Utilities have developed critical infrastructure over many decades and generations of data communications technologies, and today, they are modernizing their networks to support end-to-end packet-based communications. Some of the business drivers for these investments are:

- Integration of distributed energy resources (DERs)
- Retirement of expensive legacy architectures like SONET
- Increasing capacity required by new applications like HD video and new sources of telemetry
- Reduction of OpEx through automation, orchestration, and management visibility

To enable customers to take full advantage of the economies of scale of underlay networks with the ability to overlay layers of SDN and Network Functions Virtualization (NFV), vendors must provide an end-to-end transport solution. Juniper Networks’ strategy of disaggregating security and routing software from the underlying hardware represents a great advantage for utilities. Utilities can utilize any x86 platform to install NFV with unprecedented agility.

This solution brief describes the process by which a Juniper Networks® vSRX Virtual Firewall can be installed on an IEEE 1613 Class 2 compliant device. In this case, the x86 server of choice is the SEL-3355 Rack-Mounted Rugged Computer, with a focus on the setup and installation of virtualized routing and firewall functions. Support for legacy interfaces will be addressed in ensuing versions of this brief.

**The Challenge**

Two major trends in substation modernization are driving the migration off legacy time-division multiplexing (TDM)-based communications solutions, which are expensive to operate and lack management visibility and scale. The challenge is to provide compute resources for applications ranging from local authentication to network anomaly detection, with many other distributed automation and data aggregation applications in between. Using today’s technologies, deploying the new substation “stack” of applications requires multiple fixed configuration devices, rack-mounted units, and associated power interfaces. In addition to sparing for each device, a utility’s operational technology–information technology (OT–IT) teams must maintain, staff, and train on multiple element management systems to maintain critical operation service-level agreements (SLAs).

**The Juniper Networks Virtualized Routing and Security Solution**

Juniper Networks, a leader in NFV, gives OT–IT staffs the option to maintain routing instances and NERC Critical Infrastructure Protection (CIP)-compliant firewalls on a common x86 platform certified to IEEE 1613 Class 2 specifications. Additionally, a single pair of servers replaces the footprint of fixed configuration devices, while providing additional guest virtual machines (VMs) that other applications need to maintain the modern substation.
Solution Brief

Using Juniper Virtualized Routing and Security at Utility Substations

Solution Components

The overall goal of this design exercise is to provide an environment that would represent, as accurately as possible, a production network that employs the types of elements mentioned. This foundation allows a building block to add other automation components and test concepts.

In this environment, the Ubuntu host operating system is installed on a dedicated server. Note that the operating system/kernel-based virtual machine (KVM) could be installed as a nested application within a virtual box or VMware, but results may vary. This installation focuses on the use of KVM as the hypervisor in a typical production network.

The test KVM environment consists of three separate networks: Management, Trusted, and Untrusted.

<table>
<thead>
<tr>
<th>Network</th>
<th>vSRX_IP</th>
<th>vSRX_INT</th>
<th>KVM_INT</th>
<th>KVM_IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Untrusted</td>
<td>172.16.1.45/24</td>
<td>GE-0/0/1</td>
<td>virbr1</td>
<td>172.16.1.1</td>
</tr>
<tr>
<td>Trusted</td>
<td>192.168.122.45/24</td>
<td>GE-0/0/0</td>
<td>virbr0</td>
<td>192.168.122.1</td>
</tr>
<tr>
<td>Management</td>
<td>10.0.41.45/24</td>
<td>FXP-0/0/0</td>
<td>KVM bridge NIC</td>
<td>10.0.41.45</td>
</tr>
</tbody>
</table>

KVM Virtual Machine Manager Installation

Within Ubuntu, you must first verify virtualization support by entering the following command:

`egrep -c '(svm|vmx)' /proc/cpuinfo`

A value of 0 indicates no support for hardware virtualization. A value of 1 or greater indicates virtualization support. The qemu-system software (see below) may need to be installed for virtualization support.

```
Note that this installation focuses on the use of GNOME desktop. KVM running on Ubuntu server is managed via the command line and is out of scope for this document.
```

Open a terminal window and update package listing:

`sudo apt-get update`

Install the KVM application and required software:

`sudo apt-get install qemu-kvm libvirt-bin bridge-utils virt-manager qemu-system`

Hosts attached to the Untrusted or Trusted networks are assigned an IP address from the vSRX Virtual Firewall. The vSRX also is the default gateway for these networks and forces all host traffic through the firewall. The vSRX's default gateway is the address on the Trusted port address of the hypervisor, as this provides Network Address Translation (NAT) services to allow Internet access. Automation software, such as Juniper Networks Junos operating system (Junos OS)-PythonEZ (PyEZ), is commonly installed on a system that is connected to the Trusted network of the vSRX, as appropriate security policies can then be used.

Ubuntu Desktop v16.04: [https://www.ubuntu.com/download](https://www.ubuntu.com/download)

vSRX 15.1X49-D50.3: [http://www.juniper.net/support/downloads/?p=vsrx#sw](http://www.juniper.net/support/downloads/?p=vsrx#sw)

Root and users within the libvirtd group have access to virtual machines within KVM. Add additional existing users to libvirtd via the following command: `sudo adduser <username> libvirtd`

Completely log out of the terminal and the desktop application, and log back in to the terminal.

Issuing the following command should show a list of empty virtual machines if everything is working correctly:

`virsh --c qemu:///system list`
Click on the upper left button to search for Virtual Machine Manager. The application should be in your dashboard now.

Create a new virtual machine by clicking the button on the dashboard. The application will walk you through the creation of the new VM and required parameters.

Once the image has been successfully created, set the machine processor settings to the following values.

Select Copy host CPU configuration.

Select the appropriate Disk bus under Advanced options. The storage format should already be set to qcow2.

Copy the installation media-vsrx-vmdisk-15.1X49-D50.3.qcow2 to a directory where you plan to run the virtual machine. Make sure the appropriate file and folder permissions are set properly to allow access during installation. We will be directly importing the .qcow2 image.

Create the vSRX virtual machine by setting the following values.

Provide the local path to the installation image and set the following OS type and version.

Select forward and set the following values:

* Set the disk image size to 16 GB.
* Set the memory to 4096 and CPUs to 2.

Select the NIC. Note: The first network information collector (NIC) listed will be the vSRX FXPO management interface. Add two additional NICs representing both the Trusted and Untrusted networks. These additional interfaces will be listed as GE-0/0/0 and GE-0/0/1 on the vSRX.

Add additional interfaces as required at this time.

Note: This installation bridges the FXPO management interface to the NIC that is directly attached to the physical management network on the host.

The vSRX FXPO interface is assigned an address that resides on the same physical LAN as the NIC in the Ubuntu server. As noted by the (!), the hosts that reside on that physical LAN will have access to the SSH, Telnet, or HTTP protocols on the vSRX management interface, but the host itself (Ubuntu in this case) will not. Console access is still available within the KVM.
Remove any devices that are not required by the vSRX, such as sound.

The vSRX Trusted and Untrusted networks are represented by virtual bridge interfaces. Subnets and Dynamic Host Configuration Protocol (DHCP) information are provisioned according to the virbr0 and virbr1 network connections. This information can be viewed by clicking in the upper left of the desktop and searching for “network connections.”

The address on virbr0 is currently the default gateway for devices on the virbr0 network and provides DHCP functionality. We will move these services to the vSRX. In this example, GE-0/0/0 is assigned as a Trusted interface on this network. GE-0/0/1 is assigned as an Untrusted interface on virbr1.

Right click on QEMU/KVM details.
The Virtual Networks tab allows you to add virtual networks and relative IPv4, DHCP, etc., settings on KVM.

An Ubuntu server was set up on the Untrusted network segment and its interface was provisioned to obtain an address via DHCP.
At this point, the bridge virtual interface is configured to be the DHCP server and default router for the Untrusted segment.
We would like all Untrusted and Trusted traffic to go through and be managed by the vSRX. We will configure the vSRX to be the DHCP server and default router for the network segments.

Remove the DHCP service from the Untrusted and Trusted networks (if present) on the KVM virtual bridge interface using the virsh editor.
sudo virsh net-edit Untrust_Network

Remove the DHCP section. Save the file and open the QEMU/KVM Connection Details tab.

Stop the network interface and restart it. The DHCP range should now be Disabled.

Enable DHCP services on the vSRX for both the Trusted and Untrusted networks. Restart KVM and the vSRX.

vSRX Configuration

Following is the configuration for the packet mode vSRX described in this document.

```sh
set version 15.1X49-D50.3
set system host-name vSRX_KVM
set system root-authentication encrypted-password "$5$gjzMbfx0STCY2bmgElgdTVGIWn/ZPp5x/OGyQaiCga1YWubTxy0"
set system name-server 192.168.0.1
set system services ssh
set system services netconf ssh
set system services web-management http interface fxp0.0
set system services dhcp pool 172.16.1.0/24 address-range low 172.16.1.100
set system services dhcp pool 172.16.1.0/24 address-range high 172.16.1.120
set system services dhcp pool 172.16.1.0/24 domain-name untrust.net
set system services dhcp pool 172.16.1.0/24 router 172.16.1.45
set system services dhcp pool 172.16.1.0/24 server-identifier 172.16.1.45
set system services dhcp pool 192.168.122.0/24 address-range low 192.168.122.100
set system services dhcp pool 192.168.122.0/24 address-range high 192.168.122.120
set system services dhcp pool 192.168.122.0/24 name-server 192.168.0.1
set system services dhcp pool 192.168.122.0/24 router 192.168.122.45
set system services dhcp pool 192.168.122.0/24 server-identifier 192.168.122.45
set system syslog user * any emergency
set system syslog file messages any any
set system syslog file messages authorization info
set system syslog file interactive-commands interactive-commands any
set system license autoupdate url https://ae1.juniper.net/junos/key_retrieval
set security screen ids-option untrust-screen icmp ping-death
set security screen ids-option untrust-screen ip source-route-option
set security screen ids-option untrust-screen ip tear-drop
set security screen ids-option untrust-screen tcp syn-flood alarm-threshold 1024
set security screen ids-option untrust-screen tcp syn-flood attack-threshold 200
set security screen ids-option untrust-screen tcp syn-flood source-threshold 1024
set security screen ids-option untrust-screen tcp syn-flood destination-threshold 2048
set security screen ids-option untrust-screen tcp syn-flood queue-size 2000
set security screen ids-option untrust-screen tcp syn-flood timeout 20
set security screen ids-option untrust-screen tcp land
set security nat source rule-set Untrust_Rule from zone untrust
set security nat source rule-set Untrust_Rule to zone trust
set security nat source rule-set Untrust_Rule rule Untrust_Rule match source-address any
set security nat source rule-set Untrust_Rule rule Untrust_Rule match destination-address any
set security policies from-zone trust to-zone trust policy default-permit match source-address any
set security policies from-zone trust to-zone trust policy default-permit match destination-address any
set security policies from-zone trust to-zone trust policy default-permit match application any
set security policies from-zone trust to-zone trust policy default-permit application any
set security policies from-zone trust to-zone trust policy default-permit then permit
set security policies from-zone trust to-zone trust policy default-permit then permit
set security policies from-zone trust to-zone trust policy default-permit match source-address any
set security policies from-zone trust to-zone trust policy default-permit match destination-address any
set security policies from-zone trust to-zone trust policy default-permit match application any
```
set security policies from-zone trust to-zone untrust policy default-permit then permit
set security policies from-zone untrust to-zone trust policy Permit_Untrust_to_Trust match source-address any-ipv4
set security policies from-zone untrust to-zone trust policy Permit_Untrust_to_Trust match destination-address any-ipv4
set security policies from-zone untrust to-zone trust policy Permit_Untrust_to_Trust match application junos-icmp-ping
set security policies from-zone untrust to-zone trust policy Permit_Untrust_to_Trust match source-identity any
set security policies from-zone untrust to-zone trust policy Permit_Untrust_to_Trust then permit
set security policies from-zone untrust to-zone trust policy Permit_Untrust_to_Trust then log session-close
set security policies from-zone untrust to-zone trust policy Permit_Untrust_to_Trust then count
set security zones security-zone trust tcp-rst
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-0/0/0.0
set security zones security-zone untrust screen untrust-screen
set security zones security-zone untrust host-inbound-traffic system-services ping
set security zones security-zone untrust host-inbound-traffic system-services dhcp
set security zones security-zone untrust host-inbound-traffic system-services bootp
set security zones security-zone untrust interfaces ge-0/0/1.0
set interfaces ge-0/0/0 unit 0 family inet address 192.168.122.45/24
set interfaces ge-0/0/1 unit 0 family inet address 172.16.1.45/24
set interfaces fxp0 unit 0 family inet address 10.0.41.45/24
set routing-options static route 0.0.0.0/0 next-hop 192.168.122.1

Installing PyEZ and Ansible
vSRX Device Connectivity

Enable PyEZ to communicate with the vSRX through NETCONF. The minimal configuration necessary is:

set system services netconf ssh

Verify that the system is listening on TCP port 830.

Show system connections inet | match 830

Outlined below are the steps to install PyEZ on either the host machine itself or on a separate server on the Trusted network segment.

Install PyEZ

sudo apt-get install python python-pip libxml2-dev libxslt1-dev python-dev zlib1g-dev
sudo apt-get install python-pip
sudo apt-get install git
sudo pip install junos-eznc

Install Ansible and Junos-ansible-stdlib

sudo apt-get install software-properties-common
sudo apt-add-repository ppa:ansible/ansible
sudo apt-get update
sudo apt-get install ansible
ansible-galaxy install Juniper.junos
sudo apt-get install git
git clone https://github.com/Juniper/ansible-junos-stdlib.git

Summary—Virtualize the Grid Edge

Utilities can leverage Juniper Networks vSRX Virtual Firewall software and certified x86 platforms to modernize the grid edge, replacing legacy technology with flexible and efficient routing and security, while reducing deployment and maintenance efforts. In addition to routing and firewalls, Layer 7 services such as application security/monitoring, web filtering and IPS/IDS can be enabled on the vSRX. The vSRX can be managed with a web interface, CLI, Juniper Networks Junos Space Network Management Platform or other umbrella management system. Using standardized hardware, software, and orchestration platforms greatly reduces truck rolls, makes the network more resilient, and adds compute resources to the distribution edge to support new applications.
Next Steps
For more information about the vSRX Virtual Firewall, visit https://www.juniper.net/us/en/products-services/security/srx-series/vsrx/ or contact your Juniper Networks representative for more information about solutions for utilities.

About Juniper Networks
Juniper Networks challenges the status quo with products, solutions and services that transform the economics of networking. Our team co-innovates with customers and partners to deliver automated, scalable and secure networks with agility, performance and value. Additional information can be found at Juniper Networks or connect with Juniper on Twitter and Facebook.