sub-interfaces that are connected to client networks that need filtering.

[Inbound]
interface FastEthernet0/1
description Test WAN
ip address 1.1.1.2 255.255.255.0
ip access-group Inbound_Rules in
no ip redirects
no ip unreachables
ip nat outside

Run this command on the router

ip policy route-map HTTP(S)_Proxy_Inbound

duplex auto
speed auto

[Outbound]
Note that there are *two* interfaces listed here—one for each VLAN on the test network. The outbound route-map rule needs to be enabled for each internal interface or sub-interface to be filtered. Start with one and test.

interface FastEthernet0/0.1
description VLAN_1
encapsulation dot1Q 1
ip address 10.100.1.1 255.255.255.0
ip nat inside

Run this command on the router

ip policy route-map HTTP(S)_Proxy_Outbound

interface FastEthernet0/0.2
description VLAN_2
encapsulation dot1Q 2

ip address 10.100.2.1 255.255.255.0
ip nat inside

Run this command on the router

ip policy route-map HTTP(S)_Proxy_Outbound

Sample Cisco IOS Configuration

```
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
service password-encryption
!
hostname cisco
!
boot system flash slot1:c3660-ik9o3s-mz.122-32.bin
no logging monitor
enable secret 5 **********
!
username seadmin privilege 15 password 7 **********
ip subnet-zero
ip wccp web-cache redirect-list WCCP
!!
ip ftp username **********
ip ftp password 7 **********
ip domain-name **********
!
ip audit notify log
ip audit po max-events 100
!!
call rsvp-sync
!
!!
!!
fax interface-type fax-mail
mta receive maximum-recipients 0
!!
!
interface FastEthernet0/0
description Test LAN
```

```
no ip address
ip nat inside
duplex auto
speed auto
!
interface FastEthernet0/0.1
description Barracuda Systems
encapsulation dot1Q 1
ip address 10.100.1.1 255.255.255.0
ip nat inside
ip policy route-map HTTP(S)_Proxy_Outbound
!
interface FastEthernet0/0.2
description Other OS (Windows, Mac, Linux...)
encapsulation dot1Q 2
ip address 10.100.2.1 255.255.255.0
ip nat inside
!
interface FastEthernet0/0.100
encapsulation dot1Q 100 native
!
interface FastEthernet0/1
description CudaSE.net WAN
ip address 1.1.1.3 255.255.255.0 secondary
ip address 1.1.1.2 255.255.255.0
ip access-group Inbound_Rules in
no ip redirects
no ip unreachables
ip nat outside
ip policy route-map HTTP(S)_Proxy_Inbound
duplex auto
speed auto
!
interface FastEthernet2/0
description HTTP(S) Proxy
ip address 10.100.3.1 255.255.255.0
duplex auto
speed auto
!
ip nat inside source list Outbound_NAT interface FastEthernet0/1 overload
ip classless
ip route 0.0.0.0 0.0.0.0 1.1.1.1
no ip http server
!
!
ip access-list extended HTTP(S)_Proxy_Inbound
```

```
permit udp any eq domain 10.100.0.0 0.0.255.255
permit tcp any eq 443 10.100.0.0 0.0.255.255
ip access-list extended HTTP(S)_Proxy_Outbound
permit tcp 10.100.0.0 0.0.255.255 any eq www
permit tcp 10.100.0.0 0.0.255.255 any eq 443
ip access-list extended Inbound_Rules
permit icmp any any echo
permit icmp any any echo-reply
permit icmp any any source-quench
permit icmp any any packet-too-big
permit icmp any any time-exceeded
permit udp any any gt 1023
permit tcp any any ack
deny ip any any
ip access-list extended Outbound_NAT
permit ip 10.100.1.0 0.0.0.255 any
permit ip 10.100.2.0 0.0.0.255 any
permit ip 10.100.3.0 0.0.0.255 any
deny ip any any
route-map HTTP(S)_Proxy_Inbound permit 10
match ip address HTTP(S)_Proxy_Inbound
set ip next-hop 10.100.3.2
!
route-map HTTP(S)_Proxy_Outbound permit 20
match ip address HTTP(S)_Proxy_Outbound
set ip next-hop 10.100.3.2
!!
dial-peer cor custom
!
!!
!!
line con 0
line aux 0
line vty 0 4
privilege level 15
login local
transport input telnet ssh
line vty 5 15
privilege level 15
login local
transport input telnet ssh
!
end
```