EXOS ezvxlan.py Application

ezvxlan.py 1.0.0.6

For Extreme customers deploying a virutal network using VXLAN, this application provides an automatic mapping of certain VLANS into VXLAN VNIs when used with the EXOS virtual-network capability available on:

- X670-G2
- X770

This application works best when combined with Extreme Management Center. http://www.extremenetworks.com/product/management-center/

Requirements

- ExtremeXOS 21.1.1.4 or later
- ExtremeSwitch X670-G2
- ExtremeSwitch X770

Fixes in 1.0.0.6

- after a reboot, may not create all VNI’s for eligable VLANs
- attempts to create VNI’s during a switch ‘save’ operation would not work

- general performance improvements to support large numbers of VNI/VLANs

Application Highlights:

- Monitors VLAN/port additions/deletion
- Automatically creates VXLAN VNIs when VLANs are created with a specific name format. VNI taken from VLAN name.
- Automatically creates VXLAN VNIs when vm-tracking creates dynamic VLANs. VNI=VLAN tag
- VNI is created when first port is added to a VLAN to avoid VXLAN flooding to endpoints without assigned ports
- VNI is deleted when last port is removed from a VLAN
- VNI is deleted when entire VLAN is deleted
- If OSPF router ID is configured when ezvxlan.py is started, the Local VTEP (LTEP) with the OSPF router ID is created, unless the switch is an MLAG peer.
- If the switch is an MLAG peer, the user must create the same VLAN with the same IP address on each MLAG peer and manually configure the LTEP IP with that VLAN IP address.
- Enables the OSPF extensions, if not already enabled, when the first VNI is created
- Runs on VXLAN capable switched running EXOS 21.1 or later

VLAN Names

When ezvxlan.py is running on an ExtremSwitch running EXOS, creating a VLAN with a certain name format will automatically create the VXLAN VTEP and matching VNI.

Two VLAN name formats will cause ezvxlan.py to automatically create a VXLAN VNI.

- VNI_{vni}{text} - Manually created VLAN by user
- SYS_VLAN_xxxx - dynamic VLAN created by EXOS such as vm-tracking

VNI-{vni}{text} or VNI_{vni}{text} VLAN name format:

A VLAN name can be created manually via EXOS CLI, via SNMP or any other EXOS management interface.
The VLAN name is in the form VNI-{vni}{text} or VNI_{vni}{text} where:

- VNI - VLAN name must start with capital VNI
- a -(dash) or _(underscore) separator character
- {vni} is any number from 1-{upper VNI value}
- {text} is any additional text that describes the VLAN. The text may not start with a number.

E.g. Below are EXOS CLI commands used to create VLAN names that match the ezvxlan.py naming pattern.

- create vlan VNI-10012_vm9037 tag 100

    - ezvxlan.py will look for VNI-10012_ and then create a VXLAN VNI with 10012. The VLAN VID is 100 and is indepentent of the name.
- create vlan VNI_10012remoteOffice tag 203
    - ezvxlan.py will create VXLAN VNI 10012 and attach VLAN VID 203

SYS_VLAN_xxxx VLAN name format:

EXOS features such as vm-tracking with dynamic detection enabled

- will received a MAC address from a port
- authenticate the MAC address
- create a MAC based VLAN with the name SYS_VLAN_xxxx where xxxx is the VLAN ID (VID).

ezvxlan.py detects VLANs created with the SYS_VLAN_xxxx name and automatically creates a VXLAN VNI with the same xxxx number.

E.g. VLAN SYS_VLAN_1010 will map to VXLAN VNI 1010. ezvxlan.py creates a VXLAN VNI name of SYS_VN_1010.

VXLAN VNI Creation

The VLXAN VNI will actually be created when the first port is added to the VLAN and will be
deleted after the last port is removed from the VLAN. By requiring an

actual port within the VLAN before adding/deleting the VXLAN VNI, network traffic/flooding will not be sent to a switch that has no ports associated with the attached VLAN.

VTEP Identifier

On startup, the VLXAN VTEP is created using the OSPF router id for the IP address.
The EXOS CLI command should be used to configure an OSPF router id *before* starting ezvxlan.

- configure ospf routerid {ipAddress}
    - Example: configure ospf routerid 10.10.10.1

ezvxlan.py will use the OSPF routerid as the local VTEP address.

If the OSPF router ID is configured after ezvxlan.py is running, restart ezvxlan.py using the restart CLI command:

- run script ezvxlan.py restart

Files

- EXOS Switch summitX-ezvxlan-1.0.0.6.xmod
- README.md

ezvxlan.py Usage

ezvxlan.py Getting help

```
# run script ezvxlan.py -h
```

```
usage: ezvxlan.py [-h] [-d] {start,stop,restart,show} ...

positional arguments:
  {start,stop,restart,show}
    start                Start the ezvxlan.py application
    stop                 Stop the ezvxlan.py application
    restart              Restart the ezvxlan.py applicatio
n. Useful after upgrade
    show                 Show the running status of ezvxla
n.py.

optional arguments:
  -h, --help             show this help message and exit
  -d, --debug            Enable debug
```

ezvxlan.py start help

```
# run script ezvxlan.py start -h
usage: ezvxlan.py start [-h] [-p PORT]

optional arguments:
  -h, --help             show this help message and exit
  -p PORT, --port PORT  Controller port. Always add this
```

```
port when VXLAN VLANs
                        are created
```

ezvxlan.py stop help

```
# run script ezvxlan.py stop -h
usage: ezvxlan.py stop [-h] [-k]

optional arguments:
  -h, --help  show this help message and exit
  -k, --keep  Keep automatically created VXLAN VNIs with
names that start with
              SYS_VN_
```

ezvxlan.py restart help

```
# run script ezvxlan.py restart -h
usage: ezvxlan.py restart [-h] [-p PORT]

optional arguments:
  -h, --help            show this help message and exit
  -p PORT, --port PORT  Controller port. Always add this
port when VXLAN VLANs
                        are created
```

ezvxlan.py start

Before starting ezvxlan.py, configure the OSPF router ID and enable

OSPF. Having this information available at the start, ezvxlan.py will automatically create the VXLAN LTEP.

Example:

- configure ospf routerid 10.10.10.1
- enable ospf

To start ezvxlan, enter the CLI command:

```
# run script ezvxlan.py start
```

```
Starting ezvxlan.py
```

When ezvxlan.py first starts, it:

- enables OSPF vxlan extensions
- Creates and configures the VTEP based on the OSPF routerid (unless running in an MLAG configuraiton)
- scans all existing VLANs looking for matching names
- if a matching VLAN name is found, and the VLAN has at least one port assigned, the VXLAN VNI is created with the SYS_VN_{vni}

After the initial VLAN scan, ezvxlan.py continues to run in the background monitoring VLAN creation/deletion/port adds/port deletes.

It is only necessary to start ezvxlan.py once. If the switch is rebooted, ezvxlan.py will automatically be restarted.

ezvxlan.py stop [-k | --keep]

To stop ezvxlan.py, enter the CLI command:

```
# run script ezvxlan.py stop [-k | --keep]
```

By default, when ezvxlan.py is stopped, it will delete any automatically created VXLAN VNI.

```
Stopping ezvxlan.py
Deleting VXLAN VNI names starting with SYS_VN_
```

If you wish to leave the VXLAN VNI in place but no longer wish ezvxlan.py to monitor VLAN adds/deletes, specify the -k or --keep option in the stop command. This will keep any SYS_VN_{vni} entries that have already been created.

```
Stopping ezvxlan.py
Keeping VXLAN VNI names starting with SYS_VN_
```

ezvxlan.py restart

```
# run script ezvxlan.py restart
```

```
Stopping ezvxlan
```

```
Keeping VXLAN VNI names starting with SYS_VN_
Starting ezvxlan
```

The restart command is a convenient way to stop -k then start the ezvxlan.py application. This is useful after downloading a new version of ezvxlan.py

ezvxlan.py show

```
# run script ezvxlan.py show
```

The show option displays the running status of the ezvxlan.py applications.

If ezvxlan.py is running

```
ezvxlan.py Version: 1.0.0.6        process is running
VLANs with names SYS_VLAN_xxxx or VNI_{vni}{text} are aut
omatically mapped to SYS_VN_{vni} VTEPs
```

If ezvxlan.py is not running

```
ezvxlan Version: 1.0.0.6        process is not running
VLANs with names SYS_VLAN_xxxx or VNI_{vni}{text} are not
 mapped to SYS_VN_{vni} VTEPs automatically
```

Download

EXOS offers a variety of download methods. All of the methods below assume the EXOS switch has been configured with an IP address either on the `mgmt` VLAN (for the management port) or `default` VLAN (for the front panel ports).

Download over tftp

To download summitX-ezvxlan-1.0.0.6.xmod to an EXOS switch running ExtremeXOS 21.1 or later, place the file in a server tftp directory.

Download tftp over management port

Enter the EXOS CLI command:

- download image *serverIP* summitX-ezvxlan-1.0.0.6.xmod

E.g.
`download image 10.10.10.1 summitX-ezvxlan-1.0.0.6.xmod`

Download tftp over front panel port

Enter the EXOS CLI command:

- download image *serverIP* summitX-ezvxlan-1.0.0.6.xmod vr VR-Default

E.g.

`download image 10.10.10.1 summitX-ezvxlan-1.0.0.6.xmod vr VR-Default`

Download over http

EXOS can download files from a web site using http.

If your server does not have a web server and Python is installed, Python offers a simple HTTP web server. Python Simple Web Server

Example starting a simple python web server on port 8000

```
cd <directory>
python -m SimpleHTTPServer 8000
```

Copy summitX-ezvxlan-1.0.0.6.xmod to *directory* used in the example above.

Download http over management port

Enter the EXOS CLI command:

- download url http://*serverIP*/summitX-ezvxlan-1.0.0.6.xmod

E.g. `download url http://10.10.10.1/summitX-ezvxlan-1.0.0.6.xmod`

Download http over front panel port

Enter the EXOS CLI command:

- download url http://*serverIP*/summitX-ezvxlan-1.0.0.6.xmod vr VR-Default

E.g. `download url http://10.10.10.1/summitX-ezvxlan-1.0.0.6.xmod vr VR-Default`

Download using EXOS web (Chalet)

- Using your browser, download summitX-ezvxlan-1.0.0.6.xmod from github to your PC.
- Then using the EXOS web interface (Chalet), navigate to Apps->File Manager.
- Use: `Upload files from Local Drive:` to upload and install the file to the EXOS switch

License

Support

The software is provided as is and Extreme Networks has no obligation to provide
maintenance, support, updates, enhancements or modifications.
Any support provided by Extreme Networks is at its sole discretion.

Issues and/or bug fixes may be reported on The Hub.

> *Be Extreme*