END USER LICENSE AGREEMENT

The Pulse Secure product that is the subject of this technical documentation consists of (or is intended for use with) Pulse Secure software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at http://www.pulsesecure.net/support/eula. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Revision History
2014-12-15 – Changes for rebranding
Abbreviated Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Tables</td>
<td>v</td>
</tr>
<tr>
<td>About This Guide</td>
<td>vi</td>
</tr>
<tr>
<td>PART 1 Virtual Appliances</td>
<td>1</td>
</tr>
<tr>
<td>CHAPTER 1 Virtual Appliances Overview</td>
<td>2</td>
</tr>
<tr>
<td>CHAPTER 2 Deploying Virtual Appliances on VMware ESXi Through vCenter Using OVF Properties</td>
<td>8</td>
</tr>
<tr>
<td>CHAPTER 3 Deploying Virtual Appliances in VMware vSphere Using the Serial Port</td>
<td>16</td>
</tr>
<tr>
<td>CHAPTER 4 Using NETCONF Perl Client to Configure the Virtual Appliance</td>
<td>20</td>
</tr>
<tr>
<td>CHAPTER 5 Deploying on a Kernel-Based Virtual Machine</td>
<td>28</td>
</tr>
<tr>
<td>PART 2 Index</td>
<td>33</td>
</tr>
</tbody>
</table>
# Table of Contents

## List of Tables

<table>
<thead>
<tr>
<th>List of Tables</th>
<th>V</th>
</tr>
</thead>
</table>

## About This Guide

<table>
<thead>
<tr>
<th>Related Documentation and Release Notes</th>
<th>vi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document Conventions</td>
<td>vi</td>
</tr>
<tr>
<td>Requesting Technical Support</td>
<td>vii</td>
</tr>
<tr>
<td>Self-Help Online Tools and Resources</td>
<td>vii</td>
</tr>
<tr>
<td>Opening a Case with PSGSC</td>
<td>vii</td>
</tr>
</tbody>
</table>

## PART 1 Virtual Appliances

### CHAPTER 1 Virtual Appliances Overview

<table>
<thead>
<tr>
<th>Virtual Appliance Editions and Requirements</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware and Software Requirements</td>
<td>2</td>
</tr>
<tr>
<td>Upgrading from a Previous Version</td>
<td>3</td>
</tr>
<tr>
<td>Supported Features on Virtual Appliances</td>
<td>3</td>
</tr>
<tr>
<td>Virtual Appliance Package Information</td>
<td>4</td>
</tr>
<tr>
<td>SPE Virtual Appliance Utility Scripts</td>
<td>6</td>
</tr>
</tbody>
</table>

### CHAPTER 2 Deploying Virtual Appliances on VMware ESXi Through vCenter Using OVF Properties

<table>
<thead>
<tr>
<th>Overview of Deploying Virtual Appliances on VMware ESXi</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using the Deployment Script to Define the Initial Configuration Parameters</td>
<td>8</td>
</tr>
<tr>
<td>Example Output</td>
<td>13</td>
</tr>
<tr>
<td>Verifying Your Deployment with vmware-cmd</td>
<td>14</td>
</tr>
</tbody>
</table>

### CHAPTER 3 Deploying Virtual Appliances in VMware vSphere Using the Serial Port

<table>
<thead>
<tr>
<th>Overview of Scripts for Deploying Through the VMware ESX Hypervisor Serial Port</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using the init-network-config.exp and create-va.exp Scripts</td>
<td>16</td>
</tr>
<tr>
<td><em>init-network-config.exp</em> Parameters</td>
<td>17</td>
</tr>
<tr>
<td><em>create-va.exp</em> Parameters</td>
<td>18</td>
</tr>
</tbody>
</table>

### CHAPTER 4 Using NETCONF Perl Client to Configure the Virtual Appliance

<table>
<thead>
<tr>
<th>Downloading the NETCONF Perl Client and Prerequisites Package</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installing the NETCONF Perl Client and Prerequisites Package</td>
<td>21</td>
</tr>
<tr>
<td>Verifying the Installation and the Version of Perl</td>
<td>21</td>
</tr>
<tr>
<td>Extracting the NETCONF Perl Client and Sample Scripts</td>
<td>21</td>
</tr>
<tr>
<td>Extracting and Installing the NETCONF Perl Client Prerequisites Package</td>
<td>22</td>
</tr>
<tr>
<td>Installing the NETCONF Perl Client Software</td>
<td>24</td>
</tr>
<tr>
<td>Installing the VA-SPE Sample Scripts</td>
<td>25</td>
</tr>
<tr>
<td>Example Output</td>
<td>25</td>
</tr>
<tr>
<td>Using the <em>edit_config_ive.pl</em> Script</td>
<td>26</td>
</tr>
<tr>
<td>Troubleshooting NETCONF Error Messages</td>
<td>26</td>
</tr>
<tr>
<td>Enabling the VMXNET3 Driver</td>
<td>27</td>
</tr>
</tbody>
</table>

### CHAPTER 5 Deploying on a Kernel-Based Virtual Machine

| Deploying on a Kernel-Based Virtual Machine | 28 |
About a Kernel-Based Virtual Machine ................................................................. 28

Limitations ............................................................................................................ 29
Installing the KVM Modules .................................................................................. 29
Deploying a KVM Virtual Appliance ..................................................................... 31

PART 2 Index ....................................................................................................... 33
List of Tables

Table 1: Notice Icons .......................................................................................................................... vi
Table 2: VMware Qualified System ........................................................................................................ 2
Table 3: KVM Qualified System ............................................................................................................... 3
Table 4: create-va.pl Parameters ......................................................................................................... 9
Table 5: Management Port Behavior While Deploying a Template ....................................................... 11
Table 6: Internal Port Behavior While Deploying a Template ............................................................... 11
Table 7: External Port Behavior While Deploying a Template .............................................................. 11
Table 8: Management Port Behavior During a New Deployment ......................................................... 12
Table 9: Internal Port Behavior During a New Deployment ................................................................ 12
Table 10: External Port Behavior During a New Deployment .............................................................. 13
Table 11: init-network-config.exp Parameters ................................................................................... 17
Table 12: create-va.exp Parameters .................................................................................................... 18
Table 13: Copying Files to the NETCONF Installation Directory ....................................................... 25
Table 14: Copying Files to the NETCONF Base Directory .................................................................. 25
Table 15: kvm-va.conf Parameters ....................................................................................................... 31
About This Guide

- Related Documentation and Release Notes
- Document Conventions
- Requesting Technical Support

Related Documentation and Release Notes

For a list of related Pulse Connect Secure documentation, see https://www.pulsesecure.net/techpubs/pulse-connect-secure/pcs.

If the information in the latest release notes differs from the information in the documentation, follow the Pulse Connect Secure Release Notes.

For a list of related Pulse Policy Secure documentation, see https://www.pulsesecure.net/techpubs/pulse-policy-secure/pps.

If the information in the latest release notes differs from the information in the documentation, follow the Pulse Policy Secure Release Notes.

To obtain the most current version of all Pulse Secure technical documentation, see the product documentation page at http://www.pulsesecure.net/techpubs/.

Document Conventions

Table 1 defines notice icons used in this guide.

Table 1: Notice Icons

<table>
<thead>
<tr>
<th>Icon</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔄</td>
<td>Informational note</td>
<td>Indicates important features or instructions.</td>
</tr>
<tr>
<td>⚠️</td>
<td>Caution</td>
<td>Indicates a situation that might result in loss of data or hardware damage.</td>
</tr>
<tr>
<td>🚨</td>
<td>Warning</td>
<td>Alerts you to the risk of personal injury or death.</td>
</tr>
<tr>
<td>🟢</td>
<td>Laser warning</td>
<td>Alerts you to the risk of personal injury from a laser.</td>
</tr>
</tbody>
</table>
Requesting Technical Support

Technical product support is available through the Pulse Secure Global Support Center (PSGSC). If you have a support contract, then file a ticket with PSGSC.

- Product warranties—For product warranty information, visit http://www.pulsesecure.net.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Pulse Secure, LLC has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: http://www.pulsesecure.net/support
- Search for known bugs: http://www.pulsesecure.net/support
- Find product documentation: http://www.pulsesecure.net/techpubs/
- Find solutions and answer questions using our Knowledge Base: http://www.pulsesecure.net/support
- Download the latest versions of software and review release notes: http://www.pulsesecure.net/support
- Search technical bulletins for relevant hardware and software notifications: http://www.pulsesecure.net/support
- Open a case online in the CSC Case Management tool: http://www.pulsesecure.net/support

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: http://www.pulsesecure.net/support

Opening a Case with PSGSC

You can open a case with PSGSC on the Web or by telephone.

- Use the Case Management tool in the PSGSC at http://www.pulsesecure.net/support.
- Call 1-844 751 7629 (Toll Free, US).

For international or direct-dial options in countries without toll-free numbers, see http://www.pulsesecure.net/support.
PART 1 Virtual Appliances

- Virtual Appliances Overview
- Deploying Virtual Appliances on VMware ESXi Through vCenter Using OVF Properties
- Deploying Virtual Appliances in VMware vSphere Using the Serial Port
- Using NETCONF Perl Client to Configure the Virtual Appliance
- Deploying on a Kernel-Based Virtual Machine
Running Pulse Connect Secure or Pulse Policy Secure software in a VMware virtual machine as a virtual appliance provides service providers with robust scalability and isolation. The server software from VMware supports several virtual machines on a high-end multiprocessor platform. Deploying a dedicated virtual appliance for each customer guarantees complete isolation among systems.

- Virtual Appliance Editions and Requirements
- Supported Features on Virtual Appliances
- Virtual Appliance Package Information
- SPE Virtual Appliance Utility Scripts

Virtual Appliance Editions and Requirements

Two types of virtual appliances are available:

- Demonstration and Training Edition (DTE)
- Service Provider Edition (SPE)

The DTE is targeted at demonstration, initial evaluation, and training purposes. DTE is not a supported product; Pulse Secure Support will not assist you with any questions or problems. If you are interested in the DTE, contact your Pulse Secure sales team or reseller.

The SPE is targeted at service providers who are interested in provisioning a remote access solution for a large number of customers.

Hardware and Software Requirements

Table 2 and Table 3 list the virtual appliance systems qualified with this release.

Table 2: VMware Qualified System

<table>
<thead>
<tr>
<th>VMware Tools Version</th>
<th>vCenter/ESXi Version</th>
<th>Qualified Pulse Connect Secure and Secure Access System Versions</th>
<th>Qualified Pulse Policy Secure and Access Control System Versions</th>
<th>Hardware Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.4.0.25793</td>
<td>4.1U3</td>
<td>8.1</td>
<td>5.1</td>
<td>IBM BladeServer H chassis</td>
</tr>
<tr>
<td>5.5</td>
<td></td>
<td>8.0R5</td>
<td>5.0R5</td>
<td>BladeCenter HS blade server</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7.4R10</td>
<td>4.4R10</td>
<td>Allocation for virtual appliance: 4vCPU, 4GB memory and 20GB disk space</td>
</tr>
</tbody>
</table>
NOTE: VMware’s HA feature is qualified; VMware’s DRS & Fault Tolerance features are not qualified.

Table 3: KVM Qualified System

<table>
<thead>
<tr>
<th>QEMU/KVM Version</th>
<th>Qualified Pulse Connect Secure and Secure Access System Versions</th>
<th>Qualified Pulse Policy Secure and Access Control System Versions</th>
<th>Hardware Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1.4.0</td>
<td>8.1</td>
<td>5.1</td>
<td>Linux Server Release 6.4 on an Intel Xeon CPU L5640 @ 2.27GHz</td>
</tr>
<tr>
<td>8.0R5</td>
<td>5.0R5</td>
<td></td>
<td>NFS storage mounted in host</td>
</tr>
<tr>
<td>7.4R10</td>
<td>4.4R10</td>
<td></td>
<td>24GB memory in host</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Allocation for virtual appliance: 4vCPU, 4GB memory and 20GB disk space</td>
</tr>
</tbody>
</table>

Upgrading from a Previous Version

If you are upgrading the Pulse Connect Secure software on your SPE virtual appliance from a version earlier than 7.2 and if VMware high availability (HA) is configured with the VMware VM Monitoring feature, you must change the das.minUptime value in the HA configuration to 600 seconds. If you use the default value of 120 seconds, you will encounter problems during the post-installation processing.

Supported Features on Virtual Appliances

All features of Pulse Connect Secure and Pulse Policy Secure are available on virtual appliances with the exception of the following:

- Instant Virtual System (IVS)
- Clustering
- User record synchronization

An option is available for switching between a virtual terminal and a serial console. Switching between these options requires a restart of the virtual appliance.

Virtual appliances do not allow licenses to be installed directly on them. As such, virtual appliances can be only license clients. All virtual appliance licenses are subscription-based.

We recommend you use the same NTP server for the virtual appliance and the license server to keep the times synchronized. When synchronizing with an NTP server, the Synchronize quest time with host option in the VMware vSphere Client user interface must be enabled. On the virtual appliance, select Edit Settings > Options > VMware Tools to set this option.

Virtual appliances support the following SCSI controller types:

- BusLogic
vSphere users can select the SCSI controller type by opening their Virtual Machine Properties window, clicking the Hardware tab and then double-clicking the SCSI Controller entry.

**Virtual Appliance Package Information**

The SPE downloadable zip contains the following files:

- README-SPE.txt—A quick start guide for the SPE virtual appliance.
- README-scripts.txt—Up-to-date information on the contents of the zip file and how to run the scripts.
- VA-SPE-release-buildnumber-SERIAL-disk1.vmdk—A virtual disk file that contains the Pulse Connect Secure or Pulse Policy Secure software. The SERIAL version assumes using a serial port to set up the initial network configuration.
- VA-SPE-release-buildnumber-SERIAL.ovf—An Open Virtualization Format (OVF) specification that defines the virtual appliance and contains a reference to the disk image.
- VA-SPE-release-buildnumber-VT-disk1.vmdk—A virtual disk file that contains the Pulse Connect Secure or Pulse Policy Secure software. The VT version assumes using a virtual terminal to set up the initial network configuration.
- VA-SPE-release-buildnumber-VT.ovf—An OVF specification that defines the virtual appliance and contains a reference to the disk image.
- init-network-config.exp—A script to configure the initial network settings.
- create-va.exp—A script to import the OVF file into the ESX server and configure the initial network settings (the create-va.exp script is a superset of the init-network-config.exp script).
- setupva.conf—An example configuration file for the create-va.exp and init-network-config.exp scripts.
- create-va.pl—A script for deploying a virtual appliance connected to the VMware vCenter Server.
- va.conf—A sample configuration file for use with the create-va.pl script.
- perlclient/plugin/ive.pm—A side file for configuring virtual appliances through NETCONF.
- perlclient/plugin/ive_methods.pl—A side file for configuring virtual appliances through NETCONF.
- perlclient/examples/get_active_users.pl—A script used to get the current active users on the SPE virtual appliance. Cannot be used for configuring the SPE virtual appliance.
- perlclient/examples/get_active_users.xsl—A file used for formatting and displaying the output returned by get_active_users.pl.
- perlclient/examples/get_active_users.xml—A file used for formatting and displaying the output returned by get_active_users.pl.
- edit_config_ive.pl—A Perl script for editing the SPE virtual appliance configuration.
The DTE downloadable zip contains the following files:

- README-DTE.txt—A quick start guide for the SPE virtual appliance.
- VA-DTE-release-buildnumber-VT-disk1.vmdk—A virtual disk file that contains the Pulse Connect Secure or Pulse Policy Secure software. The VT version assumes using a virtual terminal to set up the initial network configuration.
- VA-DTE-release-buildnumber-VT.ovf—An OVF specification that defines the virtual appliance and contains a reference to the disk image.

The virtual appliance is delivered in OVF and is preconfigured as follows:

- 20-GB virtual SCSI disk
- One virtual CPU
- 2-GB memory
- Three virtual network interfaces
- Roughly 400 MB in size

You can change this configuration by editing the OVF prior to importing it or by editing the virtual machine properties once it is created.

**NOTE:** When customizing the configuration, do not reduce the disk size.

Secure Access Service version 7.3 and later and Access Control Service version 4.3 and later use VMware OVF version 7. This is the preferred version. Virtual appliances created with versions prior to Secure Access Service version 7.3 and Access Control Service version 4.3 use VMware OVF version 4. To upgrade to VMware OVF version 7, you must run Secure Access Service version 7.3 or later or Access Control Service version 4.3 or later.

The OVF specification defines three logical networks:

- Internal Network
- External Network
- Management Network

When importing the OVF file, these three networks must be mapped to the appropriate virtual networks on the ESX server.

The OVF file does not include any virtual serial port configuration. If the SERIAL OVF image is used, the virtual machine specification needs to be updated with the desired virtual serial port configuration before the virtual appliance is powered on for the first time.

When the virtual appliance is powered on for the first time, it expands the software package and performs the installation. After creating a fully installed and configured SPE virtual appliance, clone it to a template and export that template. From the template, you can then instantiate additional SPE virtual appliances.

**NOTE:** Source Network names are not retained in the exported OVF template.
Once configured, you can use any of the following methods to manage the Pulse Connect Secure and Pulse Policy Secure portion of the virtual appliance:

- Juniper Network’s Device Management Interface (DMI)
  
  **NOTE:** The inbound DMI listens to port 830 on both the internal and management interfaces.

- Pulse Connect Secure or Pulse Policy Secure admin console
- Pulse Connect Secure or Pulse Policy Secure serial and virtual terminal console menus

The DMI is an XML-RPC-based protocol used to manage Juniper Networks devices. This protocol allows administrators and third-party applications to configure and manage Juniper Networks devices bypassing their native interfaces. Virtual appliances are compliant with DMI.

By default, the inbound DMI is enabled in virtual appliances.

**Related Documentation**

- DMI Solution Guide

**SPE Virtual Appliance Utility Scripts**

Several utility scripts are included with the SPE virtual appliance package. These scripts assist with:

- Deployment
- Initial setup of the SPE virtual appliance
- Configuring the SPE virtual appliance

You can configure your network with your own set of tools. However, be aware that using tools such as vApp lists options in a different order than what you would see during a typical Pulse Connect Secure or Pulse Policy Secure initial configuration session. As such, even though the scripts included in the SPE package are optional, we recommend you use them.

The scripts are divided into the following sets:

- Deploy the virtual appliance in the VMware vSphere environment on the ESXi hypervisor through vCenter using OVF properties.
  
  Use this script if you are using VMware vCenter Server and VMware ESXi for deploying the virtual appliance. This script can be used on both Virtualization Technology and serial editions of virtual appliances.

- Deploy the virtual appliance in the VMware vSphere environment using a serial port.
  
  If you are using VMware ESX to run the virtual appliance, you can use these scripts for deployment. These scripts use the service console of ESX and can be used only with the serial edition of virtual appliances.

- Use NETCONF Perl client to configure the virtual appliance.
  
  Plug-in and sample scripts for NETCONF Perl client can be used to configure the virtual appliance after it is deployed and powered on. The scripts use DMI for connecting to Pulse Connect Secure or Pulse Policy Secure on port 830.
• Deploy the virtual appliance on KVM.

Use this script if you are using a kernel-based virtual machine (KVM) for deploying the virtual appliance.

Related Documentation

• Overview of Deploying Virtual Appliances on VMware ESXi
• Overview of Scripts for Deploying Through the VMware ESX Hypervisor Serial Port
• Installing the VA-SPE Sample Scripts
CHAPTER 2 Deploying Virtual Appliances on VMware ESXi Through vCenter Using OVF Properties

- Overview of Deploying Virtual Appliances on VMware ESXi
- Using the Deployment Script to Define the Initial Configuration Parameters
- Verifying Your Deployment with vmware-cmd

Overview of Deploying Virtual Appliances on VMware ESXi

VMware ESXi, like VMware ESX, is a hypervisor that installs on top of a physical server and partitions it into multiple virtual machines. VMware ESXi does not contain the ESX's service console and thus is a smaller footprint.

When first powering on the Pulse Connect Secure or Pulse Policy Secure, an administrator must wait for the serial console to appear and manually configure the initial settings. In the case of multiple virtual machines, this process becomes too tedious and time-consuming.

When deploying on a VMware ESXi, the dependencies on a serial console and service console are removed. Pulse Secure lets the administrator set up all initial configuration settings in one pass using a process based on the VMware Guest Customization feature.

With this approach:

1. You use a deployment script and OVF Tools to set up the initial configuration parameters.
2. ESXi passes these parameters into the VMware environment.
3. The virtual appliance retrieves the parameters from the VMware environment and configures the initial settings.

Related Documentation

- Using the Deployment Script to Define the Initial Configuration Parameters
- Verifying Your Deployment with vmware-cmd

Using the Deployment Script to Define the Initial Configuration Parameters

A create-va.pl script is included in your VA-SPE package and is used to deploy a virtual appliance connected to the VMware vCenter Server. This script can be run on any system that has Perl and VMware OVF Tools installed.
Configuration parameters can be passed to the script through a configuration file, command-line options, or a combination of the two. Command-line parameters are passed to the scripts using the following format:

- `paramname` `paramvalue`

Type two hyphens without a space between them for the "- -" string. The space shown here is for visual purposes only.

A sample configuration file (`va.conf`) is provided as an example.

Table 4 lists the parameters for `create-va.pl`. Type two hyphens without a space between them for the "- -" string. The space shown here is for visual purposes only.

**Table 4: create-va.pl Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-vCenterServer</code></td>
<td>Hostname or IP address of the vCenter Server.</td>
</tr>
<tr>
<td><code>-vCenterUsername</code></td>
<td>Username for logging in to the VMware vCenter Server.</td>
</tr>
<tr>
<td><code>-vCenterPassword</code></td>
<td>Password for logging in to the VMware vCenter Server. Special characters in the password must be escaped with a backslash (<code>\</code>). For example, <code>Pulsesecure123$</code>.</td>
</tr>
<tr>
<td><code>-datacenterName</code></td>
<td>Data center under which the Cluster/ESXi Host is present or added.</td>
</tr>
<tr>
<td><code>-clusterorHostName</code></td>
<td>Name of the VMware cluster where the virtual appliance is to be deployed.</td>
</tr>
<tr>
<td></td>
<td>When deploying the virtual appliance in a cluster, this parameter must follow the format <code>cluster-name/ESXi-server-name</code>. For example, <code>ESXI_5_cluster/mydev.pulsesecure.net</code>.</td>
</tr>
<tr>
<td><code>-datastore</code></td>
<td>Name of the datastore where the virtual appliance is to be deployed.</td>
</tr>
<tr>
<td><code>-vaname</code></td>
<td>Name of the virtual appliance to create.</td>
</tr>
</tbody>
</table>

**Pulse Connect Secure and Policy Secure-Related Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-ipAddress</code></td>
<td>IP address to assign to the internal port of the Pulse Connect Secure virtual appliance.</td>
</tr>
<tr>
<td><code>-netmask</code></td>
<td>Netmask to assign to the internal port of the virtual appliance.</td>
</tr>
<tr>
<td><code>-gateway</code></td>
<td>Gateway to assign to the internal port of the virtual appliance.</td>
</tr>
<tr>
<td><code>-adminUsername</code></td>
<td>Username for the default administrator account for the virtual appliance.</td>
</tr>
<tr>
<td><code>-adminPassword</code></td>
<td>Password for the default administrator account for the virtual appliance.</td>
</tr>
<tr>
<td><code>-primaryDNS</code></td>
<td>IP address for the primary DNS server.</td>
</tr>
<tr>
<td><code>-secondaryDNS</code></td>
<td>IP address for the secondary DNS server.</td>
</tr>
<tr>
<td><code>-domain</code></td>
<td>Domain name for the virtual appliance.</td>
</tr>
</tbody>
</table>
**Virtual Appliance-Related Parameters**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-wins</td>
<td>Windows Internet Name Service (WINS) hostname or IP address.</td>
</tr>
<tr>
<td>-commonName</td>
<td>Common name for the default device certificate.</td>
</tr>
<tr>
<td>-organization</td>
<td>Organization for the default device certificate.</td>
</tr>
<tr>
<td>-randomText</td>
<td>Random text to use during certificate creation. If spaces are included in</td>
</tr>
<tr>
<td></td>
<td>the random text, make sure the entire value is enclosed within double-quotes.</td>
</tr>
<tr>
<td></td>
<td>For example, <strong>Pulse Secure Your Net</strong>.</td>
</tr>
</tbody>
</table>

**Virtual Appliance Management Port-Related Parameters**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-ovffile</td>
<td>Path to the OVF file.</td>
</tr>
<tr>
<td>-configFile</td>
<td>Name of configuration files containing parameters to pass to the create-va.pl</td>
</tr>
<tr>
<td></td>
<td>script. Values specified on the command line override the ones specified in</td>
</tr>
<tr>
<td></td>
<td>the configuration file.</td>
</tr>
<tr>
<td>-ExternalNetwork</td>
<td>Virtual network in VMware vSwitch to map the external network of the</td>
</tr>
<tr>
<td></td>
<td>virtual appliance.</td>
</tr>
<tr>
<td>-InternalNetwork</td>
<td>Virtual network in VMware vSwitch to map the internal network of the</td>
</tr>
<tr>
<td></td>
<td>virtual appliance.</td>
</tr>
<tr>
<td>-ManagementNetwork</td>
<td>Virtual network in VMware vSwitch to map the management network of the</td>
</tr>
<tr>
<td></td>
<td>virtual appliance.</td>
</tr>
</tbody>
</table>

**Virtual Appliance External Interface Parameters**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-mgmtipAddress</td>
<td>Management network IP address.</td>
</tr>
<tr>
<td>-mgmtnetmask</td>
<td>Management network netmask address.</td>
</tr>
<tr>
<td>-mgmtgateway</td>
<td>Management network gateway address.</td>
</tr>
<tr>
<td>-managementPortReconfigWithValueInVAppProperties</td>
<td>Management port overwrite property. If set to 1, overwrite the management port-related parameters in the Pulse Connect Secure with the ones defined here. See Table 5 and Table 8.</td>
</tr>
<tr>
<td>-internalPortReconfigWithValueInVAppProperties</td>
<td>The internal port overwrite property. If set to 1, overwrite the virtual appliance’s internal port settings with the ones specified during deployment. See Table 6 and Table 9.</td>
</tr>
</tbody>
</table>

**External Port Overwrite Property**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-extipAddress</td>
<td>External network IP address.</td>
</tr>
<tr>
<td>-extnetmask</td>
<td>External network netmask address.</td>
</tr>
<tr>
<td>-extgateway</td>
<td>External network gateway address.</td>
</tr>
<tr>
<td>-externalPortReconfigWithValueInVAppProperties</td>
<td>External port overwrite property. If set to 1, overwrite the external port-related parameters in Pulse Connect Secure or Pulse Policy Secure with the ones defined here. See Table 7 and Table 10.</td>
</tr>
</tbody>
</table>

**NOTE:** The Pulse Connect Secure and Policy Secure-related parameters are used for the initial configuration of the virtual appliance. The script does not validate these parameters. If the values passed are not valid, the installation will stop at the location where a correct value needs to be provided. The administrator can connect to the virtual appliance using the VT or serial console to complete the initial setup.
Table 5 and Table 6 define the behavior based on options passed while deploying the template.

### Table 5: Management Port Behavior While Deploying a Template

<table>
<thead>
<tr>
<th>Management Port Overwrite Value</th>
<th>Management Port Configuration Values</th>
<th>Pulse Connect Secure and Pulse Policy Secure Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The management port IP address, netmask address and gateway address are valid values.</td>
<td>Because <code>managementPortReconfigWithValueInVAppProperties</code> is 0, the management port-related parameters are retained and are not overwritten with values in the passed configuration.</td>
</tr>
<tr>
<td>0</td>
<td>The management port IP address, netmask address and gateway address are not valid values.</td>
<td>Because <code>managementPortReconfigWithValueInVAppProperties</code> is 0, the management port-related parameters are retained and are not overwritten with values in the passed configuration.</td>
</tr>
<tr>
<td>1</td>
<td>The management port IP address, netmask address and gateway address are valid values.</td>
<td>You can configure the management port with the new values passed while deploying. The existing cache value is overwritten with new values.</td>
</tr>
<tr>
<td>1</td>
<td>The management port IP address, netmask address and gateway address are not valid values.</td>
<td>During the boot process, the administrator is asked whether to configure the management port. Enter N to skip the management port configuration. Enter Y to specify valid values for the management port.</td>
</tr>
</tbody>
</table>

### Table 6: Internal Port Behavior While Deploying a Template

<table>
<thead>
<tr>
<th>Internal Port Overwrite Value</th>
<th>Internal Port Configuration</th>
<th>Pulse Connect Secure and Pulse Policy Secure Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Valid or invalid configuration</td>
<td>Do nothing. The internal port should already be set in the Pulse Connect Secure or Policy Secure. If the internal port is not configured, prompt the administrator to enter the internal port configuration.</td>
</tr>
<tr>
<td>1</td>
<td>Valid configuration</td>
<td>Use the new values passed while deploying and configure the internal port.</td>
</tr>
<tr>
<td>1</td>
<td>Invalid configuration</td>
<td>During the boot process, the administrator is asked whether to configure the internal port. Enter N to skip the internal port configuration. Enter Y to specify valid values for the internal port.</td>
</tr>
</tbody>
</table>

### Table 7: External Port Behavior While Deploying a Template

<table>
<thead>
<tr>
<th>External Port Overwrite Value</th>
<th>Management Port Configuration Values</th>
<th>Pulse Connect Secure and Pulse Policy Secure Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The external port IP address, netmask address and gateway address are valid values.</td>
<td>Because <code>externalPortReconfigWithValueInVAppProperties</code> is 0, the external port-related parameters are retained and are not overwritten with values in the passed configuration.</td>
</tr>
</tbody>
</table>
The external port IP address, netmask address and gateway address are not valid values. Because `externalPortReconfigWithValueInVAppProperties` is 0, the external port-related parameters are retained and are not overwritten with values in the passed configuration.

When deploying a new virtual appliance, the Pulse Connect Secure or Pulse Policy Secure does not contain any configuration. The behavior in this case is shown in Table 8 and Table 9.

**Table 8: Management Port Behavior During a New Deployment**

<table>
<thead>
<tr>
<th>Management Port Overwrite Value</th>
<th>Management Port Configuration Values</th>
<th>Pulse Connect Secure and Pulse Policy Secure Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The management port IP address, netmask address and gateway address are valid values.</td>
<td>Valid management configuration is available. Configure the Pulse Connect Secure or Pulse Policy Secure with these values.</td>
</tr>
<tr>
<td>0</td>
<td>The management port IP address, netmask address and gateway address are not valid values.</td>
<td>Invalid management configuration is present. Do not configure the management port properties.</td>
</tr>
<tr>
<td>1</td>
<td>The management port IP address, netmask address and gateway address are valid values.</td>
<td>Valid management configuration is available. Configure the Pulse Connect Secure or Pulse Policy Secure with these values. The existing cache value is overwritten with new values.</td>
</tr>
<tr>
<td>1</td>
<td>The management port IP address, netmask address and gateway address are not valid values.</td>
<td>During the boot process, the administrator is asked whether to configure the management port. Enter <strong>N</strong> to skip the management port configuration. Enter <strong>Y</strong> to specify valid values for the management port.</td>
</tr>
</tbody>
</table>

**Table 9: Internal Port Behavior During a New Deployment**

<table>
<thead>
<tr>
<th>Internal Port Overwrite Value</th>
<th>Internal Port Configuration</th>
<th>Pulse Connect Secure and Pulse Policy Secure Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 or 1</td>
<td>Valid configuration</td>
<td>Configure the internal port based on the passed configuration values.</td>
</tr>
<tr>
<td>0 or 1</td>
<td>Invalid configuration</td>
<td>During the boot process, the administrator is asked whether to configure the internal port.</td>
</tr>
</tbody>
</table>
### Table 10: External Port Behavior During a New Deployment

<table>
<thead>
<tr>
<th>External Port Overwrite Value</th>
<th>External Port Configuration Values</th>
<th>Pulse Connect Secure and Pulse Policy Secure Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The external port IP address, netmask address and gateway address are valid values.</td>
<td>Valid external configuration is available. Configure the Pulse Connect Secure or Pulse Policy Secure with these values.</td>
</tr>
<tr>
<td>0</td>
<td>The external port IP address, netmask address and gateway address are not valid values.</td>
<td>Invalid external configuration is present. Do not configure the management port properties.</td>
</tr>
<tr>
<td>1</td>
<td>The external port IP address, netmask address and gateway address are valid values.</td>
<td>Valid external configuration is available. Configure the Pulse Connect Secure or Pulse Policy Secure with these values. The existing cache value is overwritten with new values.</td>
</tr>
<tr>
<td>1</td>
<td>The external port IP address, netmask address and gateway address are not valid values.</td>
<td>During the boot process, the administrator is asked whether to configure the external port. Enter N to skip the external port configuration. Enter Y to specify valid values for the external port.</td>
</tr>
</tbody>
</table>

After running the `create-va.pl` script, you can use the VMware vSphere CLI `vmware-cmd` utility or the VMware vSphere Client to view the status. Once vSphere reports the system is ready, you can log in to the virtual appliance.

**NOTE:** The vSphere Client may display a “VMware Tools not installed on this virtual machine” message. You can ignore this message. You do not have to install VMware Tools.

### Example Output

The following example passes the IP address of the internal port through the command line and uses the `va.conf` configuration file for the values of all other parameters.

```bash
perl create-va.pl --configFile /root/darumuga/ovf_dir/va_config_files/7.2R4_va_spe.conf
```

Your output will look similar to the following:

```
The following values are used for creating and configuring the VA

VA Name: "7.2R4_VA_SPE_125_10"
vCenter Server: 10.204.54.201
vCenter Username: darumuga
vCenter Password: Pultesecure123\$
Datacenter Name: PBU-QA
Cluster / Host Name: Dharma_ESXi_5_Cluster/asgdevex3.bngrd.juniper.net
IP Address: 3.3.125.10
Netmask: 255.0.0.0
Gateway: 3.0.0.1
Management IP Address: 10.209.125.10
Management Netmask: 255.255.240.0
```
Management Gateway: 10.209.127.254
Reconfigure Internal Port with value in VAapp properties: 0
Reconfigure Management Port with value in VAapp properties: 0
Primary DNS: 10.209.194.50
Secondary DNS: 1.1.1.1
DNS Domains: juniper.net
WINs: WINSServer
Admin Username: admin
Admin Password: neoteris
Organization: Juniper
Random Text: Juniper_your_Net
ExternalNetwork Mapped to: "IPv6-Ext"
InternalNetwork Mapped to: "IPv6-Int"
ManagementNetwork Mapped to: "ASG QA"

Command = ovftool --skipManifestCheck --name="7.2R4_VA_SPE_125_10" --prop:vaIVEConfig="vaIPAddress=3.3.125.10; vaNetmask=255.0.0.0; vaGateway=3.0.0.1; vaManagementIPAddress=10.209.125.10; vaManagementNetmask=255.255.240.0; vaManagementGateway=10.209.127.254; vaInternalPortReconfigWithValueInVAppProperties=0; vaManagementPortReconfigWithValueInVAppProperties=0; vaPrimaryDNS=10.209.194.50; vaSecondaryDNS=1.1.1.1; vaDNSDomain=juniper.net; vaWINSServer=WINSserver; vaCommonName=secure.juniper.net; vaOrganization=Juniper; vaRandomText=Juniper_your_Net; vaAdminUsername=admin; vaAdminPassword=neoteris" --net:ExternalNetwork="IPv6-Ext" --net:ManagementNetwork="ASG QA" --datastore=devesx3-datastore --powerOn /root/darumuga/ovf_dir/7.2/VA-SPE-SA-7.2R4-21662.1/VA-SPE-SA-21662.1-vt.ovf
vi://darumuga:Juniper123@$10.204.54.201/PBU-QA/host/Dharma_ESXi_5_Cluster/asgdevesx3.bngrd.juniper.net
Status: Task completed
[root@NFS_Server_117_74_Dharma VA-SPE-SA-7.2R4-21662.1]#

Related Documentation

- Overview of Deploying Virtual Appliances on VMware ESXi
- Verifying Your Deployment with vmware-cmd

Verifying Your Deployment with vmware-cmd

Once deployed, the virtual appliance powers on and configures the initial settings for the Pulse Connect Secure or Pulse Policy Secure using the parameters passed by the create-va.pl script. The virtual appliance sets the status of the initial configuration in the valnitConfigStatus guest environment variable. You can check the status of the virtual appliance setup with the VMware vSphere CLI vmware-cmd command. Use the following format:

vmware-cmd -H vCenterName -h ESXi-name vm-cfg-path getguestinfo guestinfo.valnitConfigStatus

For example:

vmware-cmd -H 10.204.54.210 -h asgdevesx2.bngrd.juniper.net \
-U Admin -P Passwd123 "'/vmfs/volumes/ds1/SecureAccess/SecureAccess.vmx" \
guestinfo.getguestinfo guestinfo.valnitConfigStatus

Your output should look similar to this:
getguestinfo(guestinfo.vaInitConfigStatus) = Status: Success Log: Configuring VA settings from OVF; Initial network configuration complete; The self-signed digital certificate was successfully created; VA Initial Configuration completed successfully.

**NOTE:** You can ignore the following message:

vmsvc[280]: [warning] [powerops] Unable to send the status RPC

This message appears when you are running Pulse Connect Secure release 8.0R5 and later with ESXi 4.1U3 or ESXi4.x and you power off and then power up the virtual appliance.

**Related Documentation**

- Using the Deployment Script to Define the Initial Configuration Parameters
CHAPTER 3 Deploying Virtual Appliances in VMware vSphere Using the Serial Port

- Overview of Scripts for Deploying Through the VMware ESX Hypervisor Serial Port
- Using the init-network-config.exp and create(va.exp Scripts

Overview of Scripts for Deploying Through the VMware ESX Hypervisor Serial Port

The following scripts are included in the VA-SPE package for deploying a virtual appliance through the serial port of a VMware ESX hypervisor.

- **init-network-config.exp** – An *expect* script that performs initial network configuration of newly instantiated SPE virtual appliances.

- **create(va.exp** – This script is a superset of the *init-network-config.exp* script. It instantiates an SPE virtual appliance and performs the initial network configuration. The *create(va.exp* script is not needed if you instantiate the virtual appliance another way.

**NOTE:** These scripts are not supported on the VMware ESXi hypervisor.

Prerequisites for these scripts are:

- Scripts must be installed and run on a Linux or UNIX system.
- The Linux or UNIX system must have the VMware Tcl/Tk Expect toolkits installed.
- The virtual appliance must be configured to use a serial console. The serial console must be accessible from the system running the scripts either through a direct connection or by telnet to a terminal server.
- *(create(va.exp Script Only)* The SPE ovf and vmdk files must be on the VMware ESX server where the virtual appliance is to be instantiated. You must use the serial version; the VT version is not supported.

*(create(va.exp Script Only)* The OVF Tool must be installed on the VMware ESX server.

Related Documentation

- Using the init-network-config.exp and create(va.exp Scripts

Using the init-network-config.exp and create(va.exp Scripts

The *init-network-config.exp* and *create(va.exp* scripts accept input from either the command line, a configuration file, or both.
Command-line parameters are passed to the scripts using the following format:

`- paramname paramvalue`

Type two hyphens without a space between them for the "- -" string. The space shown here is for visual purposes only.

Configuration file parameters use the following format:

`paramname: paramvalue`

A configuration file is specified using the `- file` parameter. If no configuration file is specified, the script looks for a `setupva.conf` file in the current directory.

Each parameter must be on a separate line. Lines starting with a pound sign (#) are treated as comments.

If a parameter appears on both the command line and in the configuration file, the command line value is used.

**NOTE:** The `init-network-config.exp` and `create-va.exp` scripts neither validate parameters nor verify connectivity of the serial connection to the virtual appliance.

**init-network-config.exp Parameters**

The `init-network-config.exp` script configures the initial network configuration once the SPE virtual appliance is instantiated.

**Table 11** lists all parameters for the `init-network-config.exp` script.

Type two hyphens without a space between them for the "- -" string. The space shown here is for visual purposes only.

**Table 11: init-network-config.exp Parameters**

<table>
<thead>
<tr>
<th>Parameters for Configuration File</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>- file filename</code></td>
<td>The configuration file. If this parameter is not present, setupva.conf is used. Specifying a configuration file is optional.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameters for Serial Console Access</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>- ctype connecttype</code></td>
<td>The serial console connection. Valid values are &quot;terminal server&quot; or &quot;serial&quot;. You must place the value in double-quotes. For example, <code>- ctype &quot;serial&quot;</code>. Use &quot;terminal server&quot; if the virtual appliance serial console is accessed through a telnet connection to a terminal server. Use &quot;serial&quot; if the virtual appliance serial console is accessed by directly opening a serial device at the system running the script.</td>
</tr>
<tr>
<td><code>- tsip terminal server ip</code></td>
<td>The terminal service IP address. This parameter is valid only when ctype is set to &quot;terminal server&quot;.</td>
</tr>
<tr>
<td><code>- tsport terminal server port</code></td>
<td>The telnet port at the terminal server used to access the virtual appliance serial console.</td>
</tr>
<tr>
<td><code>- cport serial port device</code></td>
<td>The serial device to open to access the virtual appliance serial console. This parameter is valid only when ctype is set to &quot;serial&quot;.</td>
</tr>
</tbody>
</table>
Parameters for Virtual Appliance Initial Configuration

- `ip address` IP address for the virtual appliance internal port.
- `netmask` Virtual appliance internal port netmask.
- `default gateway` Default gateway for the virtual appliance internal port.
- `primary dns` IP address of the primary DNS server for the virtual appliance.
- `secondary dns` Optional IP address of the secondary DNS server for the virtual appliance.
- `domain` Domain name for the virtual appliance.
- `wins` Optional WINS server IP address.
- `common name` Common name for the virtual appliance.
- `organization name` Organization name for the virtual appliance.
- `random text` Random text used for generating a self-signed certificate.
- `adminuser` Administrator username for the virtual appliance.
- `adminpwd` Administrator password for the administrator account at the virtual appliance.

create-va.exp Parameters

The `create-va.exp` script accepts all the `init-network-config.exp` script parameters in Table 11. In addition, the `create-va.exp` script accepts the parameters shown in Table 12 for instantiating the virtual appliance.

Type two hyphens without a space between them for the “- -” string. The space shown here is for visual purposes only.

Table 12: create-va.exp Parameters

<table>
<thead>
<tr>
<th>Parameters to Locate Resources at the ESX Server</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ovf path</code> The location of the OVF image at the VMware ESX server.</td>
</tr>
<tr>
<td><code>esxs serial device</code> Path to the serial port device on the VMware ESX server; for example, <code>esxs seriality /dev/ttyS0</code>. This value creates the virtual serial port in the virtual appliance by editing the vmx file.</td>
</tr>
</tbody>
</table>
The mapping of a logical network name in the OVF specification to an actual network on the VMware ESX server. You can use the following values for ovfnet: InternalNetwork, ExternalNetwork, and ManagementNetwork.

Multiple network maps can be specified by repeating the netmap parameter.

-storage ESX datastore

If there are multiple target datastores on the ESX server, this parameter specifies the datastore where the virtual appliance is to be created.

### Parameters Specifying Virtual Appliance Properties

- **-vaname VA name**
  
  Name of the virtual appliance to create.

- **-switchconsole yes|no**
  
  Switches console from serial to terminal service after configuring the virtual appliance. The default is "yes", meaning switch to terminal service.

The following example uses the `create-va.exp` script:

```
/create-va.exp --ip 100.10.11.11 --mask 255.255.255.0 --dgw 100.10.11.1 --pridns 100.50.10.1 --secdns 100.20.10.1 --domain company.com --wins 111.11.1.11 --cname aaa.company.com --orgname "Company Inc" --rtxt "Somerrandomtext123" --adminusr admin --adminpwd Password --ctype "terminal server" --tsip ts.comp.com --tsport 9999 --esxhost 10.20.50.1 --esxadmin root --esxpasswd pswd --vaname VA-SPE1 --ovfpath /root/ovfs/build12345.ovf --esxserial /tmp/ttyS0 --netmap "InternalNetwork"="ESXInternalNetwork" --netmap "ExternalNetwork"="ESXExternalNetwork" --netmap "ManagementNetwork"="ESXManagementNetwork" storage=datastore1
```

### Related Documentation

- Overview of Scripts for Deploying Through the VMware ESX Hypervisor Serial Port
CHAPTER 4 Using NETCONF Perl Client to Configure the Virtual Appliance

NETCONF API is an XML application that client applications can use to exchange information with Pulse Secure products. The purpose of the NETCONF Perl client is to connect and configure the device by establishing a DMI connection and sending specific remote procedure calls (RPCs). Both the general RPCs supported by Pulse Connect Secure and Pulse Policy Secure and the device-specific RPCs can be used. Some of the device-specific RPCs are used to retrieve runtime information and statistics.

The VA-SPE package contains a NETCONF plug-in for the virtual appliance and sample Perl scripts. Using the supplied scripts as an example, you can write your own scripts for any DMI RPCs supported by the virtual appliance.

See the DMI Solution Guide located on the Pulse Secure Support website.

- Downloading the NETCONF Perl Client and Prerequisites Package
- Installing the NETCONF Perl Client and Prerequisites Package
- Installing the VA-SPE Sample Scripts
- Troubleshooting NETCONF Error Messages
- Enabling the VMXNET3 Driver

**Downloading the NETCONF Perl Client and Prerequisites Package**

This topic explains how to download the compressed tar archives that contain the NETCONF Perl client distribution and the prerequisite package.

To download the compressed tar archives:

2. Click the link for the appropriate software release.
   - The virtual appliance supports version 11.1R1.14 and later.
3. Select the Software tab.
4. Click **NETCONF API Perl client** and **NETCONF API Perl client prerequisites** to download the client distribution and the prerequisites package.

**NOTE:** The NETCONF XML protocol Perl client software should be installed on a computer with a UNIX-like operating system; it is not meant to be installed on a Juniper Networks device.
CHAPTER 4: Using NETCONF Perl Client to Configure the Virtual Appliance

Optionally, you can download the packages containing the document type definitions (DTDs) and the XML schema language representation of the Junos OS configuration hierarchy:

To download the package containing the DTDs and XML:

2. Click the link for the appropriate software release.
3. Select the Software tab.
4. Click the links to download the desired packages.

Related Documentation
- Installing the NETCONF Perl Client and Prerequisites Package
- Installing the VA-SPE Sample Scripts

Installing the NETCONF Perl Client and Prerequisites Package

This topic explains how to install the NETCONF Perl client and the prerequisite package. It includes the following sections:

- Verifying the Installation and the Version of Perl
- Extracting the NETCONF Perl Client and Sample Scripts
- Extracting and Installing the NETCONF Perl Client Prerequisites Package
- Installing the NETCONF Perl Client Software

Verifying the Installation and the Version of Perl

Perl must be installed on your system before you install the NETCONF Perl client prerequisites package or client software. The NETCONF Perl client requires Perl version 5.6.1 or later. To confirm whether Perl is installed on your system and to determine which version of Perl is currently running, issue the following commands:

```bash
$ which perl
$ perl -v
```

If the issued output indicates that Perl is not installed or that the version is earlier than the required version, you must download and install Perl version 5.6.1 or later in order to use the NETCONF Perl client. The Perl source packages are located at:

http://www.cpan.org/src/.

After installing a suitable version of Perl, extract the NETCONF Perl client, extract and install the prerequisites package, and then install the NETCONF Perl client application.

Extracting the NETCONF Perl Client and Sample Scripts

To uncompress and extract the contents of the compressed tar archive that contains the NETCONF Perl client and sample scripts:

1. Create the directory where you want to store the `NET::Netconf` Perl client application and sample scripts.
2. Move the downloaded client application file into that directory.

3. Make that directory the working directory.

   $ mkdir parent-directory
   $ mv netconf-perl-release.tar.gz parent-directory
   $ cd parent-directory

4. Uncompress and extract the contents of the NETCONF Perl client package:

   - On FreeBSD and Linux systems:
     $ tar zxf netconf-perl-release.tar.gz
   - On Solaris systems:
     $ gzip -dc netconf-perl-release.tar.gz | tar xf

   where release is the release code, for example 11.1R1.14. The command creates a directory called netconf-perl-release and extracts the contents of the tar archive to it. For example, a typical filename for the compressed tar archive is netconf-perl-11.1R1.14.tar.gz. Extracting the contents of this archive creates the directory netconf-perl-11.1R1.14 directly under parent-directory and places the application files and sample scripts into this new directory.

   The netconf-perl-release/README file contains instructions for extracting and installing the Perl prerequisite modules, creating a Makefile, and installing and testing the NET::Netconf module.

Extracting and Installing the NETCONF Perl Client Prerequisites Package

The prerequisites package consists of C libraries, executables, and Perl modules. It must be installed on the client machine for the NETCONF Perl client and the included examples to work correctly. The NETCONF Perl distribution includes the install-prereqs.pl script, which you use to install the prerequisites. You have the option to install all Perl modules that are part of the prerequisites directly from the Comprehensive Perl Archive Network (CPAN) global repository.

To uncompress and extract the contents of the compressed tar archive containing the prerequisite files:

1. Move the downloaded prerequisites package into the parent-directory/netconf-perl-release directory that was created in “Extracting the NETCONF Perl Client and Sample Scripts”.

   The compressed tar archive containing the prerequisite files must be uncompressed, unpacked, and installed in that directory.

2. Uncompress and extract the contents of the package:

   - On FreeBSD and Linux systems:
     $ tar zxf netconf-perl-prereqs-release.tar.gz
   - On Solaris systems:
     $ gzip -dc netconf-perl-prereqs-release.tar.gz | tar xf

   where release is the release code, for example 11.1R1.14. This command creates a directory called prereqs/ and extracts the contents of the tar archive to it.
By default, the prerequisite Perl modules are installed in the standard directory, which is /usr/local/lib/. You can opt to install the modules in a private directory.

To install the required modules in the standard directory:

1. Log in as root.
2. Go to the netconf-perl-release directory where you extracted the contents of the prerequisites package.
3. Issue the following command:
   
   ```
   # perl install-prereqs.pl -used_by example -force
   
   where the -used_by example option is invoked to install only modules used by a specific example, and the -force option installs the module even if an earlier version of the module exists or if the make test command fails.
   ```

To install the required modules in a private directory:

1. Set the PERL5LIB, MANPATH, and PATH environment variables.
   
   ```
   $ setenv PERL5LIB private-directory-path
   $ setenv MANPATH "$MANPATH:/$PERL5LIB/../man"
   $ setenv PATH "SPATH:/$PERL5LIB/../bin"
   
   For sh, ksh, and bash shells, $PERL5LIB can be set with EXPORT PERL5LIB=private-directory-path
   ```
2. Go to the netconf-perl-release directory where you extracted the contents of the prerequisites package.
3. Issue the following command:
   
   ```
   $ perl install-prereqs.pl -used_by example -install_directory $PERL5LIB -force
   
   where the -used_by example option is invoked to install only modules used by a specific example, and the -force option installs the module even if an earlier version of the module exists or if the make test command fails. The -install_directory $PERL5LIB option installs the prerequisite Perl modules in the private directory that you specified in Step 1.
   ```

After issuing the perl install-prereqs.pl command, the script provides the option to install the prerequisites from CPAN. The CPAN module is included with standard Perl installations. If you choose to install from CPAN, the script checks that the CPAN module is installed on your system and that you have connectivity to www.cpan.org. If the CPAN module is present and connectivity is verified, installation begins automatically.

To install from CPAN:

1. Press Enter or type y when prompted.
   
   ```
   # perl install-prereqs.pl
   
   This script installs all modules required by default.
   Would you like to install the pre-requisite modules from CPAN? [y]/n y
   
   Testing MCPAN on your system...
   OK
   
   Trying to ping CPAN
   OK
   
   These modules will be installed in the system directory.
   ```
This installation takes around 15 minutes

Begin automatic installation:
<output omitted>

You might be prompted for additional information during the installation. For example, if additional dependent modules are required for a specific module, the installer might ask if the missing modules should be added to the install queue.

**NOTE:** On some systems, the firewall might reject utilities that are set to use active FTP, and CPAN installation might hang. If this is an issue, set the corresponding environment variable so that passive FTP is enabled.

Installation log files are written to netconf-perl-release/tmp/output/.

2. After installation, view any missing dependencies by issuing the following command:

$ perl required-mod.pl

This command lists the modules that still require installation.

### Installing the NETCONF Perl Client Software

After installing the prerequisites package as detailed in “Extracting and Installing the NETCONF Perl Client Prerequisites Package”, install the NETCONF Perl client software.

To install the client software:

1. Go to the netconf-perl-release/ directory that was created in “Extracting the NETCONF Perl Client and Sample Scripts”.

2. Create the makefile.
   - To install the Perl client in the standard directory (usually /usr/local/lib):
     
     # perl Makefile.PL
     
     Checking if your kit is complete...
     
     Looks good
     
     Writing Makefile for netconf-perl

   - To install the Perl client in a private directory:
     
     Make sure that the PERL5LIB, MANPATH, and PATH environment variables are set as detailed in “Extracting and Installing the NETCONF Perl Client Prerequisites Package”, and create the makefile:

     # perl Makefile.PL LIB=$PERL5LIB INSTALLMAN3DIR=$PERL5LIB/../man/man3

3. Install the Net::NETCONF module:

   # make
   
   # make install

The NETCONF Perl client installs under /usr/lib/perl5/site_perl/Perl version/Net/Netconf or /usr/local/lib/perl5/site_perl/Perl version/i686-linux/Net/, depending on which version of Perl 5 you have installed.
CHAPTER 4: Using NETCONF Perl Client to Configure the Virtual Appliance

The NETCONF Perl client application is installed and ready for use. For information about the Net::NETCONF::Manager, Net::NETCONF::Transform, or Net::NETCONF::Trace classes, consult the appropriate man page by invoking the man command and specifying the class. For example:

```
$ man Net::NETCONF::Manager
$ man Net::NETCONF::Transform
$ man Net::NETCONF::Trace
```

The sample scripts reside in the netconf-perl-release/examples/ directory. You can review and run these examples to acquire some familiarity with the client before writing your own applications.

Related Documentation

- Downloading the NETCONF Perl Client and Prerequisites Package
- Installing the VA-SPE Sample Scripts

Installing the VA-SPE Sample Scripts

After you download and install the VA-SPE and NETCONF packages, you must copy certain files from the VA-SPE installation to parent-directory/netconf-perl-release. Table 13 shows the files to copy and their required destinations.

Table 13: Copying Files to the NETCONF Installation Directory

<table>
<thead>
<tr>
<th>Copy This File</th>
<th>To This Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>get_active_users.pl</td>
<td>parent-directory/netconf-perl-release/examples</td>
</tr>
<tr>
<td>get_active_users.xml</td>
<td>parent-directory/netconf-perl-release/examples</td>
</tr>
<tr>
<td>get_active_users.xsl</td>
<td>parent-directory/netconf-perl-release/examples</td>
</tr>
</tbody>
</table>

You must also copy files to the /usr/local/lib/perl5/site_perl/Perl version/i686-linux/Net/Netconf/Access/ or /usr/lib/perl5/site_perl/Perl version/Net/Netconf directories, depending on which version of Perl you have installed. This guide uses NETCONF-Base-Directory to refer to /usr/local/lib/perl5/site_perl/Perl version/i686-linux/Net/Netconf/Access/ or /usr/lib/perl5/site_perl/Perl version/Net/Netconf.

Table 14: Copying Files to the NETCONF Base Directory

<table>
<thead>
<tr>
<th>Copy This File</th>
<th>To This Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>ive.pm</td>
<td>NETCONF-Base-Directory/Net/Netconf/Plugins/Plugin</td>
</tr>
<tr>
<td>ive_methods.pl</td>
<td>NETCONF-Base-Directory/Net/Netconf/Plugins/Plugin/ive</td>
</tr>
</tbody>
</table>

Once the files are copied, you can run the sample code.
Example Output

The following example uses user for the username, passwd123 for the password and 10.20.30.40 for the IP address. When run, it connects to the virtual appliance, retrieves the list of active users, and puts the list in a table format, translating the XML using the get_active_users.xsl file.

```perl
perl get_active_users.pl -l user1 -p passwd123 10.20.30.40
```

Using the edit_config_ive.pl Script

The edit_config_ive.pl script is used for editing the SPE virtual appliance configuration and has the following syntax:

```perl
perl edit_config_ive.pl options request target
```

where:

<table>
<thead>
<tr>
<th>options</th>
<th>One or more of the following:</th>
</tr>
</thead>
<tbody>
<tr>
<td>-l login</td>
<td>Login name accepted by the target device.</td>
</tr>
<tr>
<td>-p password</td>
<td>Password associated with the login name.</td>
</tr>
<tr>
<td>-m access</td>
<td>The access method. The only supported value is ssh.</td>
</tr>
<tr>
<td>-d level</td>
<td>Debug level. Values can be 1 (terse) through 6 (verbose).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>request</th>
<th>Name of the file containing the configuration in XML format. An example of the contents of a configuration file is:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&lt;configuration&gt; &lt;system&gt; &lt;network&gt; &lt;network-overview&gt; &lt;settings&gt; &lt;node&gt;localhost2&lt;/node&gt; &lt;hostname&gt;chandrashekar&lt;/hostname&gt; &lt;/settings&gt; &lt;/network-overview&gt; &lt;/network&gt; &lt;/system&gt; &lt;/configuration&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>target</th>
<th>Hostname of the target device.</th>
</tr>
</thead>
</table>

Related Documentation

- Installing the NETCONF Perl Client and Prerequisites Package
- Troubleshooting NETCONF Error Messages

Troubleshooting NETCONF Error Messages

The error “Connection to Netconf server lost at /usr/lib/perl5/site_perl/5.8.5/Net/Netconf/Manager.pm” while running Perl scripts has been resolved with NETCONF Perl Client 12.3R1, available on the Pulse Secure support site.

However, you might still encounter this error because of timeout issues. To resolve this issue, edit the ssh.pm file and change “10” to “30” shown in the following italicized lines and then run
CHAPTER 4: Using NETCONF Perl Client to Configure the Virtual Appliance

your Perl script again. The ssh.pm file is located in /usr/local/lib/perl5/site_perl/Perl version/i686-linux/Net/Netconf/Access/ or /usr/lib/perl5/site_perl/Perl version/Net/Netconf, depending on which version of Perl you have installed.

```perl
$ssh->log_file($self->out);
  # Send our password or passphrase
  if ($ssh->expect(30, 'password:', 'Password:', '(yes/no)?', 'yes', 'passphrase.*:')) {
    my $m_num = $ssh->match_number();
    SWITCH: {
      if (($m_num == 1) || ($m_num == 2) || ($m_num == 4)) {
        print $ssh "$self->{password}
      }
      last SWITCH;
    }
    if ($m_num == 3) {
      # Host-key authenticity.
      print $ssh "yes\r"
      if ($ssh->expect(30, 'password:', 'Password:', 'yes', 'passphrase.*:')) {
        ...
    }
```

Related Documentation

- Installing the VA-SPE Sample Scripts

Enabling the VMXNET3 Driver

To enable the VMXNET3 driver in your virtual appliance, you must deploy from the 7.2 OVF package. Upgrading from earlier versions such as 7.0 or 7.1 will continue to use VMXNET instead of the VMXNET3 driver.
CHAPTER 5 Deploying on a Kernel-Based Virtual Machine

- About a Kernel-Based Virtual Machine
- Installing the KVM Modules
- Deploying a KVM Virtual Appliance

About a Kernel-Based Virtual Machine

Kernel-based Virtual Machine (KVM) is a virtualization solution for Linux on x86 hardware containing Intel VT or AMD-V virtualization extensions. A wide variety of guest operating systems work with KVM, including Linux, Windows, OpenBSD and others. You can run a Pulse Secure virtual appliance as a guest operating system on any Linux machine with KVM hypervisor support.

NOTE: QEMU is an open source emulator that provides a monitor mode when using the KVM kernel module. This monitor mode can perform operations like powering on or off the virtual appliance. If you use this monitor mode to power on or off the virtual appliance, no logs are generated. Only administrators logged into the Pulse Connect Secure or Pulse Policy Secure console are informed of the pending shutdown.

Before proceeding, verify that your CPU supports virtualization by running one of the following commands:

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

```
cat /proc/cpuinfo | grep vmx
```

Your CPU supports virtualization if:

- The `egrep` command returns a non-zero value.
- The `cat` command returns a result that contains the string `vmx`.

You must also check that virtualization is enabled in your BIOS. After enabling this feature, you must turn your machine off and then on again for the change to take effect.

Once your machine reboots, check that everything is configured correctly by running the `kvm-ok` command. Your output should look similar to this:

```
/usr/bin/kvm-OK
INFO: Your CPU supports KVM extensions
INFO: /dev/kvm exists
KVM acceleration can be used
```
Limitations

For each KVM virtual appliance instance with 4 GB Memory and 4 CPU allocation, exceeding 5000 tunnels (Network Connect, Pulse Secure client ESP/SSL, or a combination of both) with 60 Mbps of bi-directional traffic may exhibit high CPU utilization and loss of throughput including disruption of the existing connections.

Related Documentation

- Installing the KVM Modules
- Deploying a KVM Virtual Appliance

Installing the KVM Modules

This topic describes how to install KVM modules. You can run these commands as root or by using sudo, if sudo is available on your system. The following examples are run as root.

Pulse Secure supports kernel modules version 2.6.18 and later.

To install KVM, run the following commands:

[root@localhost ~]# insmod lib/modules/KernelVersion/kernel/arch/x86/kvm/kvm.ko

[root@localhost ~]# insmod lib/modules/KernelVersion/kernel/arch/x86/kvm/kvm-intel.ko

These commands return no output unless an error has occurred. If an error occurs, details about that error are displayed.

Check that the KVM modules are installed by running the lsmod command. Your output should look similar to this.

[root@localhost ~]# lsmod | grep kvm

kvm_intel   50380  3
kvm    305081  1 kvm_intel

If the KVM modules are not installed, your output will look similar to this:

[root@localhost ~]# lsmod | grep kvm
[root@localhost ~]#

Refer to your KVM documentation if your KVM modules do not install properly.

For the virtual appliance to access the host system’s network, set up a bridge interface. The following steps create a bridge interface, br0, and map it to the physical interface eth0 making your virtual appliances accessible from your local network. These instructions assume that your host system has only one network interface, eth0.

NOTE: Depending on your installation, the bridge-util packages might be installed as part of another package. Check your installation and manually install the bridge-util packages if necessary before continuing.

1. Change directory to where the network scripts are located.

   [root@localhost ~]# cd /etc/sysconfig/network-scripts/

2. Copy ifcfg-eth0 to ifcfg-br0 to create the bridge interface.
3. Edit the ifcfg-br0 file and change the DEVICE line to `DEVICE="br0"` and set `TYPE="Bridge"`.

```bash
[root@localhost network-scripts]# vim ifcfg-br0
DEVICE="br0"    #Change
BOOTPROTO="static"
HWADDR="00:30:48:32:E0:4E"
NM_CONTROLLED="yes"
ONBOOT="yes"
TYPE="Bridge"    #Change
IPADDR="10.204.56.142"
NETMASK="255.255.240.0"
GATEWAY="10.204.63.254"
```

4. Edit the ifcfg-eth0 file and add `BRIDGE="br0"`.

```bash
[root@localhost network-scripts]# vim ifcfg-eth0
DEVICE="eth0"
HWADDR="00:30:48:32:E0:4E"
NM_CONTROLLED="yes"
ONBOOT="yes"
TYPE="Ethernet"
IPADDR="10.204.56.142"
NETMASK="255.255.240.0"
GATEWAY="10.204.63.254"
BRIDGE="br0"    #Add
```

5. Apply the new network settings by running the following command.

```bash
[root@localhost /]# /etc/rc.d/init.d/network restart
```

Note that the eth0 device will no longer have an IP address; the br0 device has the IP after bridging is operational.

6. Display the current TCP/IP network configurations to confirm the bridge network is created.

```bash
[root@localhost /]# ifconfig
```

An example output is shown here:

```
br0    Link encap:Ethernet  HWaddr 00:30:48:32:E0:4E
       inet addr:10.204.56.142  Bcast:10.204.63.255  Mask:255.255.240.0
       inet6 addr: fe80::230:48ff:fe32:e04e/64 Scope:Link
       UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
       RX packets:4406929 errors:0 dropped:0 overruns:0 frame:0
       TX packets:1080664 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:0
       RX bytes:4082423409 (3.8 GiB)  TX bytes:158009811 (150.6 MiB)
eth0   Link encap:Ethernet  HWaddr 00:30:48:32:E0:4E
       inet addr:10.204.56.142  Bcast:10.204.63.255  Mask:255.255.240.0
       inet6 addr: fe80::230:48ff:fe32:e04e/64 Scope:Link
       UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
       RX packets:4406929 errors:0 dropped:0 overruns:0 frame:0
       TX packets:1080664 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:0
       RX bytes:4082423409 (3.8 GiB)  TX bytes:158009811 (150.6 MiB)
lo     Link encap:Local Loopback
       inet addr:127.0.0.1  Mask:255.0.0.0
```

```bash
cp ifcfg-eth0 ifcfg-br0
```
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:8617 errors:0 dropped:0 overruns:0 frame:0
TX packets:8617 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:1594571 (1.5 MiB) TX bytes:1594571 (1.5 MiB)
virbr0    Link encap:Ethernet HWaddr 52:54:00:FE:C2:76
inet addr:192.168.122.1 Bcast:192.168.122.255 Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
RX packets:746 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:0 (0.0 b)  TX bytes:39254 (38.3 KiB)

Related Documentation

- About Kernel-based Virtual Machine
- Deploying a KVM Virtual Appliance

Deploying a KVM Virtual Appliance

The `kvm-create-img.pl` script is included in the VA-SPE package for deploying a Pulse Secure virtual appliance image as a guest OS on a KVM machine. The `kvm-create-img.pl` script accepts input from either the command line, a configuration file, or both. Configuration parameters can be passed to the script through a configuration file, command-line options, or a combination of the two. Command-line parameters are passed to the scripts using the following format:

```
- paramname paramvalue
```

Type two hyphens without a space between them for the "- -" string. The space shown here is for visual purposes only. A complete list of parameters is shown in Table 15.

Table 15: kvm-v.conf Parameters

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory</td>
<td>Number of GB of host virtual space (physical memory) to allocate for the guest. The default is 2 GB.</td>
</tr>
<tr>
<td>IntMapBridge</td>
<td>Bridge internal port gateway. The default is br0.</td>
</tr>
<tr>
<td>ExtMapBridge</td>
<td>Bridge external port gateway. The default is br0.</td>
</tr>
<tr>
<td>MgtMapBridge</td>
<td>Bridge management port gateway. The default is br0.</td>
</tr>
<tr>
<td>IntTapInterface</td>
<td>Internal tap networking device in the host. The default is tap1.</td>
</tr>
<tr>
<td>ExtTapInterface</td>
<td>External tap networking device in the host. The default is tap2.</td>
</tr>
<tr>
<td>MgtTapInterface</td>
<td>Management tap networking device in the host. The default is tap3.</td>
</tr>
<tr>
<td>IntMacAddr</td>
<td>Guest internal port MAC address. The default value is no MAC address.</td>
</tr>
<tr>
<td>ExtMacAddr</td>
<td>Guest external port MAC address. The default value is no MAC address.</td>
</tr>
<tr>
<td>MgtMacAddr</td>
<td>Guest management port MAC address. The default value is no MAC address.</td>
</tr>
</tbody>
</table>
Consoletype
Serial console interface to directly access the KVM guest. The default is telnet:localhost:9999,server,nowait.

Interfaceup
Location and name of script to a tap to an existing bridge when the guest starts. The default is /root/kvm-rlifupNew.

Interfacedown
Location and name of script to remove a tap when the guest exits. The default is /root/kvm-rlifdownNew.

Monitorva
Interface to access the virtual machine monitor. The default is telnet:localhost:8888,server,nowait.

If a parameter appears on both the command line and in the configuration file, the command line value is used. The `kvm-create-img.pl` script neither validates parameters nor verifies connectivity to the virtual appliance.

The script format is:

```
kvm-create-img.pl -paramname paramvalue configurationFile -valimageimageName
```

The following two examples show how to run the `kvm-create-img.pl` script.

```
./kvm-create-img.pl -configFile kvm_va.conf -valimage VA-SPE-SA-26686.1-SERIAL.img
./kvm-create-img.pl -Memory 2G -IntMapBridge br0 -Consoletype telnet::9999,server,nowait
   -valimage VA-SPE-SA-26686.1-SERIAL.img
```

Each parameter must be on a separate line. Lines starting with a pound sign (#) are treated as comments.

The following example shows the default `kvm-va.conf` file.

```
Memory = 2G
IntMapBridge = br0
ExtMapBridge = br0
MgtMapBridge = br0
IntTapInterface = tap1
ExtTapInterface = tap2
MgtTapInterface = tap3
IntIfMacAddr =
ExtIfMacAddr =
MgtIfMacAddr =
Consoletype = telnet:localhost:9999,server,nowait
Interfaceup = /root/kvm-rlifupNew
Interfacedown = /root/kvm-rlifdownNew
Monitorva = telnet:localhost:8888,server,nowait
```

Related Documentation

- About Kernel-based Virtual Machine
- Installing the KVM Modules
PART 2 Index

- Index
## PART 2 Index

<table>
<thead>
<tr>
<th>C</th>
<th>configuration file</th>
<th>va.conf ..............................................13</th>
<th>create-vd.exp script ..................................4</th>
<th>create-vd.pl script ..................................4</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>get_active_users.pl ..........................................4</td>
<td>get_active_users.xml ....................................5</td>
<td>get_active_users.xsl ....................................4</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>init-network-config.exp script ................................4</td>
<td>install-prereqs.pl script ................................23</td>
<td>ive.pm ......................................................4</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ive_methods.pl ..........................................4</td>
</tr>
<tr>
<td>K</td>
<td>kvm-create-img.pl script ......................................32</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>requirements</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| S | script | create-vd.exp ..........................................4 | create-vd.pl .............................................4 | init-network-config.exp ................................4 | install-prereqs.pl .....................................23 | kvm-create-img.pl ......................................32 |

| T | technical support ...........................................vii | contacting PSGSC ........................................vii |

| V | va.conf configuration file ..................................13 | vmware-cmd ..............................................14 |

| Z | zip file | contents .................................................4 |