riverbed

SOLUTION GUIDE

Riverbed SteelApp Solution Guide

Load Balancing Microsoft Direct Access

Mark Boddington
Version 1.0

Contents

1.0 Solution Overview

1.1 Microsoft Direct Access

Microsoft Direct Access is a VPN replacement technology which provides seamless connectivity for mobile users (remote employees) back into the corporate network. Unlike traditional VPNs, Direct Access is integrated directly into the Windows Operating System, and will attempt to establish a connection whenever an internet connection is available, and without the need for any user interaction.

The Direct Access system creates secure connections for both the machine and the user. The machine connection can be established at boot or as soon as a network becomes available. The user connection is established when the user logs in.

An important feature of Direct Access is Manage Out; this feature allows an administrator inside the corporate network to connect out to Direct Access clients and perform administration tasks and remote support.

1.2 Direct Access Architecture

Inbound Communications

Microsoft Direct Access is built entirely on IPv6 and so needs to utilise IPv4 transition technologies wherever it needs to cross or access resources on IPv4 networks. When clients connect in across the IPv4 internet, they will try to use Teredo, 6in4 or IPHTTPS tunnels.

In practice however, IPHTTPS is the only method which supports all DA features, and it is the only protocol we will deal with in this guide. It may be possible to use SteelApp with Teredo, but it's not something that we have tested.

If the internal corporate LAN is IPv4 then the Direct Access requests will go through NAT64 and DNS64 transitions to connect to internal resources.

IPHTTPS Recommended

It is our recommendation to deploy DirectAccess using IPHTTPS because:

- 6in4 tunnels cannot be used when the client is behind any kind of NAT
- Teredo tunnels cannot be used when the server is behind a NAT.

Most importantly….

- IPHTTPS is needed for Manage-Out traffic when using an ELB (External Load Balancer).

> Microsoft Note: Deploying Direct Access in a cluster: http://technet.microsoft.com/en-us/library/jj134175.aspx
>
> *"In ELB deployments, if manage out is needed, then DirectAccess clients cannot use Teredo. Only IPHTTPS can be used for end-to-end communication."*

Outbound Communications

When using IPHTTPS the Client IPv6 range will use a 59 bit prefix, and each Direct Access server will subnet that down to a 64 bit prefix. This sub-netting gives us a maximum of 32 Direct Access servers in a load balanced cluster.

IPv6 Outbound (Manage Out)

If the internal LAN is IPv6 then connecting out to the Direct Access clients is a simple case of routing, and there's no explicit requirement for an ADC. However if you are using IPv4 internally then you will need to deploy ISATAP on to the workstations which need connectivity out to the Direct Access clients. This is something which can be configured on a SteelApp ADC by following the steps in section 2.2.

Network Location Services

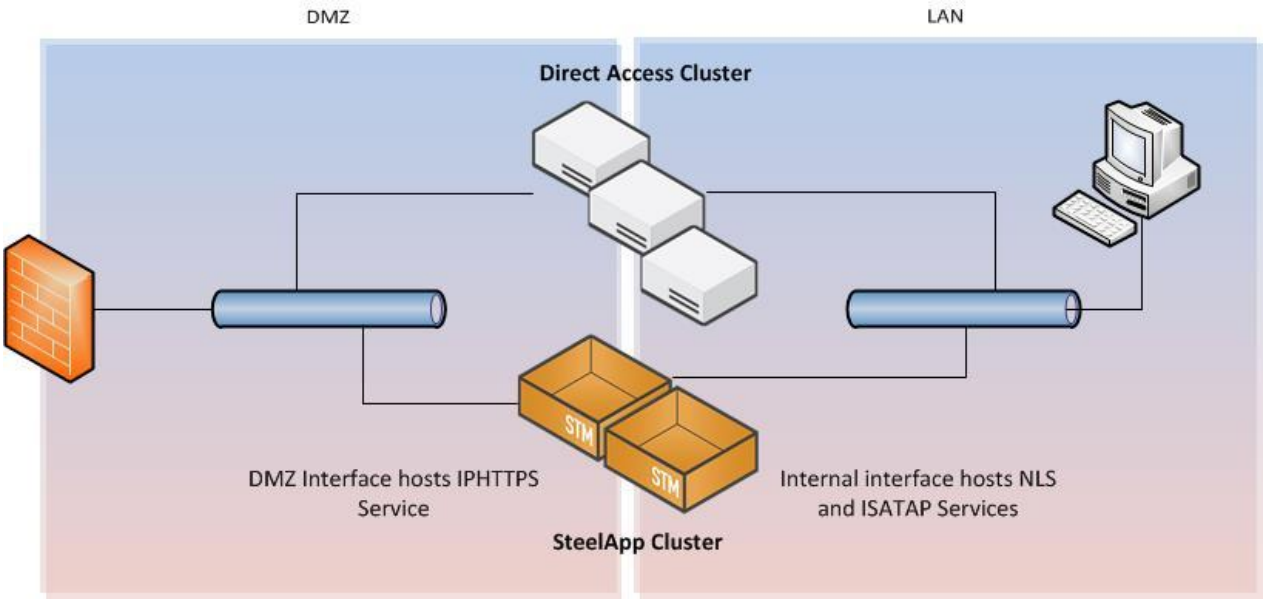
Direct Access clients make use of a Network Location Service to determine whether they are on the corporate LAN. These can be hosted on the Direct Access servers themselves, but Microsoft recommend deploying them on a separate HA service. Another option would be to host them on the SteelApp and respond to NLS requests using a simple TrafficScript. Instructions on how to

achieve this can be found in section 2.3.

1.3 Typical Deployment Scenarios

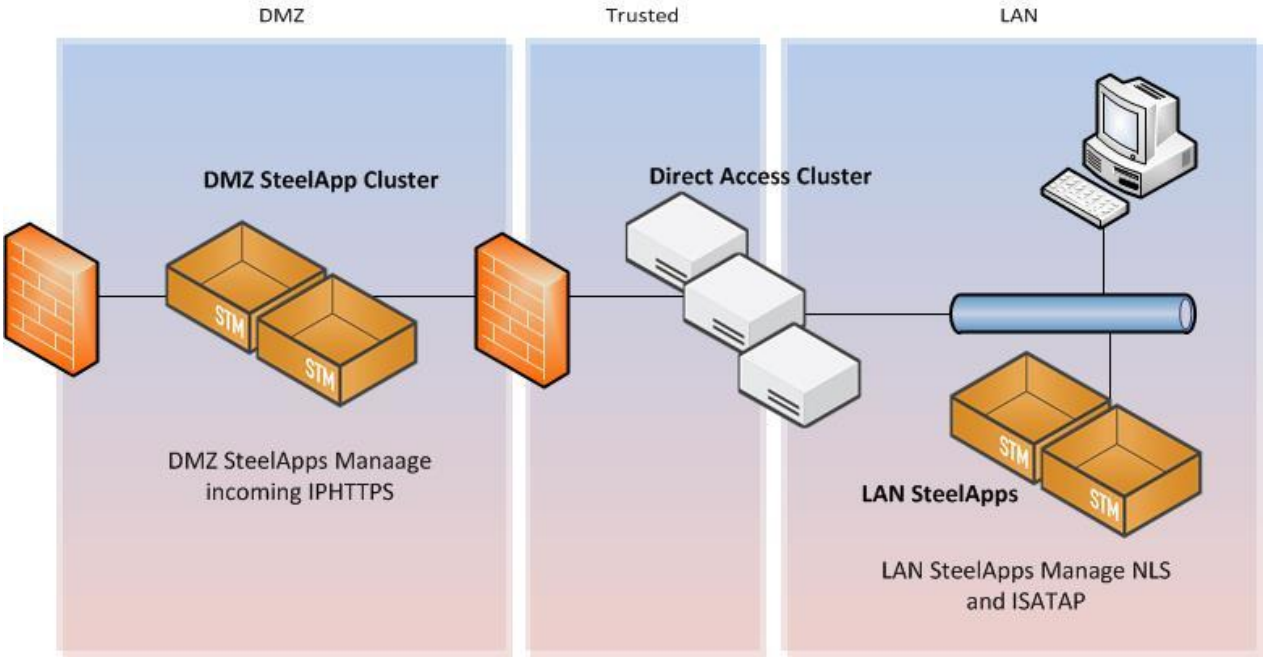
Option 1

Deploy a single SteelApp cluster to handle both the inbound IPHTTPS, and the outbound ISATAP traffic. You can also host the Network Location Service as an internal service.



Option 2

Deploy a SteelApp cluster in the DMZ to manage the inbound IPHTTPS traffic. A second cluster is deployed in the LAN to host the Network Location Service and provide ISATAP routing.



2.0 Direct Access Deployment

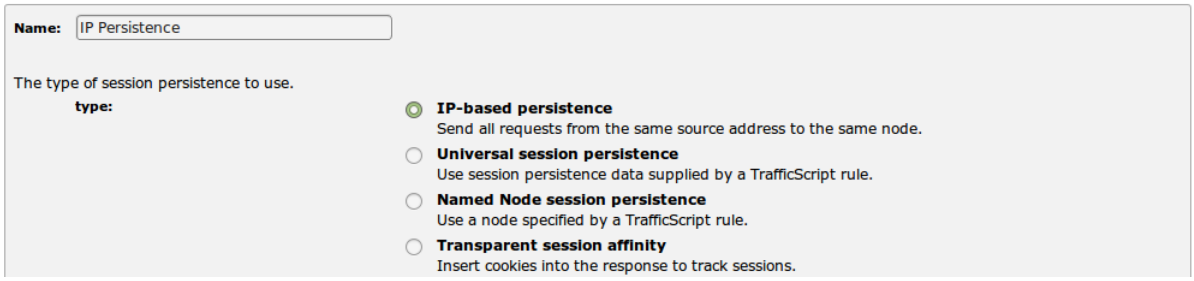
Direct Access supports two deployment models, either “Edge” or “Behind an edge device”. When deploying with SteelApp we recommend deploying in the “Behind an edge device” mode with either one or two NICs depending on your firewall zoning policy. When deployed behind an edge device, the DirectAccess server will not allow you to enable Teredo, due to the NAT restrictions mentioned earlier. Attempting to enable Teredo will result in an error.

2.1 Creating the IPHTTPS Service (INBOUND)

All client connections will use IPHTTPS which will require a simple SSL(HTTPS) Pass through service with an IP session persistence class.

Creating the IP Persistence class

Navigate to Catalogs -> Persistence and create a new persistence class called “IP Persistence”. Ensure that the persistence mode is “IP-Based persistence”, and update if necessary.



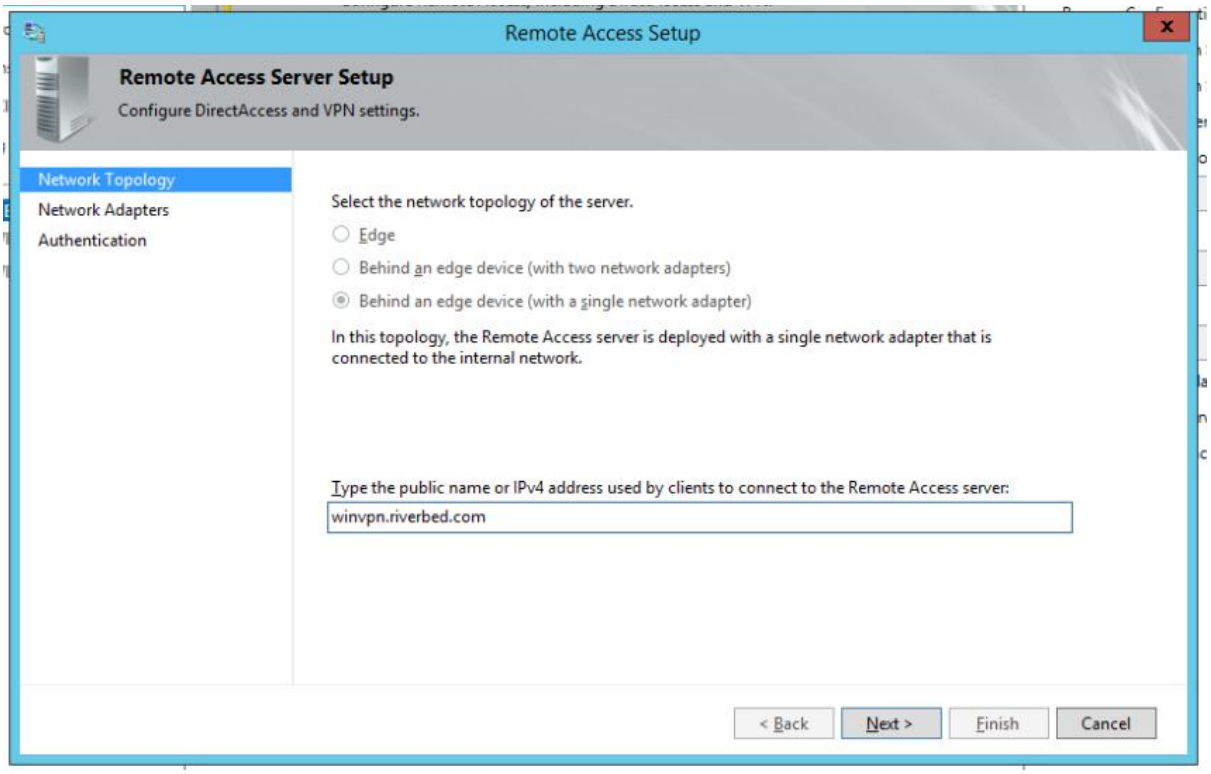
Creating the IPHTTPS Direct Access Pool

Navigate to Services -> Pools and complete the new Pool form: Use “DirectAccess-IPHTTPS” as the pool name, enter the nodes in <IP:Port> format separated by spaces, and select the “Simple HTTPS” health monitor.

Click the “Create Pool” button.
Now you should be on the pool edit page. Enter the Load Balancing section and change the Load Balancing algorithm to “Least connections”, click update.
Return to the pool page and then enter the “Session Persistence” settings. Select the “IP Persistence” persistence class we created earlier and then hit update.
Finally you will want to increase the “max_reply_time” timeout under “Connection Management” to something in the region of 300 seconds.

Creating the Traffic IP Group

When you setup Direct Access you are asked to provide a public IP or DNS hostname which the clients will use to connect to the Direct Access service. The end point for this address must be a Traffic IP Group on the SteelApp Traffic Manager.

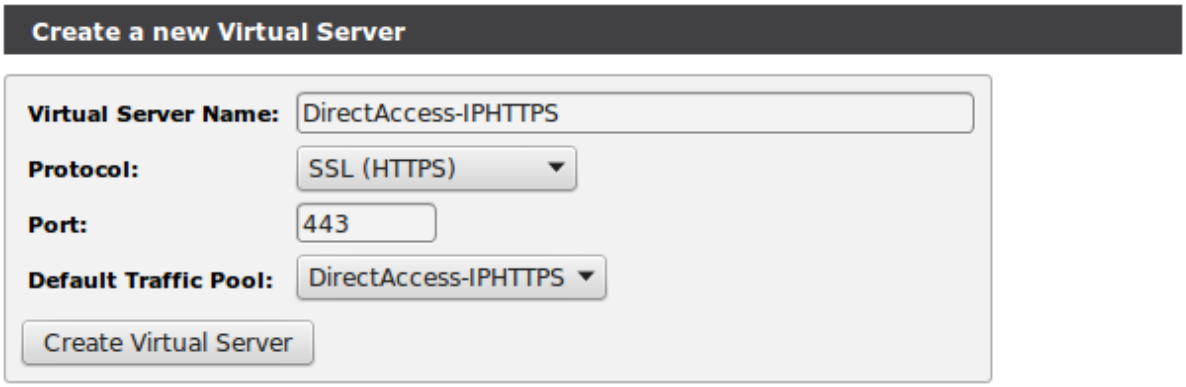


If you use a hostname then the hostname must resolve to a public address. The public address specified either needs to be used in the Traffic IP Group directly or if NAT'ed the translated IP should be used.

Navigate to Services -> Traffic IP Groups and create a new Traffic IP Group using this public address/endpoint. Call it "DirectAccess".

Creating the Virtual Server

The final step is to create the Virtual Server. Navigate to Services -> Virtual Server and use the form to create a new Virtual Server. Enter "DirectAccess-IPHTTPS" as the name, protocol "SSL (HTTPS)", Port 443, and Default Traffic Pool of "DirectAccess-IPHTTPS".



Click the "Create Virtual Server" button. The Virtual Server will be created and the Virtual Server edit page will be displayed. Under "Basic Settings" change the following options:

"Enabled" should be set to "Yes"
"Listening On" radio button to "Traffic IP Group" and select the "DirectAccess" Traffic IP Group. Click update at the bottom of the basic settings box.

Finally go into Connection Management -> Timeout Settings and modify the “timeout” value to match the “max_reply_time” value of the pool.

You now have the incoming IPHTTPS service available, and clients will be able to connect to the SteelApp and be load balanced across the Direct Access servers.

2.2 Creating the ISATAP Router Service (OUTBOUND: IPv4 LAN only)

Direct Access is an IPv6 technology and only assigns its clients an IPv6 address. If your internal LAN uses IPv4 and you want to use Manage-Out, then it is necessary to deploy ISATAP routers. When deployed as a single server, or when using Microsoft NLB for clustering, the DA servers can act as ISATAP routers. However when you elect to use an External Load Balancer (ELB), you are required to provide the ISATAP service elsewhere. The next steps will guide you through deploying an ISATAP router on SteelApp.

You only need to follow this part of the guide if your internal LAN uses IPv4 addressing and you want to use Manage-Out.

Gather IPv6 prefix information

On one of your DA servers, log in and execute the following command "Get-DA-Server". Record the "ClientIPv6Prefix" address specified. The first 48bits (3 x 16bit blocks) are your Unique Local Address (ULA) Prefix. The next 16 bits(4th block) make up the first 64bits and contain the common 59bit prefix used by all clients.

Each DA server will provide a different 64bit prefix to their clients. In order to see which prefixes are assigned by each server, you will need to log into each one and execute "ipconfig".

In the image above you can see that my IPHTTPS interface has the IPv6 address of "fdb4:3d09:fbee:1001:409f:f477:8732:a7a0". This means that clients accessing this server will use a 64 bit prefix of "fdb4:3d09:fbee:1001".

You should record the prefix information from each of your DA servers in a table. You will need to enter this into a script later.

Internal IP	IPv6 prefix	ULA	ClientNet
10.3.3.201	fdb4:3d09:fbee:1000::/64	fdb4:3d09:fbee	1000
10.3.3.202	fdb4:3d09:fbee:1001::/64	fdb4:3d09:fbee	1001

Creating the Traffic IP Group

Navigate to Services -> Traffic IP Groups and create a new Traffic IP Group using an internal IPv4 address. This address will be the IP for your ISATAP router.

You will need to publish this IP in internal DNS for the hostname of your ISATAP router. This is then usually pushed out via Group Policy to workstations which should be allowed to connect out to Direct Access clients.

Modify the ISATAP Script for your network

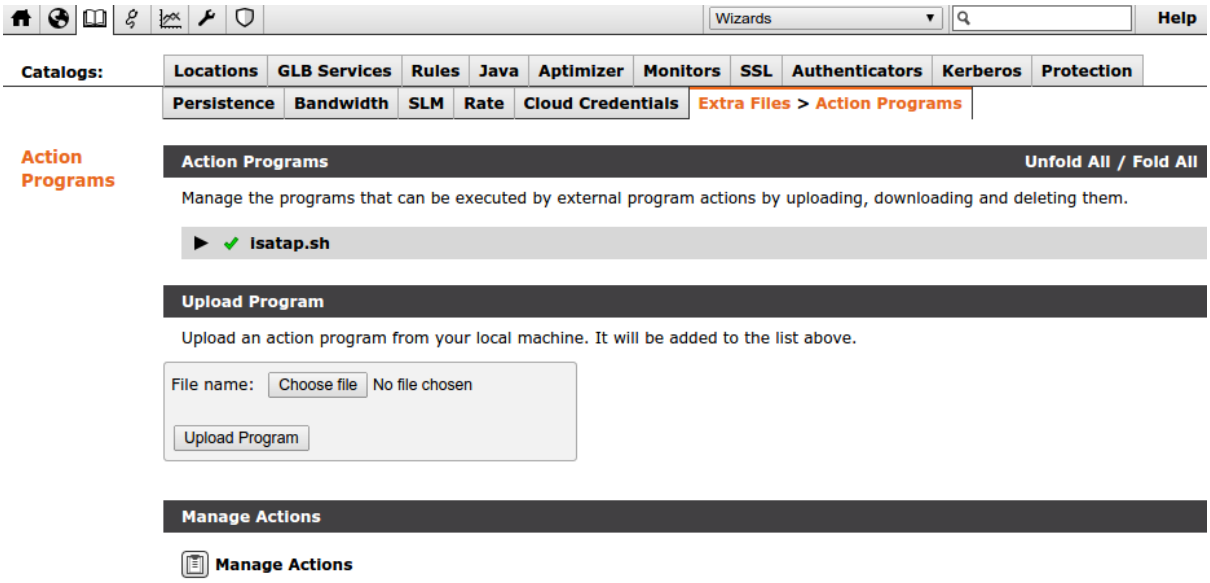
Copy the isatap.sh script from appendix 1 of this document into a new UNIX format text file.
Using the client ipv6 prefix information gathered earlier:

- Modify the ULAPrefix to match the one which your Direct Access servers are using.
- Modify the tip to match the Traffic IP Address used as your ISATAP router address
- Modify the daIP4 array to contain the internal IPv4 addresses of your Direct Access servers
- Modify the daClientNet to match the client networks prefix (minus the ULA).

Save the script as isatap.sh

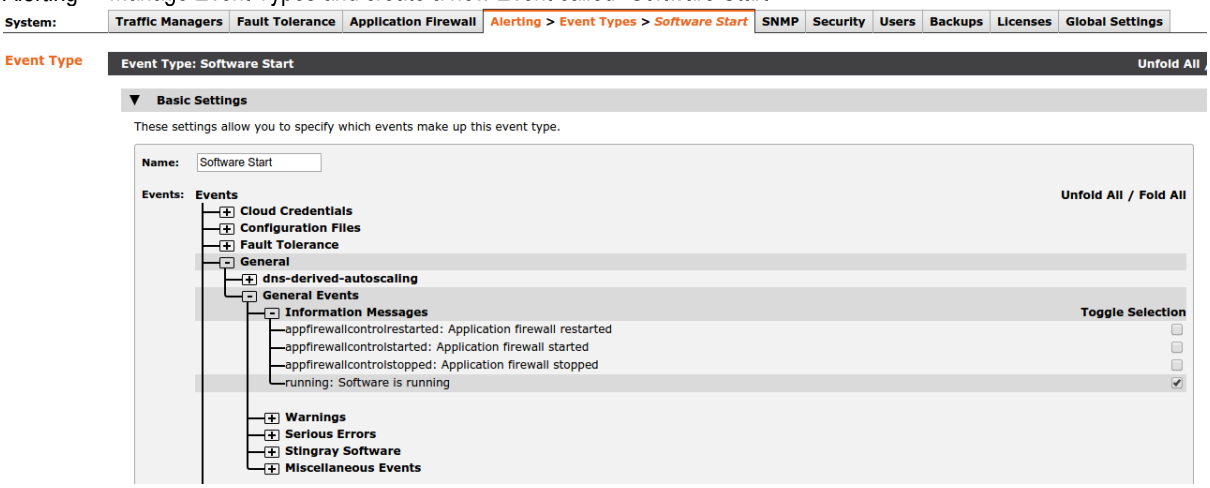
Upload the ISATAP script to SteelApp

Upload the modified script to the SteelApp in the Catalogs -> Extra Files -> Action Programs section



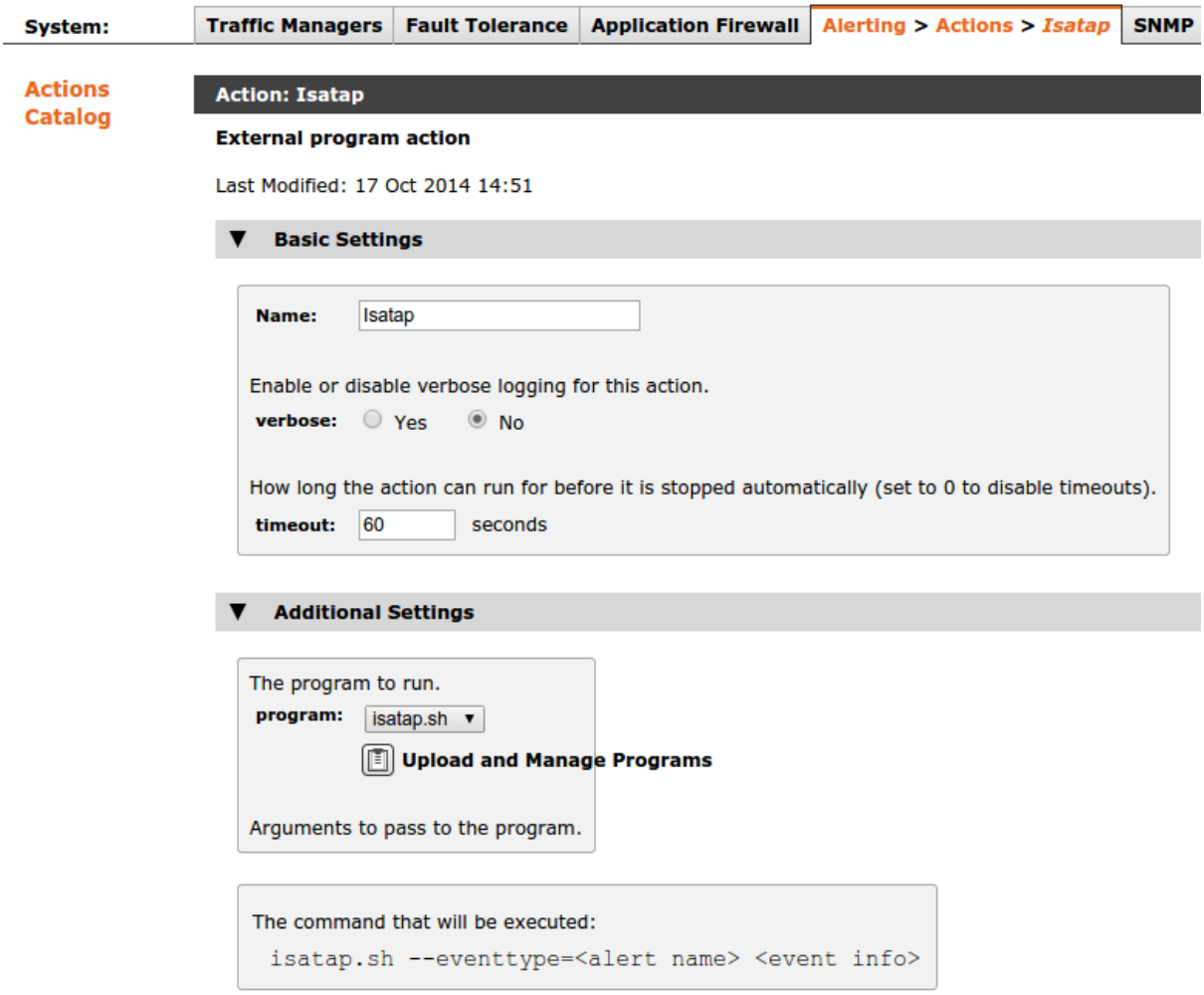
Create the Alert mapping

We're now going to create a start-up event which executes the script each time the SteelApp starts up. Navigate to System -> Alerting -> Manage Event Types and create a new Event called "Software Start"

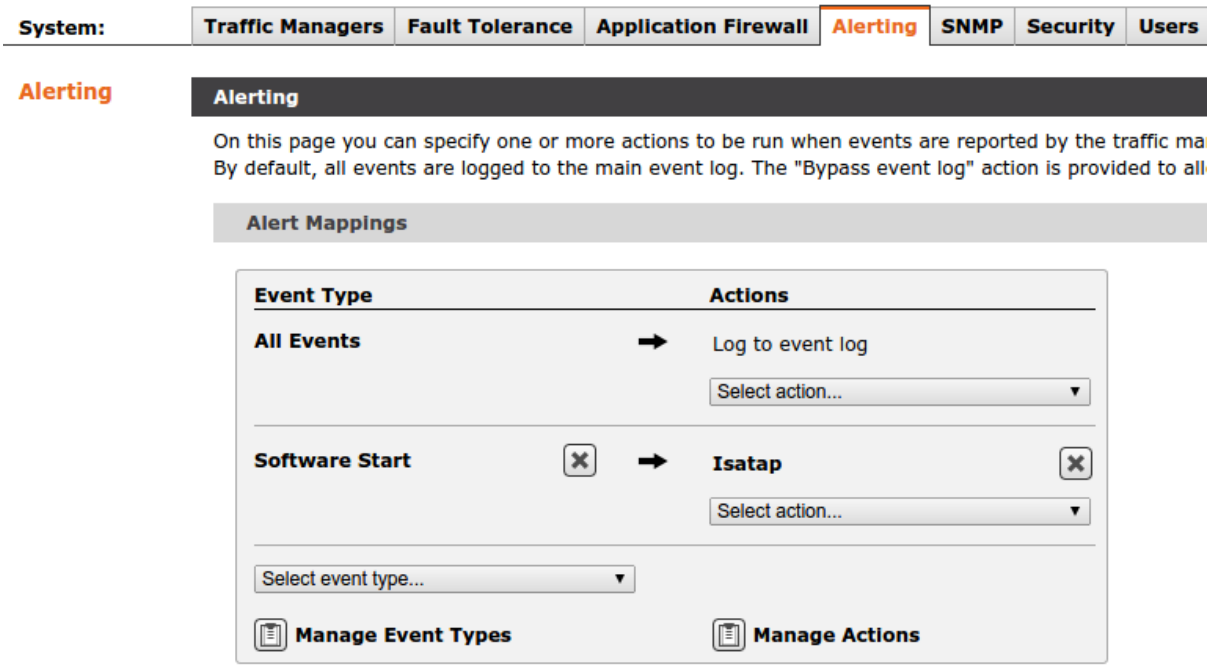


This event should be triggered on the "running: Software is running" event.

Go back to the Alerting tab and then "Manage Actions". Create a new Action called "Isatap" of type "Program". Select the isatap.sh script from the drop down box.

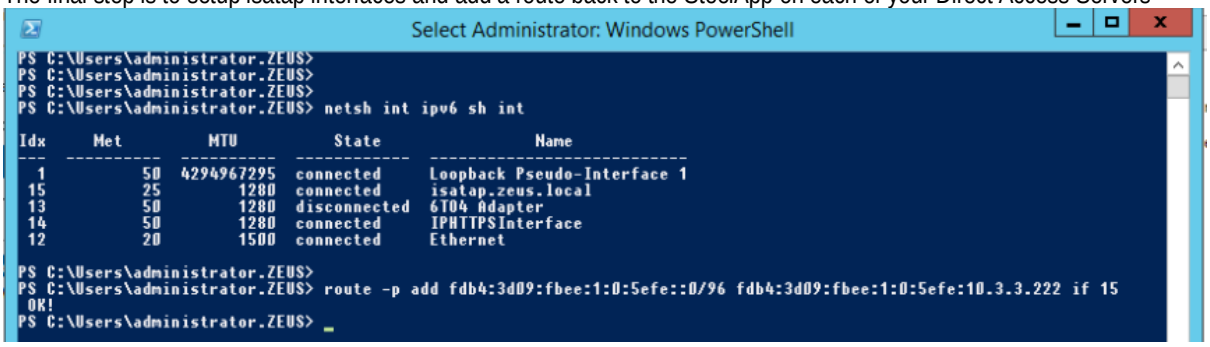


Finally navigate back to Alerting and set up a mapping between the “Software Start” event and the “Isatap” action.



Direct Access Server Routing

The final step is to setup isatap interfaces and add a route back to the SteelApp on each of your Direct Access Servers



Execute “netsh int ipv6 sh int” and make a note of your ISATAP interface number.

Next add a permanent route for ISATAP addresses in your ULA to go via the SteelApp TIP, and also ensure that the local ISATAP interface is forwarding but not advertising routes itself:

route –p add <ULA>:1:0:5efe::0/96 <ULA>1:0:5efe:<ipv4 TIP> if <interface no>
netsh int ipv6 set int <interface no> advertise=dis advertisedefaultroute=dis forwarding=en

That’s it. Each of your Traffic Managers will now be running an isatap interface and can be used to route ISATAP packets between your Manage-Out clients and the Direct Access servers.

2.3 Creating an NLS Responder (Optional)

The Network Location Service (NLS) needs to be Highly Available so that clients can quickly determine whether they are inside the corporate network or not. The Network Location Service is a simple HTTPS service which just needs to respond to client requests. First decide upon a FQDN and an IP address for your NLS service, and then complete the following steps.

Add the NLS host to DNS

Add a new A or AAAA record into DNS for the Direct Access NLS service. Eg directaccess-nls.<your internal domain>

Create a Traffic IP Group

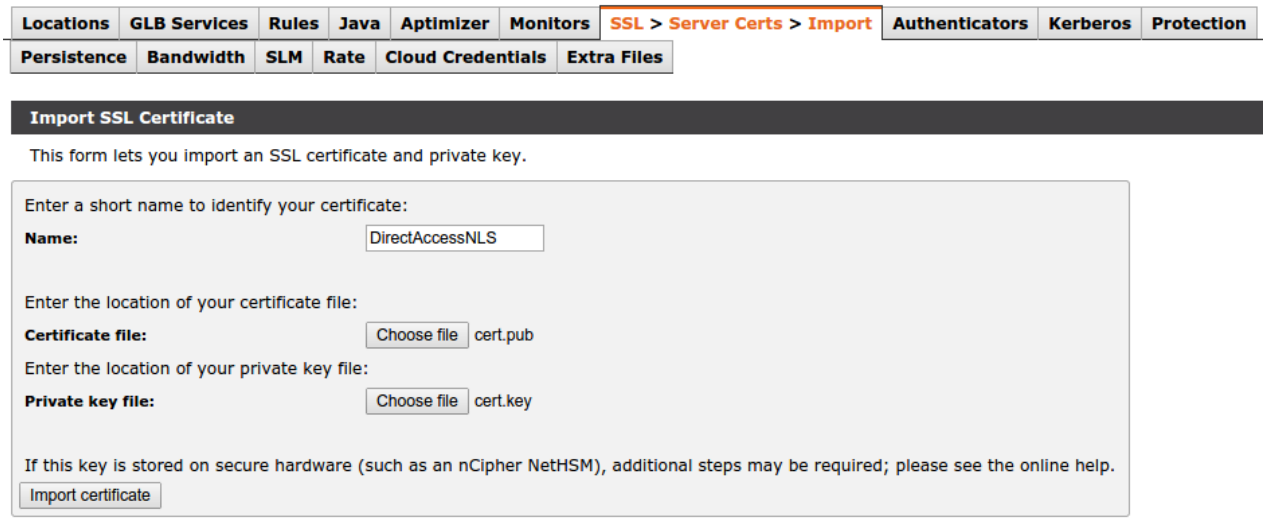
Follow the same process as covered previously to create a Traffic IP Group. This time the group should be named "DirectAccessNLS" and should be given the IP address which you entered into DNS in the previous step. Clients on the LAN need to be able to reach this IP address. It should not be reachable from outside of the LAN.

Upload a trusted certificate to the SteelApp

You will need to upload a trusted certificate to the SteelApp. This can be signed your internal Certificate Authority.

1. Create a new certificate for the DNS name you registered above and export it in PFX12 format
2. Use OpenSSL to extract the certificate and key from the PFX:
    a. openssl pkcs12 -in <YOUR PFX FILE> -nokeys -out cert.pub
    b. openssl pkcs12 -in da-nls.certs.pfx -nodes -nocerts | openssl rsa -out cert.key

3. Import the certificate files to Catalogs -> SSL -> SSL Server Certificates



Create the "NLS Response" TrafficScript Rule

Navigate to Catalogs -> Rules and created a new TrafficScript Rule called NLS Response. Enter the following into the rule box and click "update".

```
http.sendResponse("200 OK", "text/plain", "You are on the network", "");
```

Create a Virtual Server

Now navigate to Services -> Virtual Servers and create a new Virtual Server. You should call it DirectAccessNLS. The protocol should be set to HTTP (not HTTPS). The port should be changed to 443, and the pool can be set to discard. TrafficScript will be used to answer all requests to this service. Create the Virtual Server.

In the VirtualServer edit screen change the Listen Address to be the Traffic IP Group we created earlier. Set the service to

"Enabled" and click the update button.

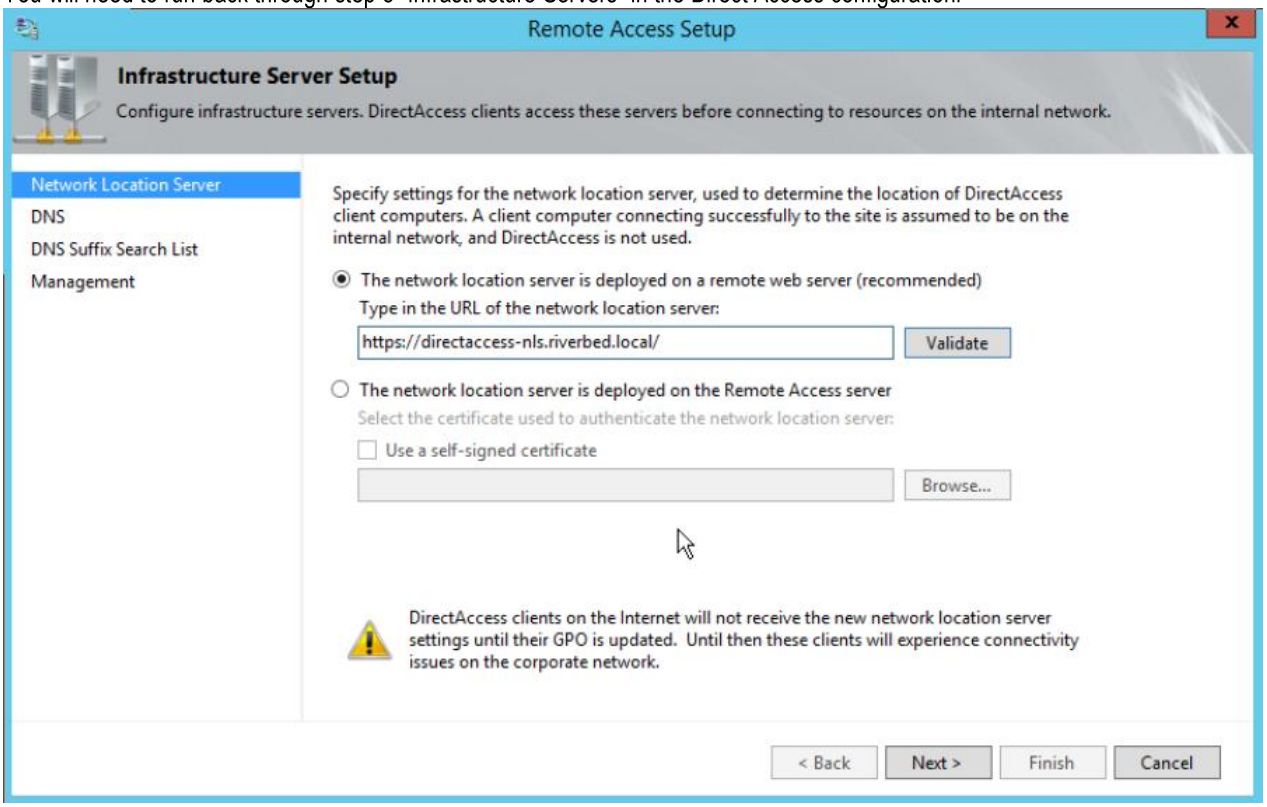
Scroll down to "SSL Decryption" and click edit. Change ssl_decrypt to "Yes", and select the certificate we imported from the list. Scroll to the bottom and click update.

Go back to the main Virtual Server page and into the Rules section. Under the Request Rules section add the "NLS Response" rule created earlier.

That's if for SteelApp. Now reconfigure Direct Access to use this service for NLS.

Modify the Direct Access NLS Configuration

You will need to run back through step 3 "Infrastructure Servers" in the Direct Access configuration.



Change the Network Location Server to "…deployed on a remote webserver…" and enter the https url for your NLS service in the box provided.

3.0 FAQ

3.1 My workstation has no ISATAP address configured?

ISATAP uses DNS to resolve the IP of the ISATAP router. All clients which you want to enable ISATAP for must be provided with this FQDN. Once you have created the Traffic IP Group for ISATAP, then you must add a DNS A record for that address. Do not use isatap.<your.local.domain> as this is blocked by default. Do not unblock this name, instead chose another and provide it only to workstations which need to use Manage-Out. Having ISATAP generally available is a bad idea, because IPv6 is preferred over IPv4, any clients with an ISATAP address will try to resolve DNS AAAA records and connect via ISATAP.

To set up a workstation for Manage-Out you will need to run the following commands on the workstation:

```
netsh int isatap set router <fqdn of ISATAP TIP>
net stop iphlpsvc
net start iphlpsvc
```

3.2 My Direct Access Server has no ISATAP gateway?

In order for the DirectAccess Server to pick up and use the SteelApp as its ISATAP gateway it must have “advertise” and “advertisedefaultroute” disabled on its ISATAP interface. In order to forward packets, it must have forwarding enabled on its ISATAP interface. Follow the instructions in the “Direct Access Server Routing” section above.

3.3 I can’t manage-out to remote clients?

This could be a number of things. You should check the following:

1. Follow the steps in 3.1 and 3.2 to ensure that the manage-out client and DA Servers have ISATAP configured correctly. A correctly configured ISATAP interface will have the same ULA prefix as used by the DA Servers IPHTTPS interface. It should end in 5efe:<Clients local IPv4 address>. The ISATAP interface should have a default gateway and it should end in 5efe:<STMs ISATAP TIP>. If the address on the ISATAP interface begins “fdb4:3d09:fbee” then you didn’t modify the isatap script before uploading it to the STM.
2. Check that you have added the permanent route for ISATAP addresses on your Direct Access servers.
3. If you are connecting to the client using it’s name, then check that the DNS servers AAAA record is up-to-date and matches the address that the client has on its IPHTTPS interface.
4. Is the client listening for the requests you’re sending? Eg: Are you trying to initiate a remote desktop session? Does the client have remote access enabled?
5. Is the traffic being dropped by a firewall on the Direct Access server, or the client?
6. If it’s still not working, try taking a tcpdump on the steelapp to determine where the connection is failing.

APPENDIX 1

```
#!/bin/bash

# Common Private ULA prefix - get this from the DA server.
ULAprefix=fdb4:3d09:fbee

# IPv4 Traffic IP used as ISATAP address
tip=10.3.3.222

# Array of Direct Access servers IPv4 addresses and
# Array of Direct Access Client nets (same order as daIP4)
daIP4=( 10.3.3.201 10.3.3.202 )
daClientNet=( 1000 1001 )

startup() {
  sysctl net.ipv6.conf.all.forwarding=1

  ip tunnel add is0 mode isatap local $tip ttl 64
  ip link set is0 up
  ip addr add ${ULAprefix}:1:0:5efe:${tip}/64 dev is0

  tmp=$(mktemp -d)
  cat > ${tmp}/radvd.conf <<-  EOF
    interface is0
    {
      AdvSendAdvert on;
      UnicastOnly on;
      AdvHomeAgentFlag off;
      prefix ${ULAprefix}:1::0/64
      {
        AdvOnLink on;
        AdvAutonomous on;
        AdvRouterAddr off;
      };
    };
EOF

  radvd -C ${tmp}/radvd.conf
  for (( i=0 ; i< ${#daIP4[@]} ; i++ ));
  do
    ip -6 route add ${ULAprefix}:${daClientNet[$i]}::0/64 via
${ULAprefix}:1::5efe:${daIP4[$i]} dev is0
    ip tunnel prl prl-nodefault ${daIP4[$i]} dev is0
  done
}

shutdown() {
  killall -q radvd
  sysctl net.ipv6.conf.all.forwarding=0
  if $( ip link show is0 >/dev/null 2>&1 )
  then
    ip link set is0 down
    ip tunnel del is0
  fi
}
```

```
case $1 in
start)
  echo "Started from init"
  startup
  ;;
--eventtype=*)
  echo "Started by SteelApp event"
  shutdown
  startup
  ;;
stop)
  echo "Stopped from init"
  shutdown
  ;;
*)
  echo "Usage: $0 (start|stop)"
  ;;
esac
```

Changes in Current Solution Guide

Number	Description	Date
1.0	Initial Release	2014-10-31

About Riverbed

Riverbed delivers performance for the globally connected enterprise. With Riverbed, enterprises can successfully and intelligently implement strategic initiatives such as virtualization, consolidation, cloud computing, and disaster recovery without fear of compromising performance. By giving enterprises the platform they need to understand, optimize and consolidate their IT, Riverbed helps enterprises to build a fast, fluid and dynamic IT architecture that aligns with the business needs of the organization. Additional information about Riverbed (NASDAQ: RVBD) is available at www.riverbed.com.

Riverbed Technology, Inc.
680 Folsom Street
San Francisco, CA 94105
Tel: (415) 247-8800
www.riverbed.com

Riverbed Technology Ltd.
One Thames Valley
Wokingham Road, Level 2
Bracknell. RG42 1NG
United Kingdom
Tel: +44 1344 31 7100

Riverbed Technology Pte. Ltd.
391A Orchard Road #22-06/10
Ngee Ann City Tower A
Singapore 238873
Tel: +65 6508-7400

Riverbed Technology K.K.
Shiba-Koen Plaza Building 9F
3-6-9, Shiba, Minato-ku
Tokyo, Japan 105-0014
Tel: +81 3 5419 1990