

Introduction

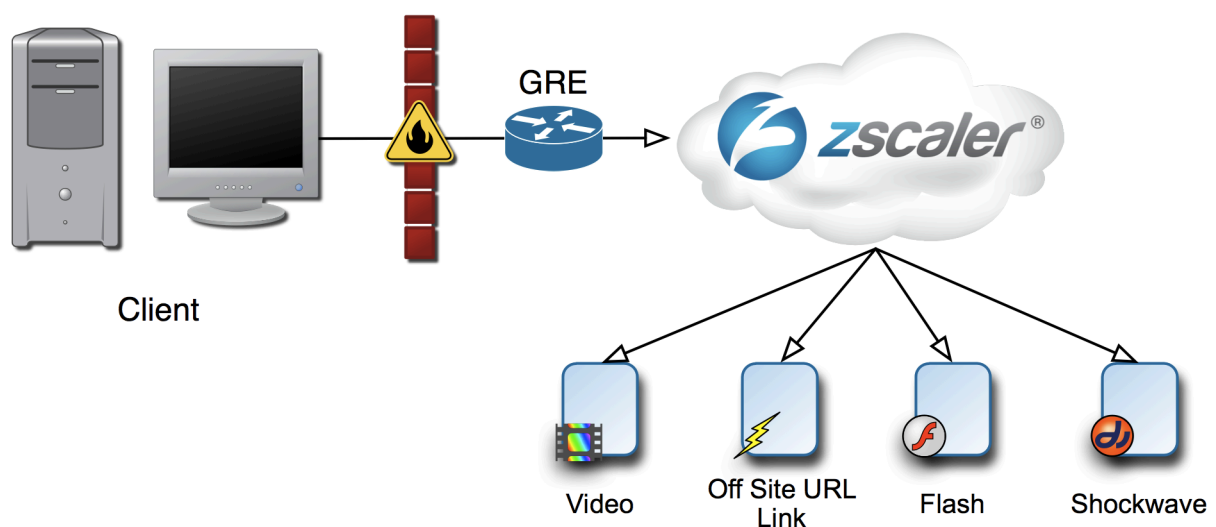
As a SaaS provider, Zscaler's web security services run in the "cloud"—a logical collection of physical Zscaler components strategically placed around the world both for redundancy and to create minimal latency. Zscaler Enforcement Nodes (ZENs) are proprietary, high-speed web proxies that enforce inbound and outbound policy for each customer, as well as block viruses, malware, and other bad content. These ZENs, in turn, communicate with other components in the cloud: Nanolog™ servers which store all your weblog data and central authority servers which provide the administrative interface to Zscaler and stores each company's policy.

The only thing required of customers is to forward or re-direct their outbound web traffic to one of Zscaler's ZENs. There are four methods for doing this:

- **GRE Tunnels:** Rules on a firewall or edge router are created to provide a secure tunnel to a specific ZEN. This is the preferred method of traffic forwarding because it supports both HTTP and HTTPS traffic, as well as fail-over to a second (or third) ZEN should the first enforcement node become unavailable for any reason.
- **Proxy Chaining:** An existing internal web proxy (e.g., Bluecoat or ISA forward proxies) is used to forward HTTP traffic sent to it by internal firewall rules to a ZEN.
- **Port Forwarding:** Firewalls (e.g., Checkpoint or Juniper firewalls) allow port 80 traffic to be forwarded to another host—in this case a ZEN.
- **Web Browser Proxy Configuration:** For those users outside of the corporate network, the browser itself can be configured to forward its HTTP/HTTPS requests to a ZEN.

This paper addresses the first bullet: **GRE Tunnels**.

Using an existing router to redirect traffic to Zscaler



Picture 1 Using GRE to redirect http/https traffic to Zscaler

Pros:

1. Multiple tunnels offer full datacenter redundancy
2. Users cannot bypass Zscaler

Cons:

1. Users off the corporate network may not be protected by Zscaler

Virtual IP Addresses with GRE Tunnels

When a customer wants to create GRE tunnels to ensure failover to a secondary Datacenter or ZEN (Zscaler Enforcement Node), Zscaler assigns Virtual IP addresses for use as the source and destination address inside the tunnel. These addresses are assigned from a pool of non-routable address space which Zscaler manages to ensure that no two customers attempt to use the same IP addresses. When a customer indicates they would like to create GRE Tunnels, Customer Support will assign eight IP addresses per Internet Gateway Location. The customer (or partner—depending on from whom the request originates) will receive something from Support like the following:

172.17.0.80 - 172.17.0.87

This is a total of eight addresses—four for the primary GRE Tunnel rule and four for the secondary GRE Tunnel rule. The customer *must* use the following addresses:

GRE Tunnel primary rule:

172.17.0.80: Not used—this identifies the network address

172.17.0.81: Used as the **SOURCE** tunnel address (customer's firewall/router)

172.17.0.82: Used as the **DESTINATION** tunnel address (Zscaler primary Datacenter IP)

172.17.0.83: Not used—this identifies the broadcast address

GRE Tunnel secondary rule:

172.17.0.84: Not used—this identifies the network address

172.17.0.85: Used as the **SOURCE** tunnel address (customer's firewall/router)

172.17.0.86: Used as the **DESTINATION** tunnel address (Zscaler secondary Datacenter IP)

172.17.0.87: Not used—this identifies the broadcast address

You must tell Zscaler Support which Datacenter IP the customer wants to use as their primary and secondary destination. When Zscaler assigns these VIPs the system will bind the source and destination addresses to the specified primary and secondary Datacenter/ZENs; the IP-Addresses will be listening specifically for traffic from the source VIP and addressed to its destination VIP. When the customer actually creates his GRE Tunnel rules and sends traffic to Zscaler, the ZEN will associate the virtual source and destination addresses with the specific customer.

In summary, only the second, third, sixth, and seventh addresses are actually used. The 2nd and 3rd addresses are used as the primary rule's tunnel source and destination, and the 6th and 7th addresses are used as the secondary rule's tunnel source and destination.

GRE Tunneling: Cisco router

This is a basic Cisco router configuration:

```
## SAMPLE ZSCALER GRE CONFIGURATION
## DO NOT USE WITHOUT CONSULTING YOUR ZSCALER SE
#
# You will be assigned 8 IP's from Customer Support. For this example,
# assume they're 172.17.0.1-.8.
#
# Tunnel0
```

```
# 172.17.0.1 - network IP (unused)
# 172.17.0.2 - customer side
# 172.17.0.3 - Zscaler side
# 172.17.0.4 - broadcast address (unused)
#
# Tunnel1
# 172.17.0.5 - network IP (unused)
# 172.17.0.6 - customer side
# 172.17.0.7 - Zscaler side
# 172.17.0.8 - broadcast address (unused)
#
# This configuration will work on routers that are already doing NAT as well
# as on those that aren't. Be sure to read this article to see why we use
# route-maps and not ACL's to match on traffic:
#
# http://tinyurl.com/2dzdq2
#
# If your router is already doing NAT with ACL's, you will have to migrate
# them over
# to route-maps.
#
# In our example, 192.168.1.0/24 is the internal network which hides behind a
# public
# IP of 1.2.3.4.
#
## Define the 2 tunnel interfaces. Be sure to change the interface
## names to the router's *outside* interface. Also change the
## tunnel destinations to the correct Zscaler nodes.
interface Tunnel0
  description Zscaler primary - FMT
  ip address 172.17.0.x 255.255.255.252
  ip tcp adjust-mss 1436
  ip nat outside
  ip virtual-reassembly
  keepalive 5 3
  tunnel source GigabitEthernet0/1
  tunnel destination 72.x.x.x

interface Tunnel1
  description Zscaler secondary - ORD
  ip address 172.17.0.x 255.255.255.252
  ip tcp adjust-mss 1436
  ip nat outside
  ip virtual-reassembly
  keepalive 5 3
  tunnel source GigabitEthernet0/1
  tunnel destination 208.x.x.x

## Define the ACLs which is the outbound traffic that we want to match on.
## Make sure access-list 151 isn't being used, or change the number.
ip access-list extended ZscalerRedirect
  permit tcp 192.168.1.0 0.0.0.255 any eq www
  permit tcp 192.168.1.0 0.0.0.255 any eq 443

## Use route-maps to define which traffic to match on, not ACL's
route-map ZS-TUNNEL0-NAT permit 10
  match ip address ZscalerRedirect
  match interface Tunnel0

route-map ZS-TUNNEL1-NAT permit 10
  match ip address ZscalerRedirect
  match interface Tunnel1
```

```
## Perform the NAT
ip nat inside source route-map ZS-TUNNEL0-NAT interface Tunnel0 overload
ip nat inside source route-map ZS-TUNNEL1-NAT interface Tunnel1 overload

## Define the policy route which sends traffic over Tunnel0 first, then
Tunnel1
## if Tunnel0 is unavailable
route-map ZS-HTTP-REDIRECT permit 10
  match ip address ZscalerRedirect
  set interface Tunnel0 Tunnel1

## Attach the route-map to the *internal* interface. Also need to make sure
that this
## same interface is defined as the inside for NAT purposes
int GigabitEthernet0/0
  ip nat inside
  ip policy route-map ZS-HTTP-REDIRECT
```

GRE Tunneling: Cisco router with redundant ISP links

1. With these assumptions you should have a router with 2 public IP's on it. You'll first need to get 4 tunnels provisioned by Customer Support, one set for your first public IP and another set for the backup public IP. Let's assume now that you have Tunnel0 through Tunnel3 defined with the local (Cisco) side of the tunnels having IP's 172.17.1.1, 172.17.1.5, 172.17.1.9, and 172.17.1.13.
2. First, make sure you have your ACL defined to match the outbound traffic you're interested in. For example:

```
ip access-list extended ZscalerRedirect
  permit tcp 192.168.1.0 0.0.0.255 any eq www
  permit tcp 192.168.1.0 0.0.0.255 any eq 443
```

3. Create the route-maps for the NAT'ing:

```
route-map ZS-TUNNEL0-NAT permit 10
  match ip address ZscalerRedirect
  match interface Tunnel0
!
route-map ZS-TUNNEL1-NAT permit 10
  match ip address ZscalerRedirect
  match interface Tunnel1
!
route-map ZS-TUNNEL2-NAT permit 10
  match ip address ZscalerRedirect
  match interface Tunnel2
!
route-map ZS-TUNNEL3-NAT permit 10
  match ip address ZscalerRedirect
  match interface Tunnel3
```

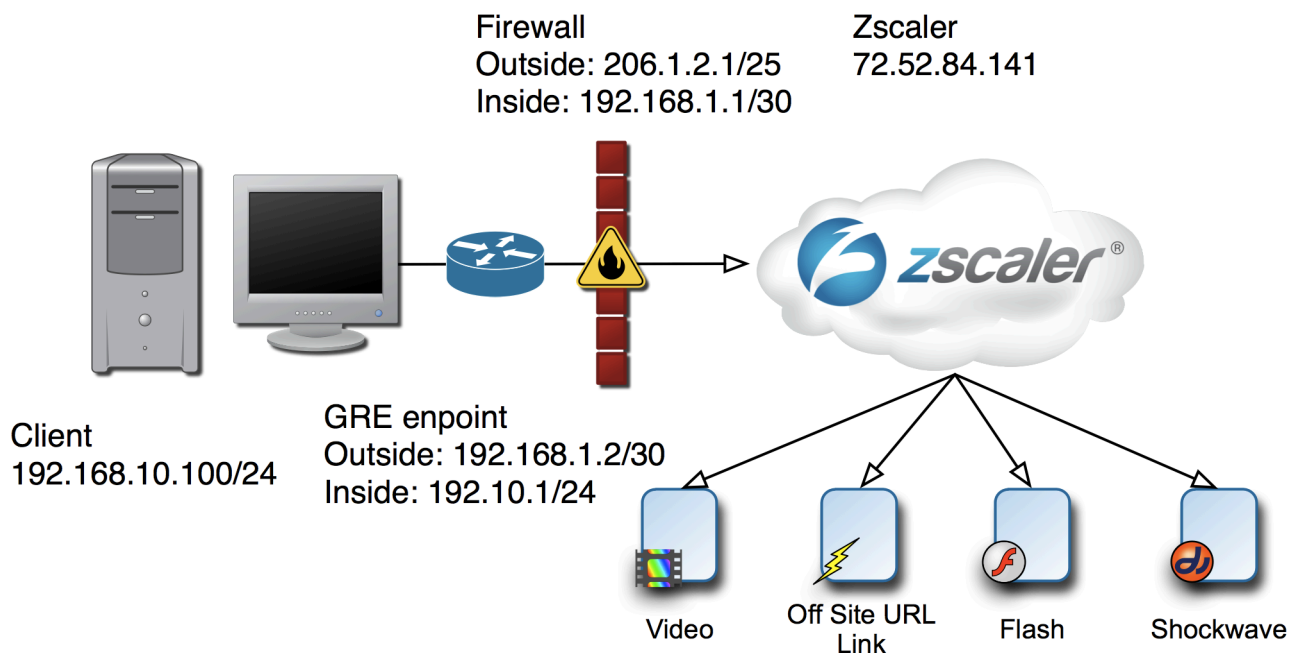
4. Perform the NAT:

```
ip nat inside source route-map ZS-TUNNEL0-NAT interface Tunnel0 overload
ip nat inside source route-map ZS-TUNNEL1-NAT interface Tunnel1
overload
ip nat inside source route-map ZS-TUNNEL2-NAT interface Tunnel2
overload
ip nat inside source route-map ZS-TUNNEL3-NAT interface Tunnel3
overload
```

- Modify the main zscaler-gre route-map to include all 4 tunnel interfaces in order. Note that in this setup Tunnel1 will only be used if Tunnel0 is down, Tunnel2 will only be used if both Tunnel0 and Tunnel1 are down, and so on....

```
route-map zscaler-gre permit 10
  match ip address ZscalerRedirect
  set interface Tunnel0 Tunnel1 Tunnel2 Tunnel3
```

GRE Tunneling: Cisco router (internal) behind a firewall



Picture 2 Zscaler with NATed GRE

Basically we will be terminating the tunnel on an internal router that has a private IP address. The challenge is that we need to have the "tunnel source" on the internal router be a public IP, otherwise the ZEN will not be able to direct the keepalive packets back to the router over the WAN.

The solution to this is to use a public IP'd loopback interface on the internal router. For this example let's assume the customer also owns the public IP 206.1.2.2 and it will be assigned to the router. **Note that you will need a dedicated public IP for this solution.**

- The internal router configuration will be similar to what we've used in the past for more simple environments. There are some differences though, namely creating the loopback interface and changing the tunnel source.

```
interface Loopback0
  ip address 206.1.2.2 255.255.255.255
  !
interface Tunnel0
  description Zscaler primary
  ip address 172.17.1.1 255.255.255.252
  ip tcp adjust-mss 1436
  ip nat outside
  ip virtual-reassembly
  keepalive 5 3
  tunnel source Loopback0
```

```
tunnel destination 72.52.84.141
```

2. On the external firewall (here we assume that it's a PIX) we will need to ensure a few things. First make sure the PIX passes thru all traffic for the router's public IP.

```
static (inside,outside) 206.1.2.2 206.1.2.2 netmask 255.255.255.255
```

3. Next, still on the PIX, we need to route that public IP to the private IP of the router.

```
route inside 206.1.2.2 255.255.255.255 192.168.1.2 1
```

4. Still on the PIX, we need to make sure the ACLs are allowing the GRE traffic through from the inside router to the outside GRE endpoint(s). **You will probably have to change the ACL names to fit your customer environment, this is only an example.**

```
...
...
access-list Inside-ACL permit gre host 206.1.2.2 host 72.52.84.141
...
...
```

GRE Tunneling: Cisco router within a MPLS cloud (VRF)

This is a Cisco router configuration hooked into an MPLS cloud using VRF:

```
version 12.4
!
hostname Router
!
boot-start-marker
boot-end-marker
!
resource policy
!
ip subnet-zero
!
!
ip cef
no ip dhcp use vrf connected
!
!
ip vrf OUTSIDE
 rd 10:1
 route-target export 10:1
 route-target import 10:1
!
!
!
interface Tunnel0
 ip vrf forwarding OUTSIDE
 ip address 172.17.0.249 255.255.255.252
 ip accounting output-packets
 ip mtu 1476
 ip nat outside
 ip virtual-reassembly
 keepalive 5 3
 tunnel source 196.30.147.108
 tunnel destination 66.8.60.251
 tunnel vrf OUTSIDE
!
interface FastEthernet0/0
```

```
ip vrf forwarding OUTSIDE
ip address 196.30.147.108 255.255.255.240
duplex auto
speed auto
!
interface FastEthernet0/1
ip vrf forwarding OUTSIDE
ip address 192.168.1.111 255.255.255.0
ip nat inside
ip virtual-reassembly
ip policy route-map ZSCALER-TRAFFIC_RTMAP
duplex auto
speed auto
!
!
ip classless
ip route vrf OUTSIDE 66.8.60.251 255.255.255.255 196.30.147.97
ip route vrf OUTSIDE 196.31.215.10 255.255.255.255 196.30.147.97
!
!
ip nat inside source list ZSCALER-TRAFFIC interface Tunnel0 vrf OUTSIDE
overload
!
ip access-list extended ZSCALER-TRAFFIC
 permit tcp host 192.168.1.53 any eq www
 permit tcp host 192.168.1.53 any eq 443
!
access-list 150 permit ip any any log
!
route-map ZSCALER-TRAFFIC_RTMAP permit 10
 match ip address ZSCALER-TRAFFIC
 set interface Tunnel0
!
route-map TUNNEL-0-NAT permit 10
 match ip address ZSCALER-TRAFFIC
!
!
!
End
```

GRE Tunneling: Juniper/Netscreen Firewall

This is a Juniper/Netscreen Firewall configuration using GRE:

```
set clock timezone 0
set vrouter trust-vr sharable
set vrouter "untrust-vr"
exit
set vrouter "trust-vr"
unset auto-route-export
exit
set auth-server "Local" id 0
set auth-server "Local" server-name "Local"
set auth default auth server "Local"
set auth radius accounting port 1646
set admin name "admin"
set admin password "nIpyAqrJDM1Oc8PCKsfMVlPtDgIMkn"
set admin auth timeout 10
set admin auth server "Local"
set admin format dos
set zone "Trust" vrouter "trust-vr"
set zone "Untrust" vrouter "trust-vr"
set zone "DMZ" vrouter "trust-vr"
```

```
set zone "VLAN" vrouter "trust-vr"
set zone "Untrust-Tun" vrouter "trust-vr"
unset zone "Trust" tcp-rst
unset zone "Untrust" block
unset zone "Untrust" tcp-rst
unset zone "DMZ" tcp-rst
set zone "VLAN" block
unset zone "VLAN" tcp-rst
set zone "Untrust" screen tear-drop
set zone "Untrust" screen syn-flood
set zone "Untrust" screen ping-death
set zone "Untrust" screen ip-filter-src
set zone "Untrust" screen land
set zone "V1-Untrust" screen tear-drop
set zone "V1-Untrust" screen syn-flood
set zone "V1-Untrust" screen ping-death
set zone "V1-Untrust" screen ip-filter-src
set zone "V1-Untrust" screen land
set interface "ethernet0/0" zone "Untrust"
set interface "ethernet0/1" zone "DMZ"
set interface "wireless0/0" zone "Trust"
set interface "bgroup0" zone "Trust"
set interface "tunnel.1" zone "Untrust"
set interface bgroup0 port ethernet0/2
set interface bgroup0 port ethernet0/3
set interface bgroup0 port ethernet0/4
unset interface vlan1 ip
set interface ethernet0/0 ip 64.105.89.6/29
set interface ethernet0/0 nat
set interface ethernet0/1 ip 192.168.12.1/24
set interface ethernet0/1 nat
set interface wireless0/0 ip 192.168.25.1/24
set interface wireless0/0 nat
set interface bgroup0 ip 192.168.24.128/24
set interface bgroup0 nat
set interface tunnel.1 ip 176.16.1.1/24
set interface tunnel.1 tunnel encap gre
set interface tunnel.1 tunnel local-if ethernet0/0 dst-ip 72.52.84.140
set interface tunnel.1 mtu 1476
set interface wireless0/0 proxy dns
unset interface vlan1 bypass-others-ipsec
unset interface vlan1 bypass-non-ip
set interface ethernet0/0 ip manageable
set interface ethernet0/1 ip manageable
set interface wireless0/0 ip manageable
set interface bgroup0 ip manageable
set interface ethernet0/0 manage ping
set interface ethernet0/0 manage ssl
set interface bgroup0 manage mtrace
set zone V1-Untrust manage ping
set interface wireless0/0 dhcp server service
set interface bgroup0 dhcp server service
set interface wireless0/0 dhcp server auto
set interface bgroup0 dhcp server auto
set interface wireless0/0 dhcp server option dns1 192.168.25.1
set interface bgroup0 dhcp server option dns1 64.105.172.26
set interface wireless0/0 dhcp server ip 192.168.25.33 to 192.168.25.126
set interface bgroup0 dhcp server ip 192.168.24.33 to 192.168.24.126
unset interface wireless0/0 dhcp server config next-server-ip
unset interface wireless0/0 dhcp server config updatable
unset interface bgroup0 dhcp server config next-server-ip
set interface tunnel.1 dip 4 176.16.1.10 176.16.1.10
set interface tunnel.1 dip interface-ip incoming
set interface "serial0/0" modem settings "USR" init "AT&F"
```

```
set interface "serial0/0" modem settings "USR" active
set interface "serial0/0" modem speed 115200
set interface "serial0/0" modem retry 3
set interface "serial0/0" modem interval 10
set interface "serial0/0" modem idle-time 10
set interface wireless0 wlan 0
set flow tcp-mss
unset flow tcp-syn-check
set domain sc.localeng.com
set pki authority default scep mode "auto"
set pki x509 default cert-path partial
set dns host dns1 64.105.172.26 src-interface ethernet0/0
set dns host dns2 0.0.0.0
set dns host dns3 0.0.0.0
set dns host name gateway.zscaler.net 72.52.84.140
set dns host name login.zscaler.net 38.102.162.130
set dns proxy
set dns proxy enable
set dns server-select domain * outgoing-interface ethernet0/0 primary-server
64.105.172.26 failover
set ike respond-bad-spi 1
unset ike ikeid-enumeration
unset ike dos-protection
unset ipsec access-session enable
set ipsec access-session maximum 5000
set ipsec access-session upper-threshold 0
set ipsec access-session lower-threshold 0
set ipsec access-session dead-p2-sa-timeout 0
unset ipsec access-session log-error
unset ipsec access-session info-exch-connected
unset ipsec access-session use-error-log
set url protocol websense
exit
set policy id 2 name "forwarding80" from "Trust" to "Untrust" "Any" "Any"
"HTTP" nat src dip-id 4 permit
set policy id 2
exit
set policy id 3 name "General" from "Trust" to "Untrust" "Any" "Any" "ANY"
nat src permit
set policy id 3
exit
set policy id 4 name "HTTPS" from "Trust" to "Untrust" "Any" "Any" "HTTPS"
nat src dip-id 4 permit
set policy id 4 application "HTTP"
set policy id 4
exit
set nsmgmt bulkcli reboot-timeout 60
set ssh version v2
set config lock timeout 5
set wlan 0 channel auto
set wlan 1 channel auto
set ssid name SCJUN
set ssid SCJUN authentication wpa-psk passphrase
pTX/40mQNd+lzasjcaC0d8X4dpnGHQS5HA== encryption auto
set ssid SCJUN interface wireless0
set snmp port listen 161
set snmp port trap 162
set vrouter "untrust-vr"
set source-routing enable
exit
set vrouter "trust-vr"
unset add-default-route
set route 0.0.0.0/0 interface ethernet0/0 gateway 64.105.89.1 preference 20
permanent
```

```
set access-list extended 1 dst-port 80-80 protocol tcp entry 1
set access-list extended 2 dst-port 443-443 protocol tcp entry 1
set match-group name http
set match-group http ext-acl 1 match-entry 1
set match-group http ext-acl 2 match-entry 2
set action-group name httpGRE
set action-group httpGRE next-interface tunnel.1 action-entry 1
set pbr policy name HTTPForward
set pbr policy HTTPForward match-group http action-group httpGRE 1
set pbr HTTPForward
exit
set interface wireless0/0 pbr HTTPForward
set interface bgroup0 pbr HTTPForward
set zone Trust pbr httpviaGRE
set vrouter "untrust-vr"
exit
set vrouter "trust-vr"
exit
```

GRE Tunneling: Fortinet on FortiOS

1. Create the GRE Tunnel Interface

```
config system interface
  edit "zscaler"
    set vdom "root"
    set ip 172.17.8.113 255.255.255.255
    set allowaccess ping
    set status up
    set type tunnel
    set remote-ip 172.17.8.114
    set interface "port2" << External Interface.
  next
end
```

2. Create Policy routes for Port 80 and 443 traffic

```
config router policy
  edit 1
    set input-device "port1" <<Internal Interface
    set src 10.0.0.0 255.0.0.0
    set protocol 6
    set start-port 80
    set end-port 80
    set gateway 172.17.8.114
    set output-device "zscaler"
  next
end
config router policy
  edit 2
    set input-device "port1"
    set src 10.0.0.0 255.0.0.0
    set protocol 6
    set start-port 443
    set end-port 443
    set gateway 172.17.8.114
    set output-device "zscaler"
  next
end
```

3. Create Firewall ACL's for traffic

```
config firewall policy
  edit (Policy ID Goes here)
    set srcintf "port1"
    set dstintf "zscaler"
    set srcaddr "INT-Internet_Access_Group"
    set dstaddr "all"
    set action accept
    set ippool enable
    set poolname "Workstations"
    set schedule "always"
    set service "HTTP"
    set nat enable
  next
end
config firewall policy
  edit (Policy ID Goes here)
    set srcintf "port1"
    set dstintf "zscaler"
    set srcaddr "INT-Internet_Access_Group"
    set dstaddr "all"
    set action accept
    set ippool enable
    set poolname "Workstations"
    set schedule "always"
    set service "HTTPS"
    set nat enable
  next
```

GRE Tunneling: without doing NAT inside of the tunnel

```
## SAMPLE ZSCALER GRE CONFIGURATION
## DO NOT USE WITHOUT CONSULTING YOUR ZSCALER SE
#
# You will be assigned 8 IP's from Customer Support. For this
example,
# assume they're 172.17.0.1-.8.
#
# Tunnel0
# 172.17.0.2 - customer side
# 172.17.0.3 - Zscaler side
#
# Tunnel1
# 172.17.0.6 - customer side
# 172.17.0.7 - Zscaler side
#
# This configuration will work on routers that are already doing
NAT as well
# as on those that aren't. RFC1918 IP space can be sent over the
GRE tunnels
# to Zscaler and routed back to the customer seamlessly.
#

## Define the 2 tunnel interfaces. Be sure to change the
interface
## names to the router's *outside* interface. Also change the
## tunnel destinations to the correct Zscaler nodes.
interface Tunnel0
  description Zscaler primary - FMT
  ip address 172.17.0.x 255.255.255.252
  ip tcp adjust-mss 1436
  ip virtual-reassembly
```

```
keepalive 5 3
tunnel source GigabitEthernet0/1
tunnel destination 72.x.x.x

interface Tunnell
description Zscaler secondary - ORD
ip address 172.17.0.x 255.255.255.252
ip tcp adjust-mss 1436
ip virtual-reassembly
keepalive 5 3
tunnel source GigabitEthernet0/1
tunnel destination 208.x.x.x

## Define the ACLs which is the outbound traffic that we want to
match on.
ip access-list extended ZscalerRedirect
 permit tcp 192.168.1.0 0.0.0.255 any eq www
 permit tcp 192.168.1.0 0.0.0.255 any eq 443

## Define the policy route which sends traffic over Tunnel0 first,
then Tunnell
## if Tunnel0 is unavailable
route-map ZS-HTTP-REDIRECT permit 10
 match ip address ZscalerRedirect
 set interface Tunnel0 Tunnell

## Attach the route-map to the *internal* interface.
int GigabitEthernet0/0
 ip policy route-map ZS-HTTP-REDIRECT
```