128T WITH ZSCALER INTERNET ACCESS DEPLOYMENT GUIDE

Abstract

How to allow your 128T router to send traffic to the Zscaler Cloud Security Platform

22 June 2018

CONTENTS

I. ZSCALER CONFIGURATION

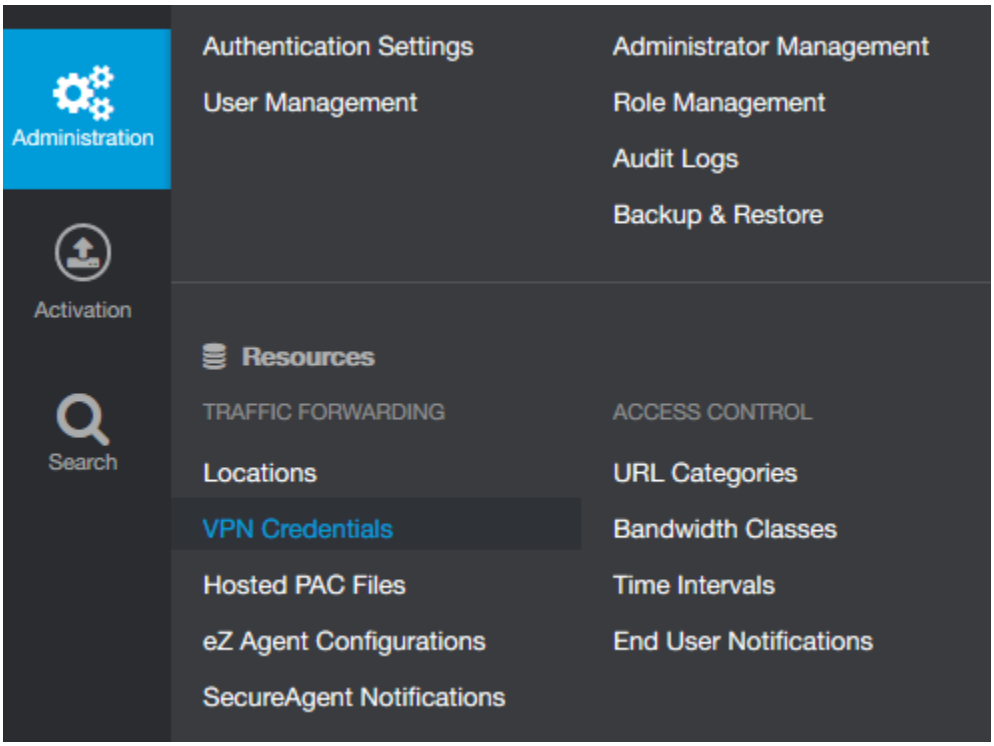
PROVISION THE PUBLIC IP ADDRESS(ES) OF THE 128T

The first step that must happen is to provision the public address from where the IPSec traffic will be initiated towards Zscaler. The Zscaler endpoint that tunnels will be established towards are referred to as ZIA Public Service Edge

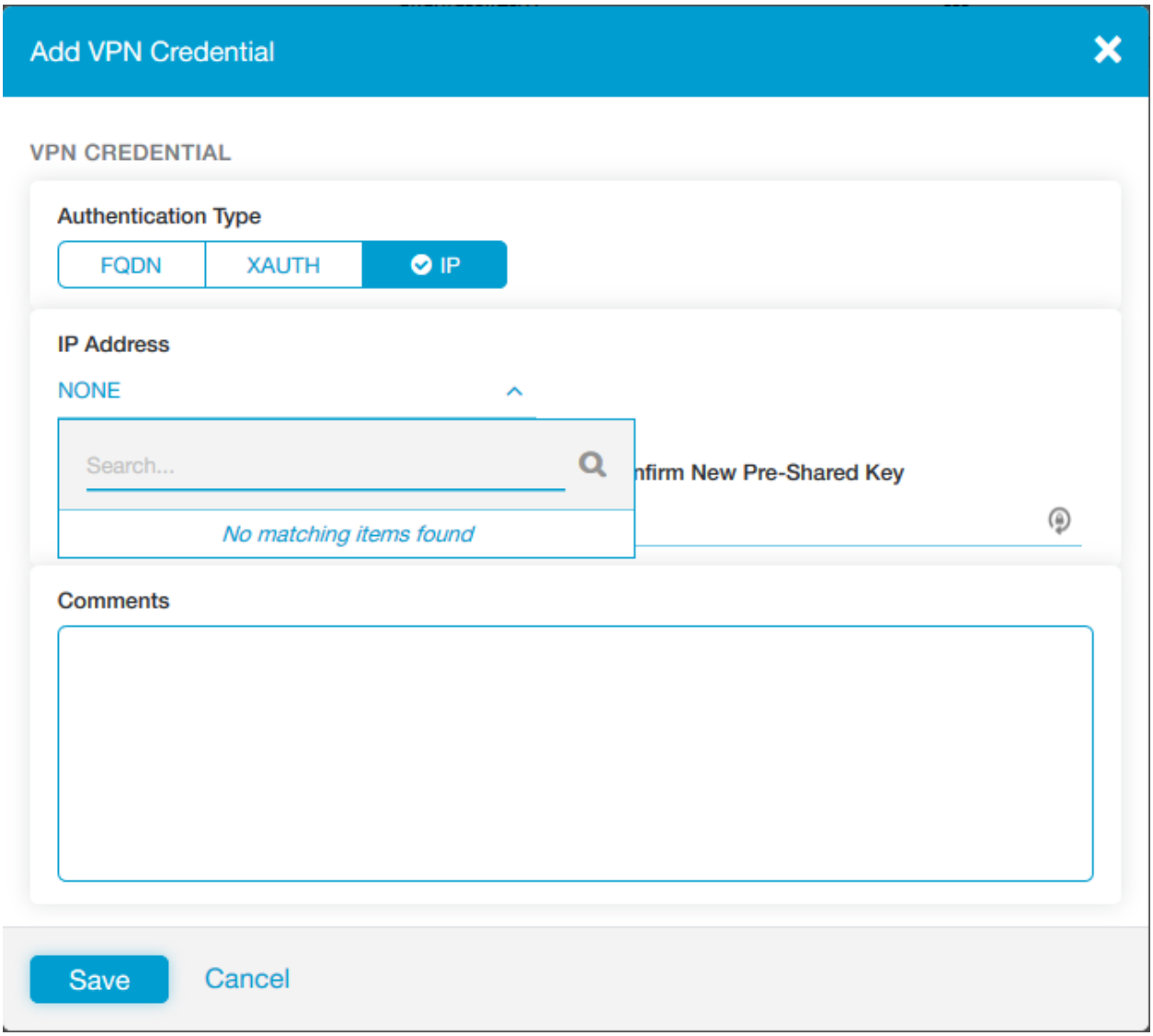
A support ticket should be opened with Zscaler listing the public IP addresses of all sites that will be connected to Zscaler so that they can be whitelisted on the Zscaler side. Once you have received word from Zscaler support that this work has been completed, you can move forward with the next steps.

PROVISION VPN CREDENTIALS

In the Zscaler portal, navigate to VPN Credentials as shown below.



From there, select the option to add VPN credentials. You will be prompted with options as shown below.



For the authentication type, select "IP". When you click the IP addresses field, you should be prompted with a list of the public IP addresses you submitted in the previous step. Select the address for the site you wish to setup.

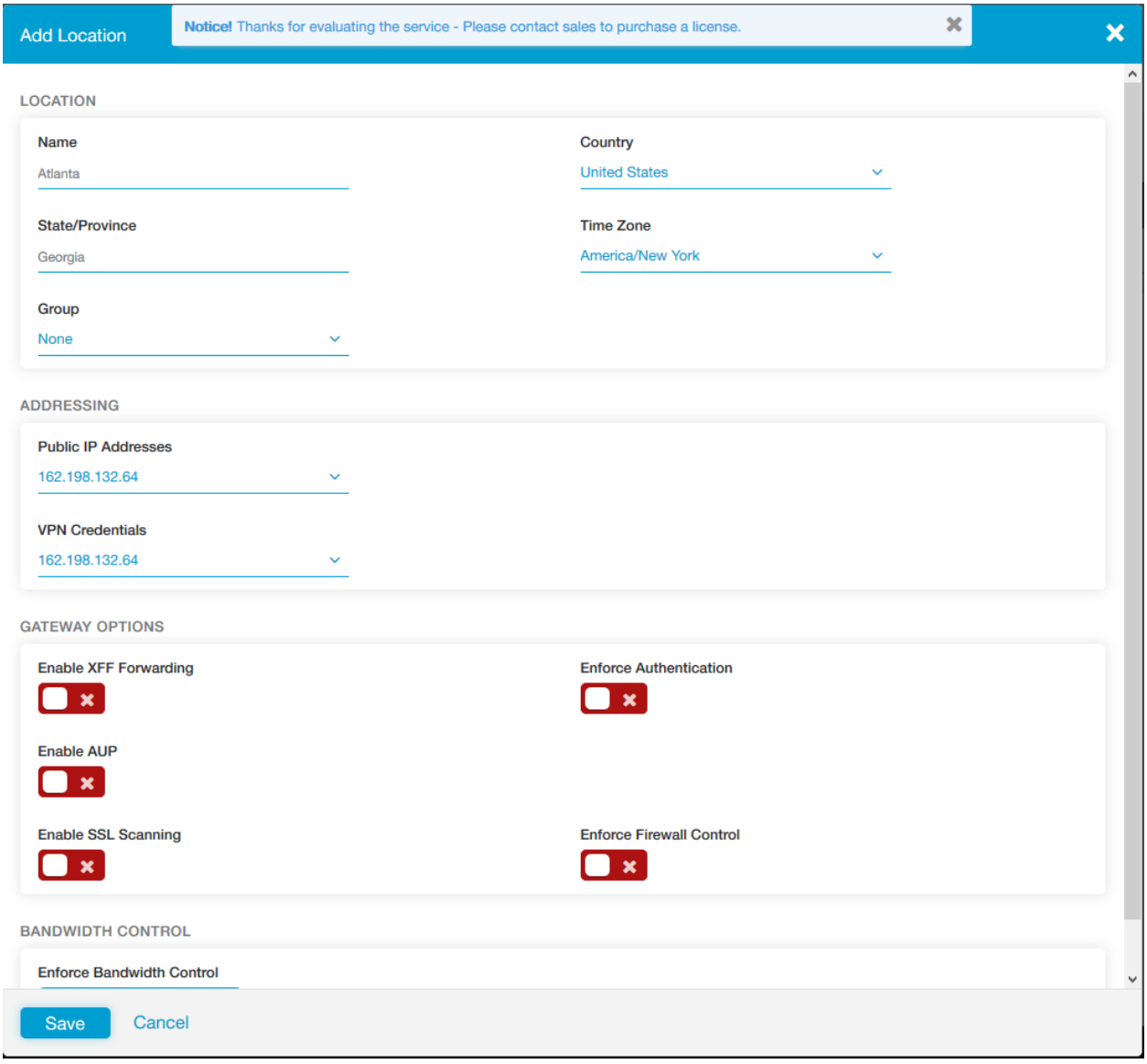
At this point you need to create a pre-shared key that will be used on both ends of the connection. It is advisable to create a random string that does not use dictionary words. One method for creating a random key is to issue following command on a Linux system with OpenSSL installed:

```
[root@west ~]# openssl rand -base64 48
```

Enter and confirm your PSK and save it somewhere for reference when you configure the 128T side, then click the Save button.

CONFIGURE A LOCATION

Once you have created your VPN credentials, you may now create a Location in the Zscaler GUI to correspond to the site where the 128T resides. From the administration menu, select the location option (just above VPN Credentials). You will be prompted with a menu as shown below. Enter the appropriate information for your site. Select this site's public IP address from the list of available addresses and select the corresponding VPN Credentials to map to this site. Click Save when completed.



FIND THE ADDRESSES OF THE TUNNEL TERMINATION ZIA PUBLIC SERVICE EDGE

Zscaler provides services on multiple cloud environments. When a customer is provisioned, a customer is provisioned in a specific cloud. For testing we were provided access to Betacloud: https://admin.zscalerbeta.net. In order to find the correct ZENs* for your cloud environment, replace "admin" with "ips", for example https://ips.zscalerbeta.net. From there click the "Cloud Enforcement Node Ranges" option from the menu on the left. An example is shown below.

Location	IP Address (CIDR Notation)	Proxy Hostname	GRE Virtual IP	VPN Host Name	Notes
Europe					Copy IP Addresses
Frankfurt IV	165.225.72.0/22	fra4.sme.zscalerbeta.net	165.225.72.38	fra4-vpn.zscalerbeta.net	
US & Canada					Copy IP Addresses
San Francisco IV	199.168.148.0/23	sunnyvale1.sme.zscalerbeta.net	199.168.148.131	sunnyvale1-vpn.zscalerbeta.net	
Washington DC	104.129.194.0/23	was1.sme.zscalerbeta.net	104.129.194.38	was1-vpn.zscalerbeta.net	

In this example, we will choose the VPN Host Name in region that is closest to our site as the primary ZEN* (was1-vpn.zscalerbeta.net) and the other as the backup ZEN (sunnyvale1-vpn.zscalerbeta.net).

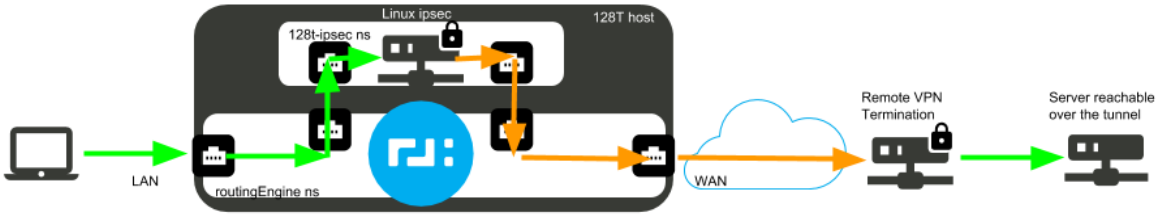
We will need to convert these names into IP addresses for later in the process. This can be done from Linux by using the ping command as shown below.

```
[t128@localhost ~]$ ping was1-vpn.zscalerbeta.net
PING was1-vpn.zscalerbeta.net (104.129.194.39) 56(84) bytes of data.
```

**ZIA Public Service Edge is the new name for what was previously known as "Zscaler Enforcement Node" or ZEN.*

II. 128T CONFIGURATION

We will use Linux to establish the IPSec tunnels. To pass the LAN traffic into the VPN tunnel, and to allow the IPSec traffic out the WAN interface managed by 128T, we will service function chain the traffic through KNI interfaces as shown in the drawing below.



To avoid conflicts with existing Linux routes, we will create a new namespace for this traffic. We will create two kni host interfaces and move them into the new namespace: one for customer traffic in and one for IPSec traffic out.

The following section provides the low-level steps to setup and configure IPSec for this purpose. This is the "phase 0" implementation of Zscaler support. In future software releases, Zscaler support will be more tightly integrated into the 128T product

SETUP IPSEC

INSTALL THE LIBRESWAN PACKAGE

Install Libreswan with yum. This setup was tested and validated with libreswan-3.20-5.el7_4.x86_64

```
[root@zscaler-128t ~]# yum install libreswan
```

CREATE THE 128T-IPSEC SYSTEMD SERVICE

This service will be used to launch the IKE daemon inside our 128t-ipsec namespace. Please open the following file in your preferred text editor and paste in the contents.

/etc/systemd/system/128t-ipsec.service:

```
[Unit]
Description=Internet Key Exchange (IKE) Protocol Daemon for IPsec running in 128T managed namespace
Wants=network-online.target
Documentation=man:ipsec(8) man:pluto(8) man:ipsec.conf(5)

[Service]
Type=notify
Restart=always
# backwards compatible with plutorestartoncrash=no
#RestartPreventExitStatus=137 143 SIGTERM SIGKILL
# Set WatchdogSec to the amount of time (in seconds) that systemd will wait
# before restarting an unresponsive pluto.
# EVENT_SD_WATCHDOG updates the heartbeat every 15 seconds, recommended values
# are 60, 90, 120. WatchdogSec=0 disables the action
NotifyAccess=all
WatchdogSec=200
# Check configuration file
ExecStartPre=/sbin/ip netns exec 128t-ipsec /usr/libexec/ipsec/addconn --config /etc/ipsec.conf --checkconfig
# Check for kernel modules
ExecStartPre=/sbin/ip netns exec 128t-ipsec /usr/libexec/ipsec/_stackmanager start
# Check for nss database status and migration
ExecStartPre=/sbin/ip netns exec 128t-ipsec /usr/sbin/ipsec --checknss
# Check for nflog setup
ExecStartPre=/sbin/ip netns exec 128t-ipsec /usr/sbin/ipsec --checknflog
# Start the actual IKE daemon
ExecStart=/sbin/ip netns exec 128t-ipsec /usr/libexec/ipsec/pluto --leak-detective --config /etc/ipsec.conf --nofork
ExecStop=/sbin/ip netns exec 128t-ipsec /usr/libexec/ipsec/whack --shutdown
ExecStopPost=/sbin/ip netns exec 128t-ipsec /sbin/ip xfrm policy flush
ExecStopPost=/sbin/ip netns exec 128t-ipsec /sbin/ip xfrm state flush
ExecStopPost=/sbin/ip netns exec 128t-ipsec /usr/sbin/ipsec --stopnflog
ExecReload=/sbin/ip netns exec 128t-ipsec /usr/libexec/ipsec/whack --listen

[Install]
```

SETUP THE ALTERNATE UPDOWN SCRIPT

The default updown script will not re-establish the routes to the vti interfaces should the tunnels go down and come back up. Therefore, we will create a new version of this file and reference it in the Zscaler IPSec configuration file. First, copy the original /usr/libexec/ipsec/_updown.netkey file to a new location:

```
[root@zscaler-128t ~]# cp /usr/libexec/ipsec/_updown.netkey /usr/libexec/ipsec_updown_route.sh
```

Using your favorite text editor, edit the new file and add the action 'uproute' to the 'up-client' command, which is called when the tunnel is re-established.

```
--- /usr/libexec/ipsec/_updown.netkey   2018-05-07 22:24:31.916720083 -0400
+++ /usr/libexec/ipsec_updown_route.sh  2018-05-07 10:12:45.166477846 -0400
@@ -676,6 +676,7 @@
        addcat
        addsource
        notifyNM connect
+       uproute
        ;;
    down-client)
        # connection to my client subnet going down
```

Please note that the above output is the Linux diff command output comparing the old and new file. Please simply open the file /usr/libexec/ipsec_updown_route.sh in your favorite text editor, go to line 752, look for the line that reads "notifyNM connect" and add a new line below that containing the text "uproute" aligned with the indentation above it. Do not include the "+" sign.

CREATE THE ZSCALER IPSEC CONFIGURATION FILE

This file defines two tunnels, one to each Zscaler ZEN *identified in the first section of this document. We will configure the tunnels to use the Zscaler specified settings for encryption/authentication and phase2. We will setup Dead Peer Detection to the Zscaler specified minimum timer of 10 seconds.

Open the file /etc/ipsec.d/zscaler.conf using your favorite text editor. Copy and paste the contents of the following text box and change the

ZIA Public Service Edge is the new name for what was previously known as "Zscaler Enforcement Node" or ZEN.

highlighted values to match your setup (there are four places that need to be changed). Each value for right must match one of the two remote ZEN* IP addresses. Give the IP address of the closer location in the section for zscaler1 and the other IP address in the section for zscaler2. The value leftid in both locations must match this site's public IP address as configured in the VPN credentials portion of the Zscaler setup from the first section of this guide.

```
conn zscaler1
  authby=secret
  auto=start
  ike=aes128-sha1;MODP1024
  ikev2=insist
  keyexchange=ike
  ikelifetime=120m
  salifetime=30m
  phase2=esp
  phase2alg=null-md5;MODP1024
  replay-window=16384
  compress=no
  pfs=no
  type=tunnel
  mark=5/0xffffffff
  vti-interface=vti01
  vti-routing=yes
  vti-shared=no
  dpddelay=10
  dpdtimeout=15
  dpdaction=restart
  leftupdown="/usr/libexec/ipsec_updown_route.sh --route y"

  metric=100
  right=104.129.194.39
  rightsubnet=0.0.0.0/0
  left=169.254.32.2
  leftsubnet=0.0.0.0/0
  leftid=162.198.132.64
```

*ZIA Public Service Edge is the new name for what was previously known as "Zscaler Enforcement Node" or ZEN. ZIA Public Service Edge IP

```
conn zscaler2
  authby=secret
  auto=start
  ike=aes128-sha1;MODP1024
  ikev2=insist
  keyexchange=ike
  ikelifetime=120m
  salifetime=30m
  phase2=esp
  phase2alg=null-md5;MODP1024
  replay-window=16384
  compress=no
  pfs=no
  type=tunnel
  mark=6/0xffffffff
  vti-interface=vti02
  vti-routing=yes
  vti-shared=no
  dpddelay=10
  dpdtimeout=15
  dpdaction=restart
  leftupdown="/usr/libexec/ipsec_updown_route.sh --route y"

  metric=200
  right=199.168.148.132
  rightsubnet=0.0.0.0/0
  left=169.254.32.2
  leftsubnet=0.0.0.0/0
  leftid=162.198.132.64
```

SETUP THE IPSEC SECRETS FILE

Using your favorite text editor, open the file /etc/ipsec.d/zscaler.secrets and enter the following content, changing the highlighted values. The entries below have word-wraped because of length. Your file should contain only two lines, both starting with a "%" sign. On each line, you should replace the IP address with the IP address of one of the Zscaler ZENs (also used for the values of "right" in the configuration

file from the previous section. Then, replace the long string between the quotation marks with the appropriate pre-shared Key for this connection as recorded from the "Provision VPN Credentials" section of this document.

```
%any 104.129.194.39 : PSK
"FAR5a/JbBfB0WKt0y2kg5wJHTK4ELDk8p2+eVaBS5oZCa5xRxN9ra639Lg3RwuX5"
%any 199.168.148.132 : PSK
"FAR5a/JbBfB0WKt0y2kg5wJHTK4ELDk8p2+eVaBS5oZCa5xRxN9ra639Lg3RwuX5"
```

CONFIGURING 128T FOR IPSEC SFC

SETUP THE PLUGIN SCRIPTS

We will create two plugin scripts to create the 128t-ipsec namespace, move the interface into the namespace, setup the interface address, and any required routes. Using your favorite text editor, please open the file /etc/128technology/plugins/network-scripts/host/zscaler-in/init (create any non-existent directories in this path) and paste in the following contents:

```
#!/bin/bash
NAMESPACE=128t-ipsec
KNI_NAME=zscaler-in
KNI_ADDRESS=169.254.31.2
KNI_GATEWAY=169.254.31.1
KNI_MASK=30

# create namespace if it doesn't exist
if [ ! -e "/var/run/netns/$NAMESPACE" ]; then
  echo "$NAMESPACE namespace does not exist...creating it."
  ip netns add $NAMESPACE
  ip netns exec $NAMESPACE ip link set lo up
  echo "$NAMESPACE created."
  echo "Setting ip_forwarding in namespace $NAMESPACE."
  ip netns exec $NAMESPACE sysctl -w net.ipv4.ip_forward=1
  echo "Disabling send_redirects in namespace $NAMESPACE."
  ip netns exec $NAMESPACE sysctl -w net.ipv4.conf.all.send_redirects=0
```

```
  echo "Disabling accept_redirects in namespace $NAMESPACE."
  ip netns exec $NAMESPACE sysctl -w net.ipv4.conf.all.accept_redirects=0
  echo "Disabling Reverse Packet Filtering for $VPN_IN_KNI_NAME."
  ip netns exec $NAMESPACE sysctl -w net.ipv4.conf.all.rp_filter=0
  ip netns exec $NAMESPACE sysctl -w net.ipv4.conf.$VPN_IN_KNI_NAME.rp_filter=0
fi

# set up KNI if it exists in the default namespace
if [ -d "/sys/devices/virtual/net/$KNI_NAME" ]; then
  echo "$KNI_NAME found in default namespace."
  echo "Moving $KNI_NAME to $NAMESPACE namespace."
  ip link set $KNI_NAME netns $NAMESPACE
  ip netns exec $NAMESPACE ip a add $KNI_ADDRESS/$KNI_MASK dev $KNI_NAME
  ip netns exec $NAMESPACE ip l set $KNI_NAME up
  # Route RFC1918 space
  ip netns exec $NAMESPACE ip r add 10.0.0.0/8 via $KNI_GATEWAY dev $KNI_NAME
  ip netns exec $NAMESPACE ip r add 172.16.0.0/12 via $KNI_GATEWAY dev $KNI_NAME
  ip netns exec $NAMESPACE ip r add 192.168.0.0/16 via $KNI_GATEWAY dev $KNI_NAME
fi
```

Next, using your favorite text editor, please open the file /etc/128technology/plugins/network-scripts/host/zscaler-out/init (create any non-existent directories in this path) and paste in the following contents:

```
#!/bin/bash
NAMESPACE=128t-ipsec
KNI_NAME=zscaler-out
KNI_ADDRESS=169.254.32.2
KNI_MASK=30
KNI_GATEWAY=169.254.32.1
IPSEC_PEER1_ADDRESS=104.129.194.39
IPSEC_PEER2_ADDRESS=199.168.148.132

# create namespace if it doesn't exist
if [ ! -e "/var/run/netns/$NAMESPACE" ]; then
  echo "$NAMESPACE namespace does not exist...creating it."
  ip netns add $NAMESPACE
```

```
  ip netns exec $NAMESPACE ip link set lo up
  echo "$NAMESPACE created."
  echo "Setting ip_forwarding in namespace $NAMESPACE."
  ip netns exec $NAMESPACE sysctl -w net.ipv4.ip_forward=1
  echo "Disabling send_redirects in namespace $NAMESPACE."
  ip netns exec $NAMESPACE sysctl -w net.ipv4.conf.all.send_redirects=0
  echo "Disabling accept_redirects in namespace $NAMESPACE."
  ip netns exec $NAMESPACE sysctl -w net.ipv4.conf.all.accept_redirects=0
  echo "Disabling Reverse Packet Filtering for $VPN_IN_KNI_NAME."
  ip netns exec $NAMESPACE sysctl -w net.ipv4.conf.all.rp_filter=0
  ip netns exec $NAMESPACE sysctl -w net.ipv4.conf.$VPN_IN_KNI_NAME.rp_filter=0
fi

# set up KNI if it exists in the default namespace
if [ -d "/sys/devices/virtual/net/$KNI_NAME" ]; then
  echo "$KNI_NAME found in default namespace."
  echo "Moving $KNI_NAME to $NAMESPACE namespace."
  ip link set $KNI_NAME netns $NAMESPACE
  ip netns exec $NAMESPACE ip a add $KNI_ADDRESS/$KNI_MASK dev $KNI_NAME
  ip netns exec $NAMESPACE ip l set $KNI_NAME up
  ip netns exec $NAMESPACE ip r add $IPSEC_PEER1_ADDRESS via $KNI_GATEWAY dev
$KNI_NAME
  ip netns exec $NAMESPACE ip r add $IPSEC_PEER2_ADDRESS via $KNI_GATEWAY dev
$KNI_NAME
  systemctl start 128t-ipsec
fi
```

Change the two highlighted IP addresses to match the IP addresses of the two remote Zscaler ZIA Public Service Edges you are using.

After you have saved both files, run the following two commands to ensure these scripts are executable:

```
[root@zscaler-128t ~]# chmod 744 /etc/128technology/plugins/network-scripts/host/zscaler-in/init
 [root@zscaler-128t ~]# chmod 744 /etc/128technology/plugins/network-scripts/host/zscaler-
out/init
```

ADD THE REQUIRED 128T CONFIGURATION ELEMENTS

Through the 128T CLI, please add the following configuration elements to the "authority" level of your configuration.

```
tenant zscaler
   name zscaler
exit

service zscaler-internet
   name              zscaler-internet
   security          internal
   address           0.0.0.0/0

   access-policy     lan
         source lan
   exit
   share-service-routes false
exit

service zscaler-ipsec
   name              zscaler-ipsec
   security          internal
   address           199.168.148.132/32
   address           104.129.194.39/32

   access-policy     zscaler
        source zscaler
   exit
   share-service-routes false
exit
```

Your access policy under the zscaler-internet service should match the name of the tenant (or tenants) on your LAN to which you want to grant access to the Internet. Also, the two IP addresses in the zscaler-ipsec service should match the addresses of the two ZIA Public Service Edges you are connecting.

Next, we must configure the 128T KNI interfaces that will connect to the 128t-ipsec namespace in order to service function chain with IPSec. Please enter the following items under the node element in the router associated with the site you are configuring:

```
device-interface  zscaler-out
  name             zscaler-out
  type             host

  network-interface  zscaler-out
        name       zscaler-out
        tenant     zscaler

        address    169.254.32.1
           ip-address     169.254.32.1
           prefix-length  30
           gateway        169.254.32.2
        exit
  exit
exit

device-interface  zscaler-in
  name             zscaler-in
  type             host

  network-interface  zscaler-in
        name       zscaler-in

        address    169.254.31.1
           ip-address     169.254.31.1
           prefix-length  30
           gateway        169.254.31.2
        exit
  exit
exit
```

Finally we will create service routes to route the traffic associated with the zscaler-internet and Zscaler-ipsec service out the appropriate interfaces. Please enter the following entries under the router object associated with the location you are configuring.

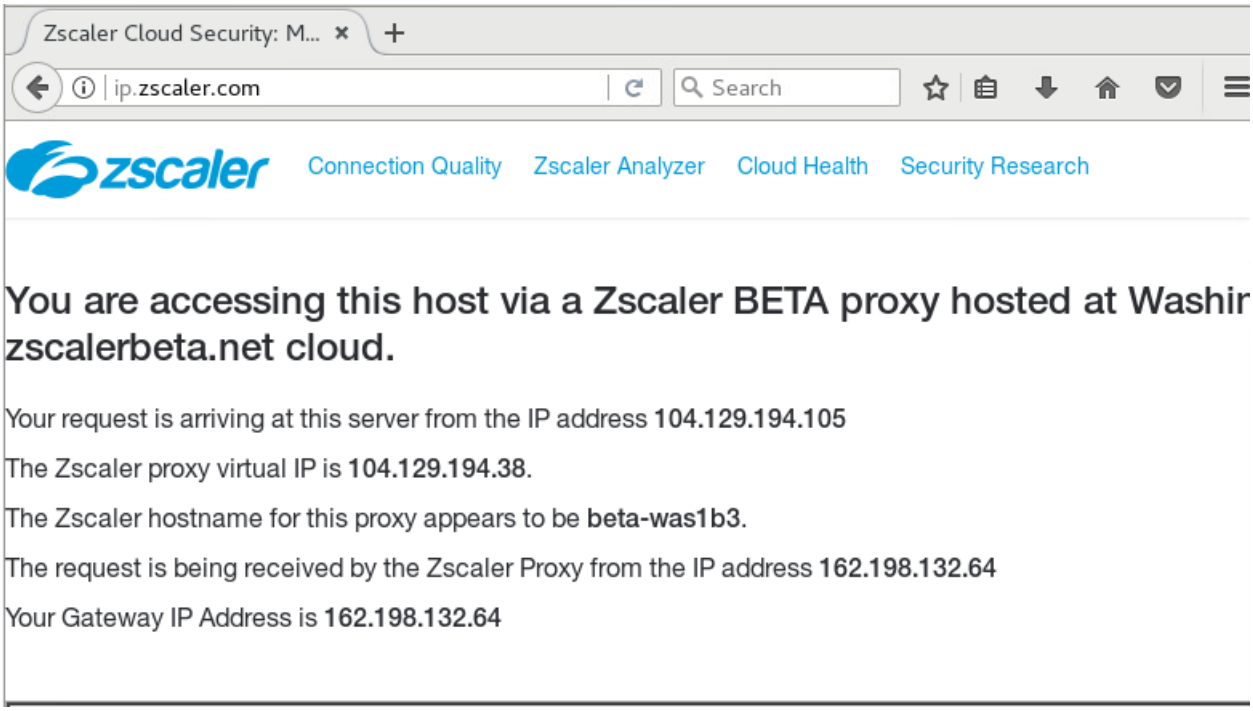
```
service-route  internet
```

```
        name          internet
        service-name  zscaler-internet

        next-hop      zscaler-test-128t zscaler-in
          node-name   zscaler-test-128t
          interface   zscaler-in
          gateway-ip  169.254.31.2
        exit
    exit

    service-route  zscaler-ipsec
        name          zscaler-ipsec
        service-name  zscaler-ipsec

        next-hop      zscaler-test-128t wan
          node-name   zscaler-test-128t
          interface   wan
          gateway-ip  172.25.0.1
        exit
    exit
```

Please replace the highlighted IP address with the value for the next hop gateway to your ISP at this location. Also replace the node name with the correct node name for the system you are configuring.

ZSCALER VERIFICATION

Once the configuration has been completed, verify that your Internet traffic is flowing through Zscaler. To do this, form a client on the LAN of your 128T router, browse to https://ip.zscaler.com. If traffic is successfully flowing through Zscaler, you should see a page that looks like the image below.



If the service is not working, either the page will not load or you will see a page that looks like the following screenshot.