Corero Network Security

SmartWall Threat Defense Director
Getting Started Guide (KVM)

Software 9.7.5

10 December 2020

Part Number: 9502-0975-00

Corero Network Security © 2020 · www.corero.com

Legal and Copyright Information

Corero Network Security, Inc. (Corero) reserves the right to revise this documentation and to make changes in content from time to time without obligation on the part of Corero to provide notification of such revision or change. Corero provides this documentation without warranty, term, or condition of any kind, either implied or expressed, including, but not limited to, the implied warranties, terms or conditions of merchantability, satisfactory quality, and fitness for a particular purpose. Corero may make improvements or changes in the product(s) and/or the program(s) described in this documentation at any time.

If you are a United States government agency, this documentation and the software described herein are provided to you subject to the following:

This paragraph applies to all acquisitions of the software by or for the United States Government, or by any prime contractor or subcontractor (at any tier) under any contract, grant, cooperative agreement or other activity with the United States Government (collectively, the "Government"). All technical data and computer software are commercial in nature and developed solely at private expense. The software and documentation respectively are "commercial computer software" and "commercial computer software documentation" as defined in DFARS 252.227-7014 (June 1995) and "commercial items" as defined in FAR 2.101(a) and, to the maximum extent permitted by law, are provided with only such rights as are provided in Corero's standard commercial license for the software and documentation and this notice. Technical data is provided with limited rights only as provided in DFAR 252.227-7015 (November 1995) or FAR 52.227-14 (June 1987), whichever is applicable. Corero's standard commercial license for the software and documentation and this notice shall govern the Government's use of the software, documentation, and technical data, and shall supersede any conflicting contractual terms or conditions. If these terms and conditions fail to meet the Government's needs or is inconsistent in any respect with Federal law, the Government must return the software and the documentation unused to Corero. The following additional statement applies only to acquisitions governed by DFARS Subpart 227.4 (October 1988): "Restricted Rights – Use, duplication and disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 (OCT. 1988)." The Contractor is Corero Network Security, Inc.

You agree not to remove or deface any portion of any legend provided on any licensed program or documentation contained in, or delivered to you in conjunction with, this document.

No part of this documentation may be reproduced in any form or by any means or used to make any derivative work (such as translation, transformation, or adaptation) without written permission from Corero.

The products described in this document are protected by US Patent No. 9,442,782, US Patent No. 10,341,364, and European Patent No. 1319296.

Any software on removable media described in this documentation, is furnished under a license agreement which is located on the Corero web site.

Corero®, First Line of Defense®, SecureWatch®, and SmartWall® are registered trademarks of Corero Network Security, Inc. All other trademarks and registered trademarks are the property of their respective holders.

For warranty, licensing and maintenance agreement information, visit http://www.corero.com/support/End_User_Agreements.html.

Copyright © 2014- 2020, Corero Network Security, Inc.

CONTENTS

Legal and Copyright Information **2**

Contents **3**

TDD Documentation **8**

SmartWall Threat Defense Director **9**

Working with the SmartWall TDD applications and documentation **10**

Core Concepts **11**

Provisioning Command Line Interface (pCLI) 11

Policy 11

Protection Profiles 11

Clusters 11

Devices 11

Segments 11

Defense Mode 12

Analytics 12

Sampled Traffic 12

Telemetry 12

NETCONF 13

SmartWall Service Portal 13

Deployment Checklist **14**

Network Configuration for TDD Deployment **16**

Securing your SmartWall applications **16**

Firewall Considerations **17**

SmartWall TDD Deployment **18**

CONTENTS

Separately hosted TDD applications managing one Juniper Networks MX Series router 18

TDD applications, on one host, managing one Juniper Networks MX Series router 19

Deploying vNTDs for Different Sampled Traffic Rates 20

Using this document to deploy a vNTD 20

Deploying the TDD Virtual Components 21

Juniper Networks MX Series router requirements 21

Virtual Editions components 21

Required information 22

System requirements 22

Host requirements 22

Virtual appliance requirements 23

Additional Requirements 24

Deploying a virtual edition on a KVM server 25

Prerequisites 25

To deploy a Corero Virtual Appliance using virt-install 25

To deploy a Corero Virtual Appliance using virsh 27

To deploy a Corero Virtual Appliance using virt-manager 27

(vNTD only) Verify ordering of network interfaces 31

Configuring the TDD Components 32

Accessing the pCLI on a virtual appliance 32

Resizing a vSWA application 32

Configuring SmartWall Components Using the pCLI 33

Using the pCLI Setup Wizard to Configure a SmartWall Component 33

CONTENTS

(Optional) SmartWall SecureWatch Analytics Considerations 37

Installing the TDD license file **39**

Uploading a vNTD license to the CMS **40**

Setting the Inbound Sample Rate between the vNTD and the CMS **41**

Adding a vNTD to the CMS **41**

Prerequisites 41

To edit the default Cluster to expect sampled traffic 41

To add a vNTD to the CMS 42

Configuring the vNTD Segment for TDD **44**

Connecting CMS with SWA **45**

(Optional) Add a signed certificate to the CMS - SWA connection 46

To add CMS credentials to the SWA **47**

(Optional) Uploading a custom SSL certificate to the CMS **47**

Configuring the Juniper Networks MX Series router **48**

Prerequisites **48**

To configure a Juniper Networks MX Series router for use with the SmartWall TDD system **50**

Adding a Juniper Networks MX Series router as a Remote Device **54**

To add a router to the SWA **54**

To configure the Juniper Alert to send mitigations to those devices **55**

Verifying the TDD System is Connected **56**

To verify the TDD system is connected **56**

To force a Remote Device Info system check 56

Configuring the TDD Policy for your Network **58**

Troubleshooting **59**

CONTENTS

Cannot access the Web UI (CMS or SWA) 59

Getting help for using the CMS or SWA 59

CMS configuration change does not take effect 59

Defense device not reachable from CMS 59

The Defense device shows out-of-sync in the CMS 60

vNTD device showing as not-licensed 60

Remote Device added to the CMS Devices table instead of the SWA 61

Cannot add a new vNTD to a CMS Cluster 61

SWA doesn't show any data from the CMS 61

Remote Device Info table (System > Health) is showing warning against new router 62

SWA doesn't show any telemetry data from a router 62

Telemetry traffic is only showing for one of my connected routers 63

Traffic is entering the network, but the Defense device does not seem to do anything with it 63

Mitigations are not performing the actions I expect 64

To change the Operating Mode to Mitigate 64

CMS shows uncleared alarms 64

Lost administrative user credentials 64

Downloading diagnostic packages 65

After restarting my server, the Corero applications haven't come back up 65

To configure the host to auto-start VMs after a restart 65

Requesting Technical Support 66

Self-Help Online Tools and Resources 66

Creating a Service Request with JTAC 66

CONTENTS

Requesting Licenses **67**

Appendix A – Deploying a vNTD for High Sampled Traffic Rates **68**

Prerequisites **69**

Configure the KVM Host **70**

Enable Intel Virtualization features 70

Isolate CPU cores to enable pinning, and optionally enable Huge Pages 70

(Optional) Performance Optimization suggestions 72

Isolate sampled traffic NICs to prepare for PCI Passthrough 73

Deploy the vNTD for high sampled traffic rates **75**

To deploy a vNTD using virt-install 75

To deploy a vNTD using virsh 76

Pinning vNTD virtual cores to isolated host cores (to complete PCI Passthrough) **77**

Verifying the vNTD is deployed correctly **78**

VM deployment 78

Core allocation 78

Memory capacity 78

Ordering of network interfaces 79

Appendix B – XML Templates for KVM Virsh Deployments **80**

TDD Documentation

There are three main documents which you can use to learn more about the SmartWall TDD:

Document	Location	Use
SmartWall TDD Getting Started Guide	The appropriate guide (KVM or ESXi) is provided by your Support representative or available on the Juniper support portal	Deploy a SmartWall TDD on your own servers. After completing the tasks in this guide, your TDD will be ready for use.
SmartWall TDD User Guide	PDF help from the top menu of the **SWA Web UI** or available on the Juniper support portal	Manage your SmartWall TDD. Contains TDD specific tasks and reference information for the SWA Web UI.
SmartWall TDD CMS User Guide	Context sensitive help site built into the **CMS Web UI** or available on the Juniper support portal	Understand general system tasks, enabling you manage your Defense devices and troubleshoot any issues. Contains reference information for the CMS Web UI, CLI, pCLI and REST API.

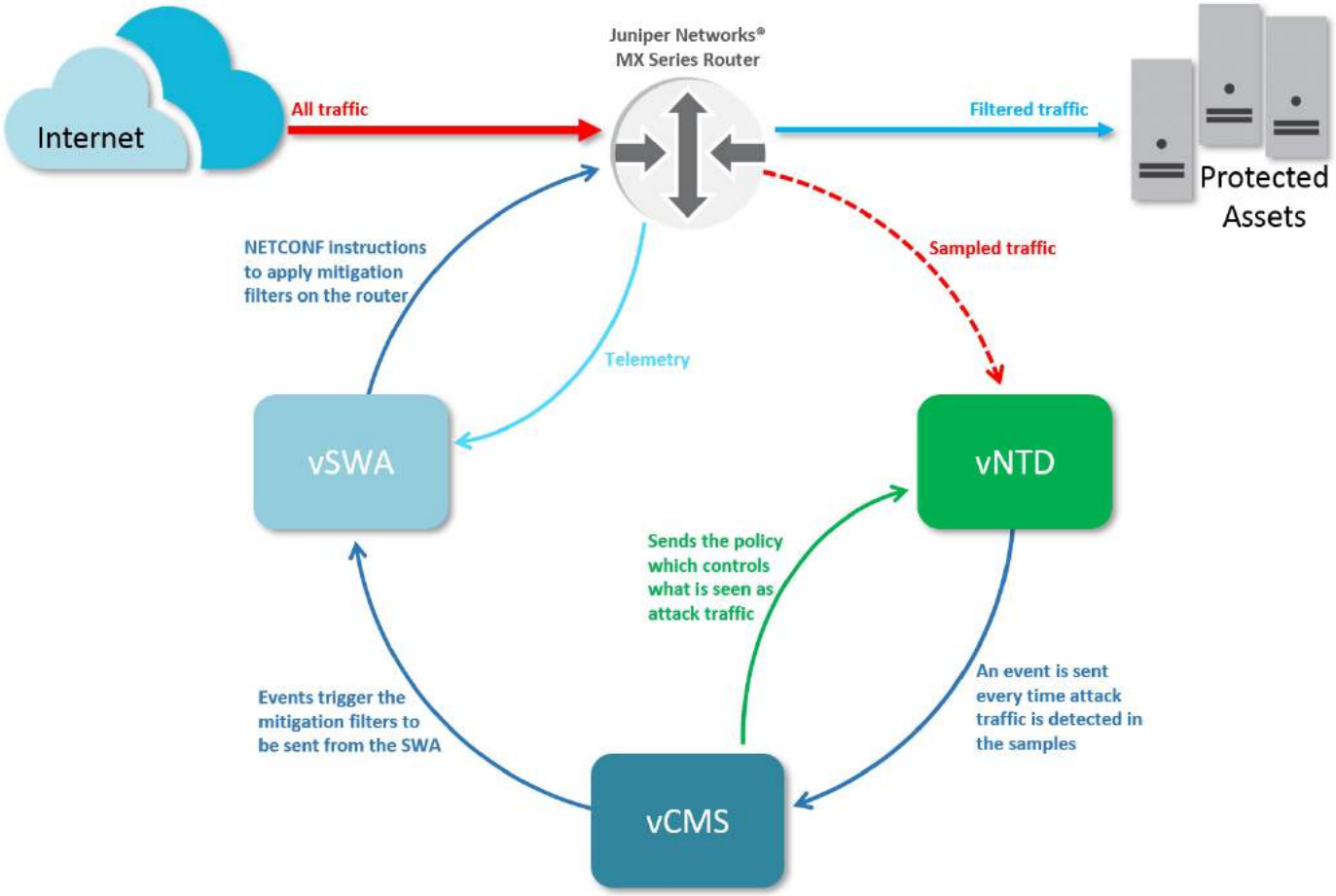
Note: The SmartWall TDD User Guide available from inside the SWA and CMS User Guide available from inside the CMS contain additional information compared to the versions of the guides available on the Juniper Support Portal. This information is only available to customers and is not publicly accessible.

SmartWall Threat Defense Director

The SmartWall Threat Defense Director (SmartWall TDD) works together with Juniper Networks® MX Series routers to filter out DDoS attack traffic at the edge of your network.

A SmartWall TDD system requires the following components:

- **Remote Devices** – The Juniper Networks MX Series router at the edge of the network being protected. They send sampled traffic to the vNTD and are directed by vSWA to apply firewall filters to block DDoS attack traffic.
- **Defense Director** – A bundle of three virtual applications:
  - **vSWA** – The SmartWall SecureWatch Analytics Virtual Edition (vSWA) receives information from the Detection Engine (via the vCMS) to identify the DDoS attacks currently active against your network. The vSWA application then sends firewall filter commands to the router to filter the attack traffic as it arrives at the router. The vSWA application also displays real-time and historical statistics that enable you to analyze attacks on your network.
  - **vCMS** – The SmartWall Central Management Server Virtual Edition (vCMS) controls the Detection Engine and enables you to configure the attack mitigation policy used to distinguish attack traffic from normal network traffic.
  - **Detection Engine (vNTD)** – The SmartWall Network Threat Defense Virtual Edition (vNTD) is the Detection Engine for the SmartWall TDD. It detects DDoS attack traffic in mirrored samples sent from the edge routers.
- **Additional Detection Engines** – The Defense Director bundle includes a single Detection Engine (vNTD). You may need to purchase additional Detection Engines for your deployment.



Working with the SmartWall TDD applications and documentation

The same three applications which power the SmartWall TDD are also used in the Corero SmartWall Threat Defense System (SmartWall TDS). The SmartWall TDS is primarily used inline or in a scrubbing configuration, where the Defense devices block traffic directly. As the system shares common components, you may see the following types of information relating to the SmartWall TDS:

- Some features in the CMS are designed for NTD inline mitigation and will not be available in a SmartWall TDD deployment. When working in the CMS, if you are unsure if a feature applies to the SmartWall TDD, click the help icon in the top left and look for a note labeled **TDD deployments**.
- In the CMS interface, events, and documentation you will see references to "blocking traffic". In a SmartWall TDD deployment, this should be interpreted as "identifying DDoS attacks".

Core Concepts

Provisioning Command Line Interface (pCLI)

When you install a SmartWall device or application, you need to execute essential configuration tasks using the Corero Provisioning Command Line Interface (pCLI). The pCLI is a set of commands you can use to define the initial configuration of each SmartWall® component. For initial configuration of any component, type `setup` in the pCLI to launch a wizard which will guide you through the initial configuration options.

Policy

A Policy is a configuration of the attack mitigation features which tells the Defense devices how to handle incoming traffic. Each policy is contained in a Protection Profile.

Protection Profiles

A Protection Profile is a container for a configuration of the attack mitigation features (Policy) in the CMS. When you associate a Protection Profile with a Cluster, it provides all the Defense devices in that Cluster with the same Policy for handling incoming traffic. You can create one Protection Profile for your network or multiple Protection Profiles each containing a different Policy.

Clusters

A Cluster is a set of identically configured Defense devices. When you create a new Cluster you must associate it with a Protection Profile containing the Policy which controls how the devices in that Cluster respond to traffic.

Devices

There are two types of devices in the SmartWall TDD system:

- **Defense devices** – This is broader term for the vNTDs (SmartWall Network Threat Defense Virtual Edition devices) which are used purely as Detection Engines in a SmartWall TDD deployment
- **Remote Devices** – This is a broader term for the Juniper Networks MX Series router used to mitigate DDoS attack traffic

While the SmartWall TDD only uses the above device types, in the user interface and documentation you should be aware that device can refer to any of the Defense devices compatible with the SmartWall TDS system (vNTD, NTD1100, NTD280, and NTD120) or a Bypass Device.

Segments

A Segment is an interface pair to which DDoS protection is applied. The segment is associated with a port pair on a Defense device. The first time you connect a Defense device to the CMS, it identifies the available interfaces and records them as Segments.

> **Note:** A vNTD has two available interface ports which act as one Segment. For SmartWall TDD deployments only the 1st interface will be used. The 2nd interface should be disabled in the CMS.

Defense Mode

The Defense Mode is the default traffic handling mode which tells the system whether it should use the rest of the Policy features to block attack traffic, just inspect the traffic, or send the traffic to the internal network without any inspection.

For a TDD deployment, when you select a defense mode you have the following options:

- **Mitigate** mode – The TDD system instructs the router to discard attack traffic.
- **Monitor** mode – The router will complete all steps as if it was mitigating traffic (i.e. sending telemetry to SWA) but will accept the attack traffic.

> **Note:** In the CMS documentation and user interface, the Defense Mode is described for an inline SmartWall TDS deployment where the Defense device is able to directly block traffic. In the TDD system the blocking is only ever performed by the routers. Pass-though mode only applies to the TDS system.

Analytics

Analytics is the process of collecting and analyzing the event and system information generated by the Defense devices. The Defense devices send analytics syslog messages to the CMS where that information is aggregated and sent to SWA.

Sampled Traffic

This is a feed of a proportion of the traffic received by the Juniper Networks MX Series router ahead of any mitigation. The vNTD uses this traffic to detect DDoS attacks, and enables the TDD system to generate the filter instructions it sends to the Remote Device to block that attack traffic and permit non-attack traffic. For example, if you have 1Tbps of traffic coming into a Remote Device, and a sample rate of 1:1000, the vNTD will see 1Gbps of sampled traffic.

> **Caution**: Do not use truncated samples on the Juniper Networks MX Series router.

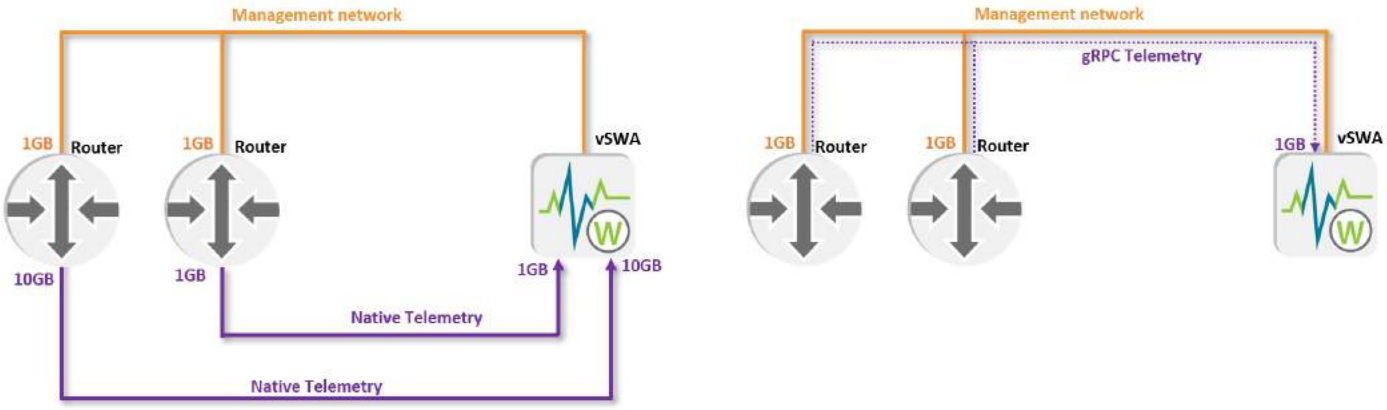
Telemetry

Telemetry is sent from the Juniper Networks MX Series router to the vSWA. It shows the network traffic processed by the router including what was permitted or blocked by the TDD system.

The TDD requires a telemetry feed from every monitored router to the SWA application. There are two main telemetry delivery methods:

- **Native telemetry** (UDP) – Telemetry is sent over your traffic network between the router and SWA. This requires a dedicated 10GB port on the router and a 10GB, or 1GB, port on the SWA host.
- **gRPC telemetry** – Telemetry is sent over the 1GB Management network. With gRPC telemetry you have the option to encrypt the telemetry traffic using SSL certificates.

You decide which telemetry type is used when you configure the Juniper Networks MX Series router. If you choose gRPC telemetry, you must download 2 additional software files to the router during set up and then provide additional configuration information when adding the router to the SWA as a remote device.



NETCONF

The TDD system uses NETCONF to configure the ephemeral firewall rules in the Juniper Networks MX Series router to block or permit network traffic.

SmartWall Service Portal

The SmartWall Service Portal enables you to offer Corero SmartWall DDoS Protection, as a managed service, to your customers. The Service Portal is a customer-facing DDoS protection portal which uses traffic data from your SmartWall TDD and displays the information in easy to read dashboards and reports. Your customers can log in to the portal and view the attacks you have protected them against. For information on Service Portal versions which are compatible with your SmartWall TDD, see the SmartWall TDD release notes.

> **Note:** If you do not have a Service Portal and would like to add one to your existing TDD system, contact your support representative for more information.

Deployment Checklist

The following checklist provides an overview of the deployment process. Verify each step is complete before moving on:

Location	Action	Done?
On your PC	Save the TDD license file you received when you purchased the TDD.	
On the host server	Setup host server(s) and make any network changes.	
On the host server	Download the following install files from https://corero.force.com : corero-ntd-virtual-edition_9.7.5.xxxx-kvm.zip, corero-cms_9.7.5.xxxx-kvm.zip, corero-swa_9.7.5.xxxx-kvm.zip	
On the host server	Deploy vSWA and use the pCLI to setup management and secondary network interfaces.	
On the host server	Deploy vCMS and use the pCLI to setup management interface.	
On the host server	Deploy vNTDs (or high sampled traffic rate vNTDs) and use the pCLI to setup management interfaces.	
On your PC: SWA web UI	Upload TDD license file to SWA.	
On your PC	Request your vNTD Licenses and save files.	
On your PC: CMS web UI	Upload the vNTD license to the CMS.	
On your PC: CMS web UI	Add vNTD to CMS.	
On your PC: CMS web UI	Disable 2nd interface and disable LSP for each vNTD.	
On your PC: CMS web UI	Setup the connection between CMS and SWA.	
On router CLI	Configure routers to send traffic samples to the vNTDs and receive filters from SWA.	
On your PC: CMS web UI and **SWA web UI**	Add router authentication credentials to CMS and SWA.	
On your PC: CMS CLI	Configure default defense policy for TDD system.	

Location	Action	Done?
On your PC: CMS web UI and **SWA web UI**	Tune Policy for your network and, optionally, setup multiple Clusters and Protection Profiles.	

Caution: When the TDD system is installed, it is recommended to set it to Monitor Mode (the CMS Defense Mode which shows how the system will affect traffic when mitigating but does not block any traffic). When you have evaluated and tuned the CMS Policy for your network traffic, you can then switch the system into Mitigate Mode and begin blocking DDoS attack traffic.

Network Configuration for TDD Deployment

To operate correctly, a SmartWall TDD installation will require network configuration settings and certain ports to be available between components.

- You may want to have the IP addresses ready for any other syslog clients you intend to use.
- You must allow outbound TCP port 443 traffic to enable the TDD to connect to the Corero License server. To establish licensing, you must install a SecureWatch package on the vSWA.
- When hosting a SmartWall TDD, truncated samples should not be used on the Juniper Networks MX Series router. The JunOS command `set forwarding-options port-mirroring maximum-packet-length` should have a value of `0` (disabled).

Securing your SmartWall applications

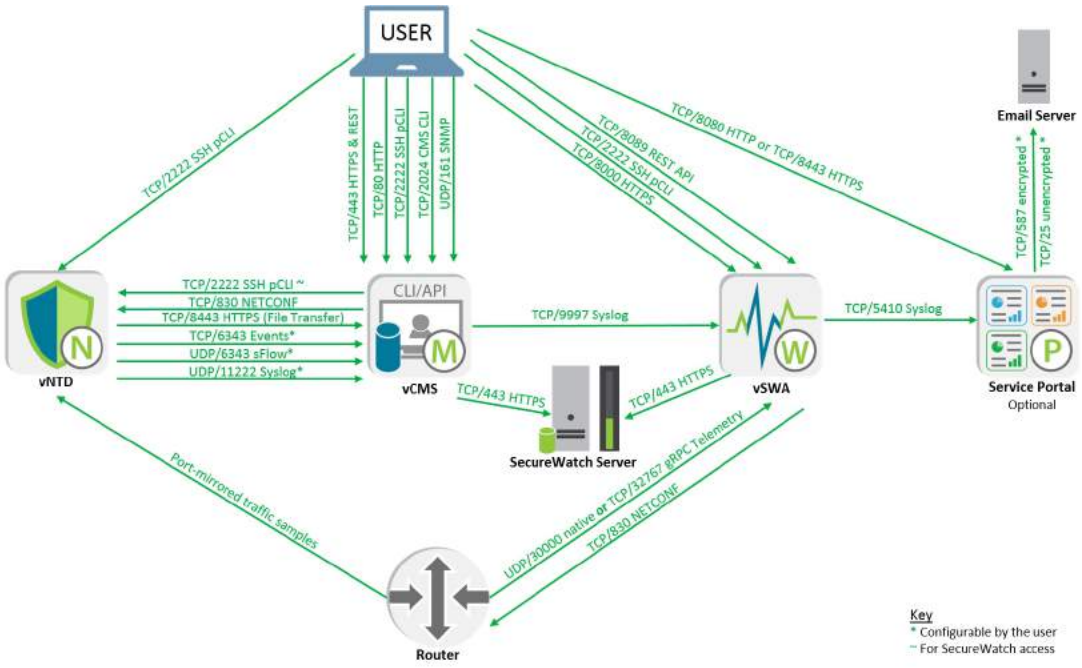
Corero recommends following best practice industry standards to secure your SmartWall deployment:

- Connect management interfaces to secure networks, isolated from the Internet, to prevent unauthorized access to the SWA, CMS, or NTD.
- If practical, run Corero CMS and SWA virtual machines on a dedicated server to physically isolate them from other guest virtual machines.
- If running CMS and SWA on shared virtual infrastructure, ensure you apply latest security fixes as they emerge. This may require:
    - Patching the server BIOS for new CPU firmware.
    - Patching the hypervisor.
    - Patching other guest virtual machines operating systems
    - Patching other guest VMs and applications that may be running on the same server as Corero management components.
- Limit access to the CMS and SWA to authorized personnel. Ensure adequate access controls are in place.
- Ensure authorized personnel are associated with the correct access roles on the CMS and SWA.
- Use strong access credentials, regularly changed, through authentication service such as LDAP or RADIUS to authenticate access to your users.
- Limit knowledge of privileged account credentials.
- After deployment, change default NTD credentials and manage using Authentication Groups in the CMS. **Note:** The same NTD credentials are used for managing NTDs from the CMS and for accessing the NTD pCLI.
- Apply the latest software updates from Corero which contain the latest available security patches.

Firewall Considerations

Your SmartWall deployment should be protected by your network infrastructure's firewall. You will need to modify your firewall policy to allow access on certain ports.

The following diagram shows the default network ports required for a full SmartWall TDD deployment:



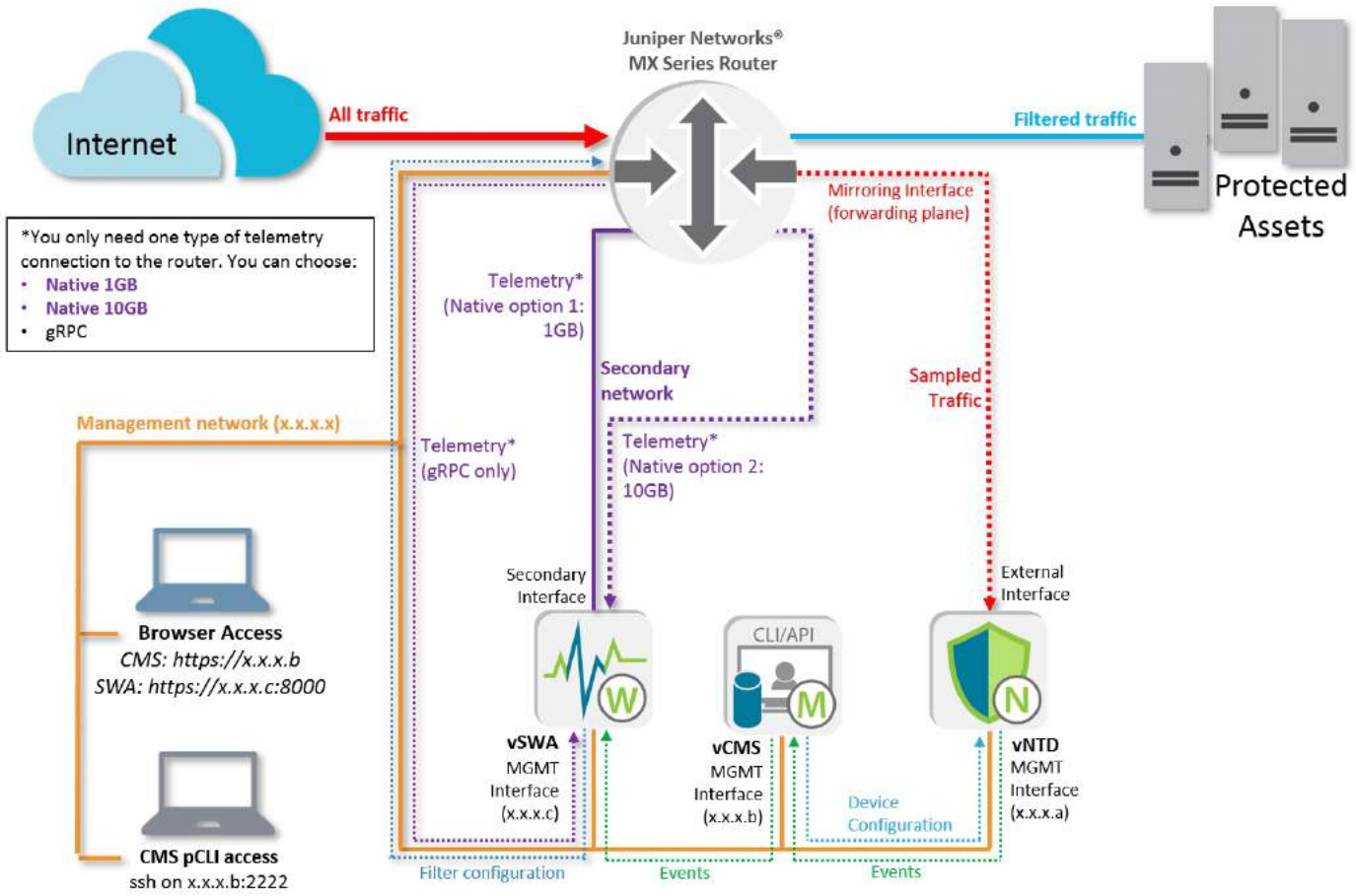
Note: If your SmartWall TDD comes with Corero's SecureWatch Service, this also requires a change to your firewall policy. Contact your support representative for the required information.

SmartWall TDD Deployment

The following diagram shows a common deployment scenario for the SmartWall TDD. If you require further assistance configuring the best deployment for your network, contact your support representative.

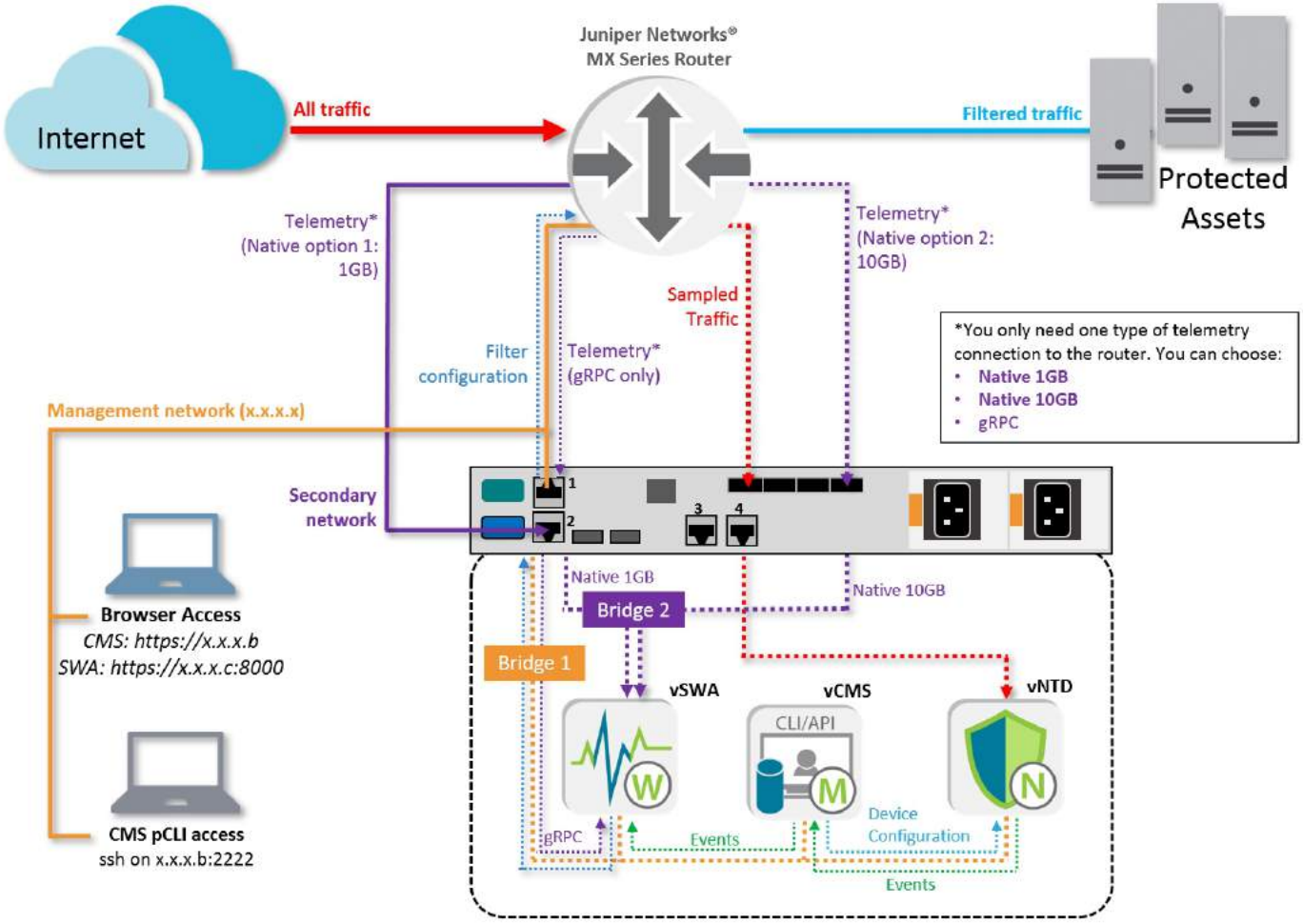
Separately hosted TDD applications managing one Juniper Networks MX Series router

In this scenario, the three SmartWall TDD applications are all installed separately, on the same management network as the router they are receiving sampled traffic from. The diagram shows the three possible routes for telemetry to be passed from the router to the SWA.



TDD applications, on one host, managing one Juniper Networks MX Series router

In this scenario, the three SmartWall TDD applications are all installed on a single server and are receiving sampled traffic from a single router. The diagram shows the three possible routes for telemetry to be passed from the router to the SWA.



Deploying vNTDs for Different Sampled Traffic Rates

There are multiple ways to deploy a vNTD on your own server. If you expect a high sampled traffic rate, you must dedicate hardware resources on the host for the vNTD. If you expect a lower sampled traffic level, you can speed up your installation by skipping those steps.

Sampled Traffic Rate (sampled mirror traffic of mixed packet sizes)	KVM
Sampled traffic rates up to 250 Kpps or 1.2Gbps per core . **Note:** With a sample rate of 1-1000, this equates to up to 1.2 Tbps of protected traffic (or up to 250 million pps).	NICs: VirtIO (single core) 2 core vNTD No PCI Passthrough
Sampled traffic rates between 1.2Gbps and 10Gbps	NICs: X710/X520 7 core vNTD PCI Passthrough required (see Appendix A)

Caution: For the vNTD External Interface, e1000 NICs are not supported. You will experience issues viewing some statistics like packet drops and bad CRC packets. Contact your support representative for more information on choosing a NIC.

Using this document to deploy a vNTD

The main deployment method focuses on vNTDs expecting sampled traffic rates of less than 1.2Gbps, which do not need dedicated hardware resources to achieve necessary performance. Follow the standard vNTD deployment instructions.

If you need to handle up to 10Gbps sampled traffic rates, you should use the alternative high sampled traffic rate vNTD deployment guides in Appendix A using PCI Passthrough.

Deploying the TDD Virtual Components

The virtual components of the SmartWall Threat Defense Director (TDD) can be installed on your own KVM server. Your Corero representative will have provided you with the files you need for KVM installation.

Juniper Networks MX Series router requirements

Your Juniper Networks MX Series router must meet the following criteria:

- It must support Sampled Mirror, Flexible Filtering, Ephemeral Configuration, and Remote Telemetry.
- Your router should be running one of the following JunOS versions:
    - For production deployments:
        - 17.2R3
        - 17.3R3
        - **17.3R3-S8 recommended**
        - 17.4R2
        - 18.1R3
        - 18.2R2
        - 18.3R1
        - **18.3R3-S2 recommended**
        - **19.2R3 recommended**
        - **20.1R2 recommended**

        Note: Recommended versions have had a broad and successful use with Corero SmartWall TDD.

    - For lab tests or proof of concept deployments:
        - Any of the above
        - 16.2R3 minimum

Caution: For JunOS versions not listed, please refer to your support representative for compatibility.

Virtual Editions components

You should have the following files saved locally (where xxxx represents the version numbering section of the file name):

- corero-ntd-virtual-edition_9.7.5xxxx-kvm.zip – The vNTD zip folder containing the vNTD disk image
- corero-cms_9.7.5.xxxx-kvm.zip – The vCMS zip file containing the CMS disk image

- corero-swa_9.7.5.xxxx-kvm.zip – The vSWA zip file containing the two SWA disk images

Required information

You need to have the following network information available for each virtual component before you begin to install:

- **IP address and subnet mask** – The IP address you will use to access the application from your core network.
- **DNS IP address** – The address of the DNS server for your site. You may have more than one of these.
- **Default gateway IP address** – The address of the default gateway for your site.
- **NTP IP address** – The address of the NTP server for your site. You may have more than one of these. Corero can provide you with an NTP server address if you do not have one of your own.

For vCMS and vSWA, you also need **A SecureWatch® package file and unlock code**. To connect your CMS and SWA applications to the SecureWatch® Service, you will need to upload a SecureWatch® package file to both applications after deployment.

System requirements

Host requirements

Before you begin, make sure the server you plan to deploy an application on meets the following requirements:

	Defense Director (vCMS)	Defense Director (vSWA)	Detection Engine (vNTD)
Supported OS	Linux server (Red Hat Enterprise Linux 7, Centos 7, Ubuntu 16.04, or Debian 9.9) using kernel-based Virtual Machines (KVM)	Linux server (Red Hat Enterprise Linux 7, Centos 7, Ubuntu 16.04, or Debian 9.9) using kernel-based Virtual Machines (KVM)	Linux server (Red Hat Enterprise Linux 7, Centos 7, Ubuntu 16.04, or Debian 9.9) using kernel-based Virtual Machines (KVM)
Memory	ECC memory is required.	ECC memory is required.	ECC memory is required.
CPU	The CPU must support VT-x.	The CPU must support VT-x.	The CPU must support VT-x (and VT-d for high sample traffic rates). They must be enabled.
Datastore	Datastore provided by redundant RAID storage: • Minimum SAN or NAS datastore • Recommended VirtIO	Datastore provided by redundant RAID storage: • Minimum SAN or NAS datastore • Recommended VirtIO	Datastore provided by redundant RAID storage: • Minimum SAN or NAS datastore • Recommended VirtIO

	Defense Director (vCMS)	Defense Director (vSWA)	Detection Engine (vNTD)
NICs	Management NIC: e1000 or virtIO	Management NIC: e1000 or virtIO	Management NIC: e1000 or virtIO External Interface: • For low sampled traffic rates – VirtIO • For high sampled traffic rates – X710 (recommended to use firmware 6.8 or higher) or equivalent NICs utilizing VT-d directed I/O technology (PCI Passthrough).

KVM Host requirements

To host any of the three applications, the KVM host must meet the following criteria:

- KVM must be installed on the host OS
- libvirt must be installed on the host OS
- libvirtd must be running on the host OS
- QEMU should be running on the host OS

Virtual appliance requirements

The table below lists the minimum requirements for each virtual appliance that you install.

	Defense Director (vCMS)	Defense Director (vSWA)	Detection Engine (vNTD)
Memory	Minimum **8 GB** of memory.	Minimum **12 GB** of memory.	Minimum **12GB per socket** (with seven 1GB -or equivalent- huge pages available for high sample traffic rates).

	Defense Director (vCMS)	Defense Director (vSWA)	Detection Engine (vNTD)
CPU	Intel® Xeon® 64-Bit CPU (x86_64) Processor minimum 2GHz vCMS must have **4 cores**	Intel® Xeon® 64-Bit CPU (x86_64) Processor minimum 2GHz vSWA is recommended to have **12 cores**	Intel® Xeon® 64-Bit CPU (x86_64) Processor **from E3 family or later** with minimum 2GHz vNTD must have the correct core number for its external NIC type: • VirtIO NIC – **2 cores** • X710 (recommended to use firmware 6.8 or higher) or equivalent NIC – **7 or more cores**
Datastore	Minimum **200 GB** datastore. Recommended **1 TB**.	Minimum **400 GB** datastore. Recommended **1 TB**. Disk defaults to 240GB on install but can be extended to the recommended size. **Caution:** After extending the disk size, indexes must be resized.	Required **20 GB** datastore.

Additional Requirements

Observe the following additional requirements when deploying virtual appliances:

- Changing of virtual hardware components or updating the virtual hardware version is not supported.
- To avoid risk of overcommitting allocated storage during production, and causing a failure, qcow disks should be converted to **preallocation=falloc** (See prerequisite section at the beginning of application deployment).
- You need at least one management network connection with connectivity to SmartWall devices, CMS and SWA. The SWA requires a secondary network connection on the same subnet as the router's telemetry port, to receive native telemetry from the routers.
- All TDD applications must use NTP servers for system time. Any time difference between applications can cause unexpected behavior.
- NUMA Memory/CPU Affinity is recommended but not required on a NUMA system.

Deploying a virtual edition on a KVM server

The following instructions cover how to deploy a vNTD, vCMS or vSWA and assumes you already have the required software installed on a KVM server. See the Appendix for deployment instructions for vNTDs expecting high inspection rates.

Prerequisites

Prepare the disk images for deployment:

- Save and export the zip files for the 3 application (*vntd_9.7.5.nnn-kvm.zip*, *vcms_9.7.5.nnn-kvm.zip* and *vswa_V9.7.5.nnn-kvm.zip*),. This creates a folders containing the required qcow2 disk images.
- Modify the disk images to use preallocation=falloc (the equivalent of thick provisioning the disk to ensure correct storage allocation) and rename the files for ease of deployment. There is 1 disk image for the vNTD and CMS, and 2 disk images for the SWA. Use the following command to modify each disk image:
  `qemu-img convert -f qcow2 <disk image file name> -O qcow2 <newFileName>-disk [1|2].qcow2 -o preallocation=falloc`
  For example, converting the first vSWA disk image you would use something like the following command:
  `qemu-img convert -f qcow2 vswa_9.7.5.133-030-disk1.qcow2 -O qcow2 new-swa-disk1.qcow2 -o preallocation=falloc`

> **Tip:** To free up space on your host, once you have the modified the disk images, you can delete the zip files and original qcow2 images. For deployment, you only need the modified and renamed qcow2 disk images: 1 for vNTD, 1 for CMS, and 2 for SWA.

To deploy a Corero Virtual Appliance using virt-install

> **Caution:** When you run the commands below, they should not contain line breaks. Copying and pasting directly from the PDF may create unnecessary line breaks. Always paste into a text editor first and edit the commands.

Before you run the install command, **libvirtd** must be running. Run the required command (below), replacing the following information:

- *<vmName>* – A name for this VM.
- *<qcow2DiskName1>* – The file name of the qcow2 disk (e.g. `new-swa-disk1.qcow2`).
- *<qcow2DiskName2>* – **(vSWA only)** The file name of the additional data-disk qcow2 disk (e.g. `new-swa-disk2.qcow2`).

> **Tip:** If you didn't save the disk files in the same location you're creating the VM, you can add a file path to the file name field (e.g. `/home/kvmfiles/vswa_9.7.5.133-030-disk1.qcow2`)

- *<diskType>* – The type of disk you want to use: `ide`, `virtio`, `sata`.
- *<managementInterface>* – The name of your management interface (e.g. eth0).
- *<telemetryInterface>* – **(vSWA only)** The name of the interface on your host reserved for telemetry from the router (e.g. eth1). This will be the secondary interface on the SWA.
- *<ExternalInterface>* – **(vNTD only)** The name of the interface on your host reserved for external traffic coming to the vNTD (e.g. eth2).
- *<ExternalInterface>* – **(vNTD only)** The name of the interface on your host reserved for internal traffic coming to the vNTD (e.g. eth2).

vSWA:

```
virt-install -n <vmName> -r 12288 --os-type=linux --os-variant=rhel7 --disk
<qcow2DiskName1>,device=disk,format=qcow2,bus=<diskType> --disk
<qcow2DiskName2>,device=disk,format=qcow2,bus=<diskType>,size=500 --vcpus=12 -w
source=<managementInterface>,type=direct,source_mode=bridge,model=virtio -w
source=<telemetryInterface>,type=direct,source_mode=bridge,model=virtio --import
--vnc --noautoconsole
```

vCMS:

```
virt-install -n <vmName> -r 8192 --os-type=linux --os-variant=rhel7 --disk
<qcow2DiskName1>,device=disk,format=qcow2,bus=<diskType> --vcpus=4 -w
source=<managementInterface>,type=direct,source_mode=bridge,model=virtio --import
--vnc --noautoconsole
```

vNTD:

```
virt-install -n <vmName> -r 12288 --os-type=linux --os-variant=rhel7 --disk
<qcow2DiskName1>,device=disk,format=qcow2,bus=<diskType> -w
source=<managementInterface>,type=direct,source_mode=bridge,model=virtio -w
source=<ExternalInterface>,type=direct,source_mode=bridge,model=virtio -w
source=<InternalInterface>,type=direct,source_mode=bridge,model=virtio --import
--vnc --noautoconsole --cpu host-passthrough --vcpus
sockets=1,threads=1,cores=2,placement=static,vcpus=2, --memballoon=none
```

> **Tip:** The commands above create a new bridge for each interface. If you already have bridges (and their assigned interfaces) configured on your host, you can replace the `-w` networking commands with the following syntax which enables you to specify the bridge you want to use rather than the interface: `-w source=<BridgeName>,type=bridge, model=virtio`

Verify deployment

After you use a virt-install command, you should verify the virtual machine deployed correctly. Use the command `virsh list --all` to see all VMs currently running. You should see the new VM in that list.

To deploy a Corero Virtual Appliance using virsh

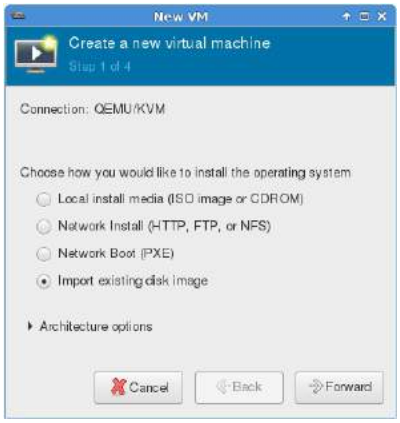
> **Caution:** Deployments using virsh do not support a serial console.

1. Create a new XML file and give it the name you want to use for the vNTD VM (*<vmName>*.xml).
2. From the appendix in this guide, copy the relevant template into your XML file and replace the green place-holders with the necessary information for your new VM.
3. Type the following command to create a VM from the xml file: `virsh define <vmName>.xml`
4. Type the following command to start the VM : `virsh start <vmName>`

To deploy a Corero Virtual Appliance using virt-manager

> **Note:** virt-manager is not the recommended deployment method for a vNTD as it can create additional VM features which are not required and may adversely affect performance. For optimum performance the virt-install or virsh method should be used.

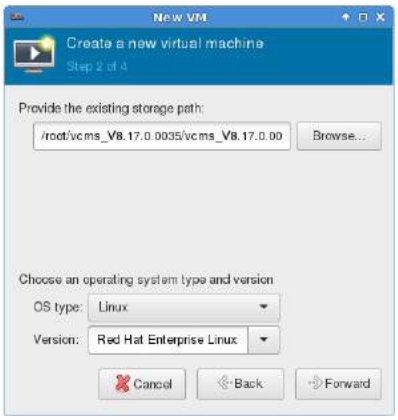
1. Use the following command to open the virt-manager GUI: `virt-manager`
2. Click on the **Create new virtual machine** button to open the New VM wizard.
3. Select **Import existing disk image**. Click **Forward**.



4. Click **Browse**. Then click **Browse Local** and open the folder containing the disk images. Select the image for disk 1, then click **Choose Volume** or **Open**.

5. Select:
    - **OS type** – Linux
    - **Version** – This will vary depending on your system (e.g. Ubuntu 16.04 or Centos 7)

    Click **Forward**.



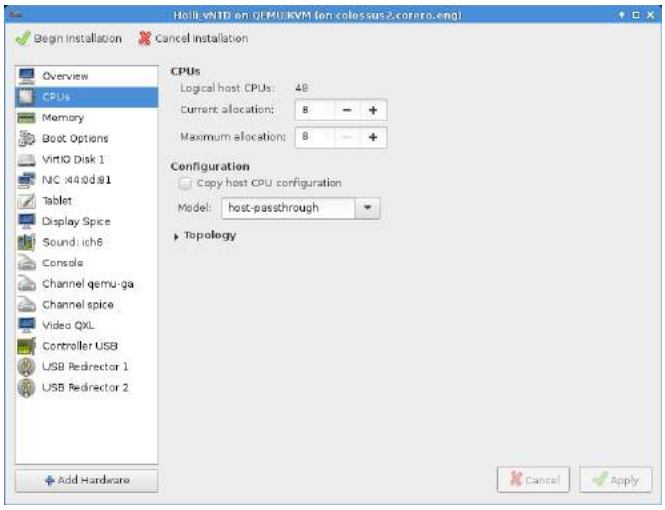
6. Set:
    - **Memory (RAM)** to minimum recommendations or higher:
        - 8192 for a vCMS or vNTD deployment
        - 12288 for a vSWA deployment
    - **CPUs** to minimum recommendations or higher:
        - 4 for a vCMS
        - 12 vSWA deployment
        - For a vNTD deployment, select the cores required for your NICs:
            - VirtIO – 2 cores
            - X710 or equivalent – 7 cores

    Click **Forward**.

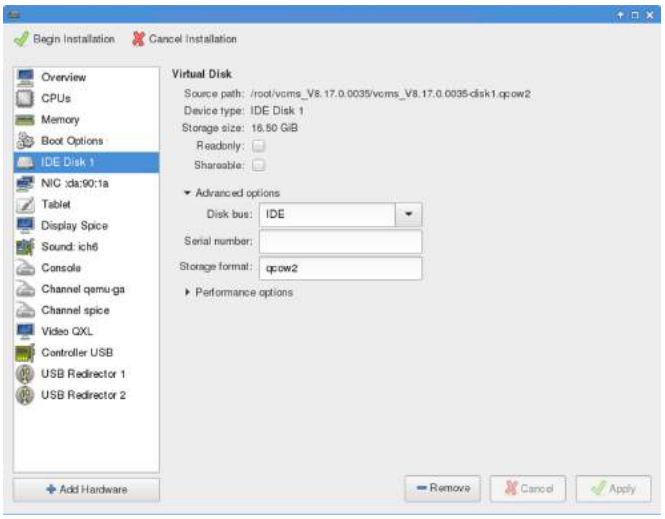
7. Type a **Name** for your new VM and select **Customize configuration before install**.

8. Expand **Network selection** and, from the drop-down, choose the network port you're mapping to. Click **Finish**.

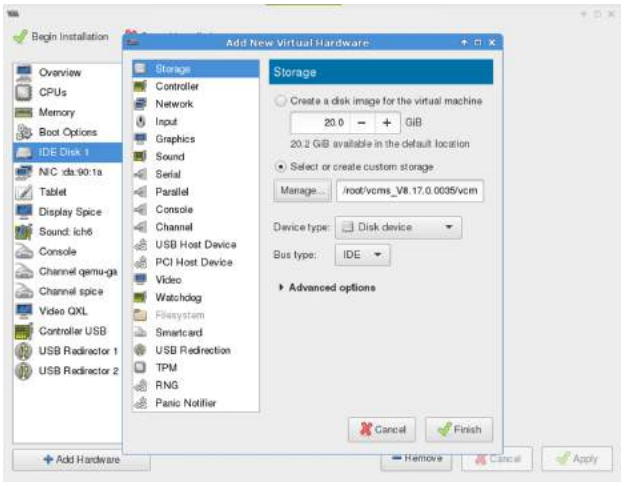
9. **For vNTD only:** From the list on the left, select **CPUs** and, under **Configuration**, click **Copy host CPU configuration**. In the **Model** field, type `host-passthrough`.



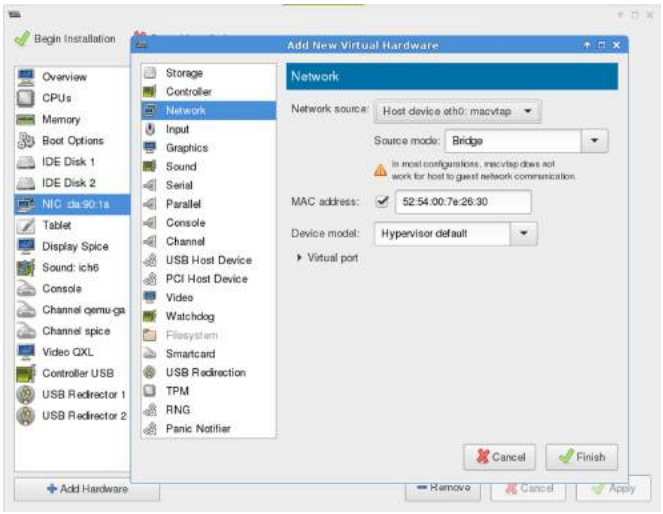
10. From the list on the left, select the first virtual disk and expand **Advanced options**. From the **Disk bus** drop-down select the disk type you want to use (**IDE**, **virtio**, or **SATA**) and make sure that the **Storage format** is qcow2. Click **Apply** if you had to make any changes.



11. **For vSWA only:** Add the second disk image:
    a. Click **Add Hardware** and select **Storage**.
    b. Select **Select or create custom storage**, then click **Manage**. Navigate to the same folder you got the first disk image from (it should now be available here, without having to browse locally) and select the image for disk 2. Click **Choose Volume** or **Open**.
    c. From the **Disk bus** drop-down, select the disk type you want to use (**IDE**, **virtio**, or **SATA**) and click **Finish**.



12. **For vSWA only:** To add the secondary interface, click **Add Hardware** and select **Network**. Select the appropriate **Network Source** and click **Finish**.



13. **For vNTD only:** (Optional, for bridging the management network only) From the list on the left select the NIC. Check the correct Network source is selected and, from the Source mode drop-down, select **Bridge**. Click **Apply** if you had to make any changes.
14. **For vNTD only:** To add the first NIC, click **Add Hardware** and select **Network**. Select the appropriate **Network Source** and click **Finish**. Repeat this step to add the second NIC.
15. Click **Begin Installation**.

(vNTD only) Verify ordering of network interfaces

All interfaces on the vNTD are treated as PCI devices and each interface has a PCI-ID value which identifies the interface. The management interface (eth0) is assigned to the interface with the lowest PCI-ID value.

1. Access the vNTD pCLI. Note: The vNTD application may take 2 minutes to initialize when first installed.
2. Type the command: `show nic`
3. Identity the PCI Address shown for the Management interface.
4. Return to the host.
5. Type the following command to open the VM XML file: `virsh edit <vmName>`
6. Check the PCI address of the management interface matches the one from the pCLI.

If your PCI addresses match, then the management interface is allocated as expected. If they do not match, then your expected management interface did not have the lowest PCI-ID value. Edit the VM XML file to use the correct management interface PCI ID.

Configuring the TDD Components

After deploying the three SmartWall TDD components (vNTD, vCMS, and vSWA), there are some immediate configuration actions you need to complete to get them fully functional. Each component needs to be setup using the pCLI, then you must connect the CMS to the SWA and add the vNTD to the CMS.

Accessing the pCLI on a virtual appliance

After deployment, you need to configure the virtual appliance. To do this you need to access the pCLI and run the setup wizard:

1. Open a console session to the virtual appliance:
   - **KVM server** – Type the following command: `virsh console <vmName>`
2. Log in to the virtual appliance using the default user account (`admin`) and password (`smartwall`).

> **Tip:** Once you have an IP address associated with the appliance, you can use SSH on port 2222 using the default admin user account (e.g. `ssh -p 2222 admin@<ipAddress>`) and type the default password `smartwall` when prompted.

Resizing a vSWA application

The default disk image for a vSWA deployment is 249GB. You can choose to resize disk two following deployment:

1. After deployment, make sure the vSWA VM is running, and access the pCLI.
2. Enter the following command to check the current disk size: `show data-disk`. Make a note of the current disk size and exit the pCLI.
3. Enter the following command to shut down the vSWA VM: `virsh shutdown<swaVmName>`
4. Check the VM has stopped: `virsh list --all`
5. When you resize the data disk of SWA, you can add or subtract from the current disk size. In this example command we add 500G (replace the example disk 2 file name with your own): `qemu-img resize new-swa-disk2.qcow2 +500G`
6. Enter the following command to restart the vSWA VM: `virsh start<swaVmName>`
7. Access the pCLI. If you enter the `show data-disk` command at this stage you will still see the previous disk size; that is expected.
8. Enter the following command to re-partition the disk to match the new size: `setup data-disk`
9. You should see something similar to this message: `Found 500 GB of unallocated space. Do you want to continue? <Y,[N]>:`
10. Enter `y` to repartition the disk.
11. You should then see the option to autosize your indexes. Enter `y` to continue.
12. If you are happy with your changes, when prompted enter `A` to accept and apply the changes.

13. You can check this was successful by using the command `show data-disk`, it should now display the updated size.

Configuring SmartWall Components Using the pCLI

This section describes how to use the provisioning CLI (pCLI) to set each component's basic configuration. To access the pCLI for each SmartWall® component see the application specific instructions earlier in this guide.

The pCLI provides the initial interface for configuring SmartWall™ components. Once you use the setup wizard to configure the application, you should perform all other tasks in the corresponding web interfaces. You can return to the pCLI if you need to edit these basic configuration settings later.

> **Tip:** You can type `help` at the pCLI console command prompt to see a list of all the other available pCLI commands and Tab-completion is supported, enabling you to type a portion of the command that identifies it uniquely and press **Tab** to finish it.

Using the pCLI Setup Wizard to Configure a SmartWall Component

> **Note:** This example shows the use of the pCLI to run the setup wizard on the vCMS. Other setup wizard sessions may vary slightly, depending on the component.

To use the setup wizard:

1. Open the pCLI for the component. When you log in to the pCLI, you will see something similar to the following example:

```
Corero SmartWall Central Management Server
Copyright (C) 2016-2017 Corero Network Security, Inc. All rights reserved.

cms login: admin
Password:
Last login: Thu Apr 28 10:42:38 on tty1

Welcome to the Corero Network Security initial setup CLI.
Please type 'setup' to start the initial configuration wizard.
For details of further options, please type 'help'.
cms>
```

2. Type `setup` to start the initial setup wizard, which will prompt you to change a number of basic configuration settings. For each group of settings, type `A` to accept changes you've made, type `C` to go back and change a setting, or type `E` to leave the current settings unchanged.

```
cms>setup
You will be asked a number of questions, along with the current values
in brackets. Please type in the desired configuration or you may press
Enter to keep the current value. After each section, you will be given
a chance to confirm and apply your configuration.
```

3. Change the username and password from their defaults. This is strongly recommended. If you are configuring SWA, you will be asked if you want to enable LDAP, as well.

```
 Please configure the authentication settings:

 Warning: Applying changes here will overwrite any users created in the
application

 Enter user-name for the administrative account [admin]:
 Enter password (press Enter to leave unchanged):

 Enter [C]hange or [E]xit without changing [C]: E
```

4. Configure the management port settings: MTU and DHCP. If you choose not to use DHCP, specify the management IPv4 address, management IPv4 subnet mask, and management IPv4 default gateway (Note: All SmartWall applications require IPv4 addresses for management. IPv6 is not supported).

```
Please configure the Management Interface:
 Enter MTU [1500]:
 Enable DHCP? Y,[N]: N
 Enter IPv4 Address [10.20.27.100]:
 Enter IPv4 Subnet Mask [255.255.255.0]:
 Enter IPv4 Default Gateway [10.20.27.254]:
```

5. vSWA only: You must set up your secondary interface to receive native telemetry from the Juniper Networks MX Series router. The IP address for the secondary interface must be on the same subnet as the Telemetry port on the router. The Default Gateway should be left blank for most TDD deployments.

```
Please configure the optional Secondary Interface:
Enable interface? Y,[N]:
Management Interface:
MTU : 1500
IPv4 Address : 192.168.54.13
IPv4 Subnet Mask : 255.255.255.0
IPv4 Default Gateway :

Secondary Interface:
State : Disabled
```

6. Configure the DNS settings: hostname, and the option for manual DNS configuration (DNS servers, DNS domain, DNS search domains).

```
Please configure the Domain Name System (DNS) settings:
Enter hostname [cms]:
Enter primary DNS server [None]:
Enter DNS domain [None]: corero.com
Enter DNS search domains (separated by space) [None]:

Configuration:
Hostname : cms
DNS Servers : None
DNS Domain : corero.com
DNS Search Domains :

Enter [A]ccept, [C]hange, or [E]xit without saving [C]: A
```

7. Configure the time settings: NTP server (primary, secondary, and tertiary) and time zone.

> **Caution:** You must use NTP servers for all TDD applications. Time differences between applications can cause unexpected behavior.

```
Please configure the time settings:
Would you like to enable NTP? Y,[N]:
Enter time zone or '?' for complete list [America/New_York]:

Configuration:
Time source : Hypervisor
Time zone : America/New_York

Enter [C]hange or [E]xit without changing [C]: E
cms>
```

8. After you complete the basic configuration using the pCLI, you can review your changes by typing the command `show`.

```
cms>show

 Management:
 State : Enabled
 Link State : Up
 MAC Address : 00:50:56:88:62:82
 MTU : 1500
 DHCP Enabled : No
 IPv4 Address : 10.20.27.100
 Subnet Mask : 255.255.255.0
 Default Gateway : 10.20.27.254

 Secondary:
 State : Disabled
 Link State : Down
 MAC Address : 00:50:56:88:20:ad
 MTU : 1500
 IPv4 Address : None
 Subnet Mask : None


 SecureWatch Status:
 State : Disabled
 SecureWatch ID : No package loaded
 cms>
```

(Optional) SmartWall SecureWatch Analytics Considerations

The following pCLI options are specific to configuring the SWA:

SSL Certificates

The pCLI can be used to load a signed SSL certificate for use with the web UI, to avoid the browser security warnings which appear when you sue the default unsigned certificate. The certificate must be in PKCS#12 format, and include the key, certificate, and CA certificate change to be used for SSL. The common name should match the hostname assigned to the SWA appliance.

To load a certificate, type `ssl-certificates https` followed by the URI to the certificate file. The supported protocols are FTP, SFTP, HTTP, and HTTPS. For example:

```
ssl-certificates https sftp://admin@10.20.30.40/certs/my_cert.p12
```

Installing the TDD license file

You must install a TDD license file in the SWA application as it enables the SWA to check you are licensed to use the TDD system. The license file is provided in a SecureWatch Package format which is compatible with the SWA application upload process.

> **Note:** A SecureWatch Package File without a SecureWatch Service connection does not create a persistent connection to the license server. It only requires a periodic check and no data is ever sent over that connection. If you have the SecureWatch Service included in your package, this also creates a secure VPN connection enabling attack monitoring from SecureWatch Operations Center.

1. You will receive the SecureWatch package and unlock code from Customer Support.
2. Save this file in a location that you can easily access from the computer you're using to access the SWA web UI.
3. Open the SWA Web UI in a browser.
4. Navigate to **System** > **SecureWatch Packages**.
5. Click **Choose File** and select the saved package file.
6. If required, type in the **Unlock Code**.
7. Click **Install Package**. This will cause the SWA to restart and may take several minutes. When it's complete you can log back in.

Uploading a vNTD license to the CMS

For each vNTD you want to add to your CMS, you need a 10G license. After your vCMS is deployed and configured, you can contact support for a license file unique to your system.

1. In the CMS Web UI, you can see your CMS's UUID at the top of the **Home** screen.
2. Contact your support representative to request a vNTD license file . You must quote your CMS UUID and your Juniper SSRN (the SSRN will be on your sales agreement information).

   Caution: License files are created to be specific to your CMS and cannot be transferred.

3. Save the license file somewhere you can easily access, from the computer you're using to access the CMS web UI.
4. Open the CMS application in a browser and log in. (If you have not yet changed them, the default user-name/password is *admin*/*smartwall*.
5. Navigate to **System** > **Licensing**.
6. Click **Add**.
7. You can either:
   - Copy and paste the contents of the license file into the text box. You must include the license header and footer: ----BEGIN-CORERO-LICENSE---- and ----END-CORERO-LICENSE----
   - Select **Upload License File** and click **Choose file** and browse for the license file on your computer. Click **Open**.
8. Once you have a license file by one of those methods, click **Save**.

Setting the Inbound Sample Rate between the vNTD and the CMS

The Port-Mirror Sample Rate you will configure on the router (and the Ingress Sample rate you configure when adding a vNTD to the CMS) controls how many samples are sent from the router to the vNTD. In addition to this, you need to control the rate of samples sent from the vNTD to the CMS. The default setting in a CMS (1 in 1999) is not suitable for a TDD deployment and must be adjusted.

1. In a browser, open the CMS Web UI and log in.
2. Use the left-hand menu to navigate to **Network** > **Devices**.
3. Click the **ADVANCED SETTINGS** tab.
4. Edit the **Inbound Sample Rate** for **sFlow** and **aFlow**. This is the rate of samples sent from the Defense devices to the CMS. The following sample rates should be used for your deployment type:
    - SmartWall TDD production system – Change the value to **16**. This samples 1 in every 16 packets.
    - SmartWall TDD lab test system using smaller traffic volumes and attacks (from a traffic generator) – Change the value to **1**. This samples every packet received by the Defense device.
5. If you want to save the new configuration, and push your changes to any affected Defense devices, click 

. Then, on the pop-up dialog, click **Commit** to push the changes (alternatively, you can click **Discard** to discard any uncommitted changes).

> **Note:** The default sample rates given above are correct for the majority of deployments and you should make sure the Port-Mirror Sample Rate on your router is correct before adjusting the Inbound Sample Rate.

Adding a vNTD to the CMS

You must add a SmartWall Defense device to the CMS before you can manage the attack mitigation Policy on that device. The Defense devices are managed in a Cluster. You must edit this Cluster to expect devices which are using sampled traffic.

Prerequisites

- Deploy and configure (using the pCLI) a CMS
- Deploy and configure (using the pCLI) a vNTD
- (Optional) Contact Corero for a vNTD license and upload the license to the CMS. If you haven't uploaded the license before you add the vNTDs, you can upload the license and apply it to the vNTDs after deployment.

To edit the default Cluster to expect sampled traffic

> **Note:** The Port-Mirror Sample Rate (configured on the router) and Ingress Sample rate should be identical. Both must be the rate factor which reduces the amount of traffic seen by the router to a manageable size for the vNTD. A default value of **1000** would normally scale to approx 1Tbit/sec.

Values less than 1000 will give better fidelity on attack detection and traffic visualization but will add load to the vNTD. A single vNTD has a peak capacity of 10Gbit/sec of sampled traffic when optimized. (Note: the sample rate assumes a run-length of 0)

1. Open the CMS application in a browser and log in.
2. Navigate to **Network** > **Clusters**.
3. From the table, locate the Cluster you want to add the device to, and click the edit button. You can type a text string into the Search field to narrow down the list.
4. In the **Ingress Sample Rate** field, type the sample rate: **1000**
5. Click **Save**.
6. Click Commit. Then, on the pop-up dialog, click **Commit** to push the changes.

To add a vNTD to the CMS

1. Open the CMS application in a browser and log in.
2. Navigate to **Network** > **Devices**.
3. At the Devices table, click **Add**.
4. Type a **Name** for the device.
5. (Optional) Type a **Description** of this device.
6. Type the device's IP **Address** (you will have configured this in the vNTD pCLI).
7. Select the **Cluster** you want to add this device to. The CMS has a **default** Cluster you can add all your devices to. This Cluster uses the policy stored in the default Protection Profile.
8. Select the **Authentication Group** which matches the authentication credentials on this device. The CMS has a **default** Authentication Group which uses the admin/smartwall credentials. If you have changed the device credentials during pCLI set up, you may need to create/edit an Authentication Group.
9. Click **Save**.
10. Repeat this method to add all the devices you require.
11. Click Commit. Then, on the pop-up dialog, click **Commit** to push the changes.
12. If you have already uploaded your vNTD license, your devices will have been automatically licensed when they deployed. If you haven't uploaded the vNTD license, you should do so now following the method above and then manually apply the license to each vNTD:
    a. In the CMS, use the left-hand menu to navigate to **Network** > **Devices**.
    b. On the Devices table, locate the vNTD you want to license.
    c. In the Actions column, click ••• and select **License**.

Note: Adding a device to the CMS does not cause any existing Policy to be pushed to it. You must first add the device to a Cluster. To learn more about creating Clusters and managing Policy, see the SmartWall Central Management Server User Guide.

Configuring the vNTD Segment for TDD

After you've connected your vNTD to the vCMS, you can view information on the available interfaces on the device. By default the vNTD comes with 1 Segment, containing 2 enabled interfaces, but for a TDD deployment, only the 1st interface is required. Follow these steps to disable Link State Propagation (LSP) and to disable the additional interface for the vNTD segment.

1. Use the left-hand menu to navigate to **Network** > **Segments**.
2. From the Segments table, locate the Segment you want to edit and click  the edit button. You can type a text string into the Search field to narrow down the list.
3. (Optional) Edit the **Name**. This must be unique among Segments. You must only use alphanumerics, spaces, or .-&()_/@:= symbols.
4. (Optional) Type a **Description** of up to 265 characters.
5. The **External** Interface on the vNTD will already be selected.
6. In the **Internal** Interface drop-down, select **none-detector**.
7. (Optional) For networks using a tunnel to send traffic samples to the vNTD:
    a. Set the External **IPv4 Address** to the IP address of the external interface on the vNTD (the tunnel end-point)
    b. Set the External **Peer IPv4 Address** to the IP address of the interface which is the last hop before the traffic arrives at the vNTD (e.g the interface on the router which has received the sampled traffic and is directly connected to the vNTD)
8. Under **Link State Propagation**, at the **Admin State** drop-down, select **disabled**.
9. Click **Save**.
10. If you want to save the new configuration, and push your changes to any affected Defense devices, click  . Then, on the pop-up dialog, click **Commit** to push the changes (alternatively, you can click **Discard** to discard any uncommitted changes).

Connecting CMS with SWA

By default, SWA listens for syslog messages on port 9997. You will need to configure CMS to send syslog messages to SWA on this port.

1. Open the CMS application in a browser and log in. (If you have not yet changed them, the default user-name/password is *admin*/*smartwall*.
2. Use the left-hand menu to navigate to **System** > **Analytics & Syslog**. Make sure the SERVERS tab is selected.
3. At the **Analytics Servers** table, click **Add Server**.
4. Type a **Name** for this server. You must only use alphanumerics, spaces, or .-&()_/@:= symbols.
5. Type the IP **Address** of the server (or its DNS name).
6. Enable or Disable **Encryption** for this server. The CMS and SWA come with self-signed SSL certificates. You can choose to upload signed certificates to the CMS and SWA- see optional steps below.
7. Leave the default **Port**: **9997** for unencrypted or **9998** for encrypted connections.
8. Click **Save**.
9. Click 

. Then, on the pop-up dialog, click **Commit** to push the changes.

(Optional) Add a signed certificate to the CMS - SWA connection

If you enable encryption, the connection between the CMS and SWA uses, by default, an in-built self-signed certificate. If you want to use a signed certificate, you need to upload a PKC#S12 certificate to both sides of the connection.

1. Add a signed SSL certificate in the CMS side of the connection:
    a. Open the CMS application in a browser and log in. (If you have not yet changed them, the default user-name/password is *admin*/*smartwall*.
    b. Use the left-hand menu to navigate to **System** > **Analytics & Syslog**.
    c. Open the **SSL CERTIFICATE** tab.
    d. Click **Upload Certificate**.
    e. Select a pkcs12 certificate file on your computer, and click **Open**.
    f. (Optional) Type in the **Password** for the certificate file.
    g. Click **OK**.
    h. If necessary, refresh the browser to ensure the new certificate has been loaded.
    i. Click 

. Then, on the pop-up dialog, click **Commit** to push the changes.

2. Add a signed certificate to the part of the SWA that receives information from CMS:
    a. Access the SWA pCLI:
        - Open a console session. On an ESXi server, you can use VMware (select the VM and click **Open Console**) or on a KVM server you can use virsh (command: `virsh console <vmName>`).
        - SSH to the pCLI: `ssh -p 2222 admin@<ipAddress>`
    b. Log in. If you haven't yet changed them, the default username and password is admin/smartwall.
    c. To load a certificate, type `ssl-certificates forwarder` followed by the URI to the PKCS#12 format certificate file. The supported protocols are FTP, SFTP, HTTP, and HTTPS. For example: `ssl-certificates forwarder sftp://admin@10.20.30.40/certs/my_cert.p12`
    d. You will be prompted for a password to access the file location. If you password protected the PKCS#12 file, you will also be prompted for that password.

To add CMS credentials to the SWA

You must add a set of CMS admin credentials to enable the SWA to communicate mitigation changes back to the CMS.

1. Open the SWA application in a browser and log in.
2. Use the top menu to navigate to **Mitigation** > **Remote Devices**.
3. At the table, click **Add Device**.
4. Type a **Name** for your CMS. You must only use alphanumerics, spaces, or .-&()_/@:= symbols.
5. Type the IP **Address** (IPv4) of your CMS.
6. In the **Type** field, type: **CMS**. **Caution:** You must use uppercase for all letters or it will not be recognized.
7. (Optional) Type a **Description** of your CMS.
8. Enter a **Username** for an admin account on your CMS.
9. Enter a **Password** for an admin account on your CMS.
10. Click **Save**.

(Optional) Uploading a custom SSL certificate to the CMS

The CMS comes with a default self-signed Corero SSL certificate which your browser will list as "not secure". As soon as possible, you should replace this with a signed certificate. This must be packaged in pkcs12 format and can optionally be password protected.

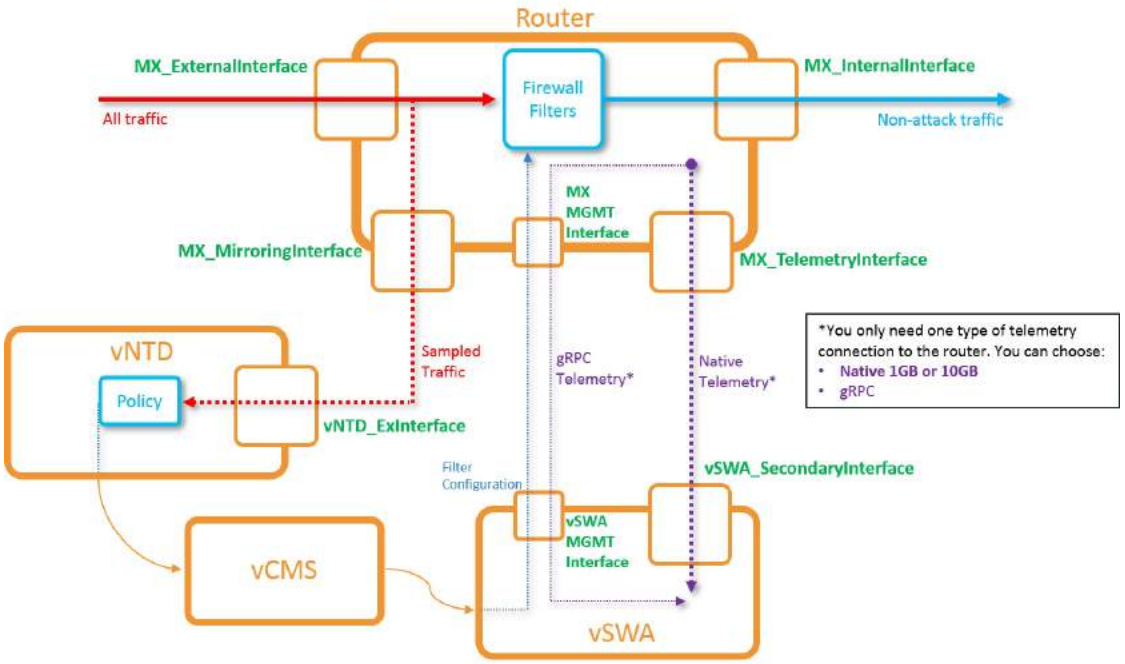
1. Open the CMS application in a browser and log in. (If you have not yet changed them, the default user-name/password is *admin*/*smartwall*.
2. Navigate to **System** > **HTTPS**.
3. Click **Upload Certificate**.
4. Select the pkcs12 certificate file on your computer, and click **Open**.
5. (Optional) Type in the **Password** for the certificate file.
6. Click **OK**.
7. Refresh the browser to view the new security rating.

> **Note:** The certificate must be in PKCS#12 format, and include the private key, signed certificate, and CA certificate change to be used for SSL. The common name should match the hostname assigned to the SWA appliance.

Configuring the Juniper Networks MX Series router

To enable a router to accept DDoS mitigation filters from the SmartWall TDD, you must configure the following settings. The configuration provided assumes you have a new system, so you may need to leave out some of the earlier commands if you're using an existing router.

> **Caution:** When you configure a Juniper Networks MX Series router to send sampled traffic to a vNTD, you MUST set up `port-mirroring`. DO NOT set up `sampling`. The router's port-mirroring function sends 1 out of every n packets to the vNTD without alterations, but the router's sampling function sends a summary of the traffic in that period. The vNTD cannot process a summarized sample.



Prerequisites

The configuration provided below assumes you have a new system with initial configuration already setup for your network. At a minimum, the system must be licensed, have SSH enabled, be connected to your management network and have a host name (`set system host-name <name>`).

> **Note:** The hostname must be enabled on the forwarding plane of the Juniper Networks MX Series router.

For gRPC telemetry only: You must download two additional software files. Contact your support representative for assistance.

- network-agent-x86-32-17.4R1.16-C1.tgz
- junos-openconfig-x86-32-0.0.0.9.tgz

Before you begin, you also need to know the following information which you will use to replace the placeholders in the commands below:

- *<MX_ExternalInterface_Name>* – The names of the external interfaces on your router which you want to protect with the TDD system (e.g. `xe-0/0/0`).
- *<MX_MirroringInterface_Name>* – The name of the interface on your router which you have allocated for sending mirrored traffic to the vNTD (e.g. `xe-0/0/2`).
- *<MX_MirroringInterface_IPv4_Subnet>* – The IPv4 address of the interface on your router which you have allocated for mirroring, formatted as a CIDR (e.g. `192.168.66.201/24)`.
- *<MX_MirroringInterface_IPv6_Subnet>* – The IPv6 address of the interface on your router which you have allocated for mirroring, formatted as a CIDR (e.g. `2000:2000:cccc:cccc::1/64`).
- *<MX_TelemetryInterface_Name>* – **(Native telemetry only)** The name of the interface on your router which you have allocated for sending telemetry to the vSWA (e.g. `xe-0/0/3`).
- *<MX_TelemetryInterface_IP>* – **(Native telemetry only)** The IPv4 address of the interface on your router which you have allocated for telemetry (e.g. `192.168.99.201`).
- *<vNTD_ExInterface_MAC>* – The MAC address of the external interface on your vNTD (e.g. `00:0c:29:36:94:5a`). To find this, log into the vNTD pCLI and type the command `show nic` and look for the MAC address under `External`.
- *<vNTD_ExInterface_IPv4>* – The IPv4 address you want to allocate to the external interface on the vNTD (e.g. `192.168.66.105`). This must be in the same subnet as the interface on the router which you allocated for mirroring (MX_MirroringInterface). Note: The vNTD is not a layer 3 device and does not have a configurable IP address on its external interface, you will add this IP address using a static ARP associated with its MAC address.
- *<vNTD_ExInterface_IPv6>* – The IPv6 address you want to allocate to the external interface on the vNTD (e.g. `2000:2000:cccc:cccc::2`). This must be in the same subnet as the interface on the router which you allocated for mirroring (MX_MirroringInterface). Note: The vNTD is not a layer 3 device and does not have a configurable IP address on its external interface, you will add this IP address using a static ARP associated with its MAC address.
- *<vSWA_SecondaryInterface_IP>* – (Native telemetry only) The IP address of the secondary interface on the vSWA (e.g. `192.168.99.113`). You can configure this IP address by logging into the vSWA pCLI and using the setup network wizard. The IP must be on the same network as the interface on the router which you allocated for telemetry (MX_TelemetryInterface).

To configure a Juniper Networks MX Series router for use with the SmartWall TDD system

> **Caution:** If copying and pasting from the PDF, you can experience some loss of characters. If possible, use the online help version of the documentation. Alternately, first copy this set of commands into a plain text word processor (e.g. notepad) and check none of the hyphens or spaces have been removed and that no additional returns have been added.

1. Open the Juniper Networks MX Series router CLI using an SSH client: `ssh <username>@<ipaddress>`
2. Enter your password to log in.
3. Enter the following commands. Each line is an individual command- paste a single line and press **Enter** before moving on to the next line.
4. **gRPC telemetry only:** Enter the following commands to enable the router to send telemetry over the management interface:
   ```
   request system software add network-agent-x86-32-17.4R1.16-C1.tgz
   request system software add junos-openconfig-x86-32-0.0.0.9.tgz
   ```
5. Enter configuration mode: `configure`
6. Unless already enabled, you must enable the router to accept NetConf . This enables the TDD to push configuration to the router:
   ```
   set system services netconf ssh port 830
   ```

   > **Note:** If you want to use a custom NetConf port, you can replace 830 with your required port number. You must also add the custom port number to the Remote Devices table entry for this router.

7. Create an ephemeral instance of the configuration database named Corero and limit it to only store the last 500 commits in memory. This is where configuration from the TDD is sent.
   ```
   set system configuration-database ephemeral instance Corero
   set system configuration-database ephemeral purge-on-version 500
   ```
8. Identify an interface on the router which can be used for sending mirrored traffic to the vNTD. Set and label this interface (using the description field):
   ```
   set interfaces <MX_MirroringInterface_Name> description Sample_Port_to_vNTD
   ```
9. **Native telemetry only:** Identify another interface on the router which can be used for sending telemetry to the vSWA. Set and label this interface (using the description field):
   ```
   set interfaces <MX_TelemetryInterface_Name> description Interface_for_telemetry
   ```

10. Configure Port-Mirroring to forward a sample rate of traffic from the router to the vNTD. As the vNTD is not a layer 3 device, you cannot assign an IP address to it's external interface. Instead of using an IP address, you forward traffic from the router using the vNTD's MAC address. However, because the router requires an IP address, you must also assign a dummy IP address to the vNTD's external interface (in the same subnet as the allocated forwarding interface on the router). This IP address is not used for routing, which is handled by setting up a static arp.

```
set forwarding-options port-mirroring input rate 1000
set forwarding-options port-mirroring input run-length 0
set forwarding-options port-mirroring family inet output interface <MX_Mir-
roringInterface_Name> next-hop <vNTD_ExInterface_IPv4>
set forwarding-options port-mirroring family inet6 output interface <MX_Mir-
roringInterface_Name> next-hop <vNTD_ExInterface_IPv6>
set interfaces <MX_MirroringInterface_Name> unit 0 family inet address <MX_Mir-
roringInterface_IPv4_Subnet> arp <vNTD_ExInterface_IPv4> mac <vNTD_ExInterface_
MAC>
set interfaces <MX_MirroringInterface_Name> unit 0 family inet6 address <MX_Mir-
roringInterface_IPv6_Subnet> ndp <vNTD_ExInterface_IPv6> mac <vNTD_ExInterface_
MAC>
```

> **Note:** The `1000` value you enter, in the top command here, is the rate factor which reduces the amount of traffic seen by the router to a manageable size for the vNTD (10Gbps or less). 1 in every 1000 packets is sampled and sent to the vNTD. This enables you to sample from up to 1Tbit/sec of traffic for each vNTD. (Note: the sample rate assumes a run-length of 0).

11. Add new filters to the firewall (IPv4 and IPv6) . The default terms configured here count the traffic for telemetry, use the port-mirroring configuration to mirror the traffic samples to the vNTD, and accept the actual traffic. When the TDD sends configuration to the router to block attack traffic, that configuration is added as an ephemeral term.

```
set firewall family inet filter CORERO-MITIGATE term default-term then count
Corero-Allowed
set firewall family inet filter CORERO-MITIGATE term default-term then port-mirror
set firewall family inet filter CORERO-MITIGATE term default-term then accept
set firewall family inet6 filter CORERO-MITIGATE6 term default-term then count
Corero-Allowed6
set firewall family inet6 filter CORERO-MITIGATE6 term default-term then port-mir-
ror
set firewall family inet6 filter CORERO-MITIGATE6 term default-term then accept
```

12. For every external interface whose traffic you want to protect with the TDD system, you must add the CORERO filters (IPv4 and IPv6).

```
set interfaces <MX_ExternalInterface_Name> unit 0 family inet filter input CORERO-MITIGATE
set interfaces <MX_ExternalInterface_Name> unit 0 family inet6 filter input CORERO-MITIGATE6
```

13. Configure the type of telemetry sent to the SWA:
    - For Native telemetry use the following commands:

```
set services analytics streaming-server Corero remote-address <vSWA_SecondaryInterface_IP>
set services analytics streaming-server Corero remote-port 30000
```

    - For unencrypted gRPC telemetry use the following command. Note: `clear-text` is a hidden option and must be typed in full:

```
set system services extension-service request-response grpc clear-text port 32767
```

    - For SSL encrypted gRPC telemetry, you must already have an SSL certificate on this router. If you already have a certificate on the router, use the following command:

```
set system services extension-service request-response grpc ssl local-certificate <certificateName> port 32767
```

Tip: If you need to add a new certificate to this router, you can use the following command:

```
set security certificates local <certificateName> load-key-file <URL>
```

14. **Native telemetry only:** Configure the telemetry sent to the SWA.

```
set services analytics export-profile Corero-FF-mitigate local-address <MX_TelemetryInterface_IP>
set services analytics export-profile Corero-FF-mitigate local-port 22222
set services analytics export-profile Corero-FF-mitigate reporting-rate 1
set services analytics export-profile Corero-FF-mitigate payload-size 9000
set services analytics export-profile Corero-FF-mitigate format gpb
set services analytics export-profile Corero-FF-mitigate transport udp
set services analytics sensor Corero-FF-mitigate server-name Corero
set services analytics sensor Corero-FF-mitigate export-name Corero-FF-mitigate
set services analytics sensor Corero-FF-mitigate resource /junos/system/linecard/firewall/
set services analytics sensor Corero-FF-mitigate resource-filter CORERO.*
```

Note: If your router is using Input Lists, you should replace the last command with the following:

```
set services analytics sensor Corero-FF-mitigate resource-filter "^[^_].*"
```

15. **(Optional)** Configure role-based access to routers by creating a new login class called TDD and a user within that class called corero.

```
set system login class TDD permissions configure
set system login class TDD permissions view
set system login class TDD permissions firewall-control
set system login user corero class TDD
set system login user corero authentication plain-text-password
```

> **Note:** When you add your routers to the Remote Devices table in SWA, you must enter the username `corero` and the same password you configured here.

16. Once you have performed all the commands, you must commit the changes, enter the following command: `commit`
17. Exit configuration mode: `exit`

Adding a Juniper Networks MX Series router as a Remote Device

To enable the SmartWall TDD system to send instructions to a Juniper Networks MX Series router, that router must be added to the SWA as a Remote Mitigation Device. Once you add all the Remote Devices to the SWA, you must complete the configuration by editing the Juniper Alert Action to send the mitigations to those devices.

To add a router to the SWA

1. Open the SWA application in a browser and log in.
2. Use the top menu to navigate to **Mitigation** > **Remote Devices**.
3. At the table, click **Add Device**.
4. In the **Name** field type the host name for this router. The host name is configured on your router and must match what you enter here exactly. You must only use alphanumerics, spaces, or .-&()_/@:= symbols.
5. In the **Address** field, type the IPv4 address of the router or the router hostname (must be DNS resolvable).
6. (Optional) If you're using the standard NetConf port 830, leave this field blank. Otherwise, you can specify the custom NetConf port you want to use to communicate with the router. **Caution:** You must also use the custom port when configuring the router.
7. In the **Type** field, type: **MX**. **Caution:** You must use uppercase for both letters or it will not be recognized.
8. (Optional) Type a **Description** of the router.
9. Enter a **Username** for a user account with permission to configure the router.

   > **Note:** If you configured role-based access when you set up the router, you must enter the username `corero` and the same password you configured there.

10. Enter an authentication method for this device. Either:
    - Enter a **Password** for the user credentials to allow the SmartWall TDD system access to edit configuration on the router.
    - **(Native Telemetry only)** Paste in an **SSH Key** and **SSH Key Passphrase**. The SSH Key must be valid ASCII data for the private key, in PEM format (text starting with '-----BEGIN DSA PRIVATE KEY-----' or '-----BEGIN RSA PRIVATE KEY-----').

11. **(gRPC Telemetry only)** Set up gRPC Telemetry (if you don't want to use gRPC, leave the telemetry drop-down as **Native**).
    a. From the Telemetry drop-down, select **gRPC**.
    b. If required, edit the **gRPC Port**. The default port is 32767.
    c. (Optional) If you're using SSL encryption on your gRPC telemetry connection:
        i. Upload a **gRPC SSL Certificate Authority** using the **Choose File** button to select a certificate from your computer.
        ii. In the **gRPC SSL Expected Server** field, type the CN name from the router's certificate. The CN name must be formatted as a DNS name. **Tip:** If the CN name is the same as the router hostname you entered in the Address field, you can leave this field blank.
12. Click **Save**.

> **Tip:** On the table, you can use the following action options to **Edit** or **Delete** a remote device.

To configure the Juniper Alert to send mitigations to those devices

1. Open the SWA application in a browser and log in.
2. Use the top menu to navigate to **Alerts**.
3. At the table, locate **Real-Time Juniper 3**.
4. In the Action column click **Edit**.
5. Select **Enable**.
6. In the Action column click **Edit** again, then select **Edit Alert**.
7. Under Trigger Actions, expand **Corero Autonomic Response**.
8. In the **Devices Name** field you must type the hostnames of all your Remote Devices separated by commas (e.g. router1,router2,router3). This must exactly match the hostname on the router, and the name you provided in the Remote Mitigation table.
9. Click **Save**.

Verifying the TDD System is Connected

After completing all the tasks in this guide. Your TDD system should be in the following state:

Juniper Networks MX Series router:

- Configured to accept instructions from the vSWA
- Configured to send telemetry to the vSWA
- Configured to send port-mirrored traffic samples to the vNTD

vNTD:

- Accessible on your Management Network (pCLI only)
- Connected to the vCMS
- Licensed with your unique vNTD license (uploaded to the vCMS)

vCMS:

- Accessible on your Management Network (pCLI, CLI and Web UI)
- Connected to the vNTD and vSWA
- Operational Mode left in Monitor, ready to switch to Mitigate once you're happy with your defense Policy

vSWA:

- Accessible on your Management Network (pCLI and Web UI)
- Connected to vCMS
- Upload a SecureWatch Package File to manage your TDD license
- Remote Mitigation table contains all routers
- Add the names of the routers to the Corero Autonomics Alert

To verify the TDD system is connected

You can check your system is now fully connected in the SWA web UI.

1. Open the SWA web UI in a browser.
2. Open the **System** > **Health** screen.
3. You should now see green checks against every item, except:
    - Remote Device Info – This should be amber (warning) until a filter has been active on the system.
    - Slack Notifications – Will show blue (information). This is normal.

To force a Remote Device Info system check

On the **System** > **Health** screen, the **Remote Device Info** table won't show green ticks against a router until a filter has been sent and successfully received by the router. If you want to verify the connection before sending an active filter,

you can use the method below to send a dummy filter to your connected routers:

1. Open the SWA web UI in a browser.
2. Navigate to the **Dashboards** screen.
3. Click on **Flexible Configuration Tool** to open the dashboard.
4. From the **Action** drop-down, select **Detect**.
5. In the **Destination Host or CIDR** field, type **1.1.1.1/32** and press enter.
6. From the **Protocol** drop-down, select **IP** or **UDP**.
7. At the bottom right of the green area, click **Add**.
8. Click **Add** to send the filter to your routers.
9. Navigate to **System** > **Health** screen. The **Remote Device Info** table should now show green ticks against the routers.

Configuring the TDD Policy for your Network

You should now have a fully connected SmartWall TDD system communicating with your Remote Devices. By default, the SmartWall TDD system is in Monitor mode. In this mode, when it identifies DDoS attack traffic, it sends filters to the routers that will detect DDoS attacks but not block any traffic. You can see in the SWA analytics application which traffic the system would be blocking if it was in Mitigate mode.

You should leave the system in this mode for a few days, until you have analytics on enough traffic to determine what is normal for you. You can then adjust the Policy settings for your network.

Only once you have evaluated the defense Policy for your network traffic, should you then switch the system into **Mitigate** Mode and begin blocking DDoS attack traffic.

Troubleshooting

This section describes methods for addressing some problems that can occur when installing the system and making it operational.

Cannot access the Web UI (CMS or SWA)

Access to the CMS web UI and the SWA web UI is provided via the management IP address configured for each at setup time; make sure to use HTTPS for both, and specify port 8000 when accessing SWA:

- **CMS**: https://x.x.x.x [management IP address]
- **SWA**: https://x.x.x.x:8000 [management IP address]

To log in to CMS and SWA, make sure to use the administrator credentials that were specified at setup time. The default username is *admin* and the default password is *smartwall*. If a different username/password combination was specified during setup, you need to use those credentials instead.

Getting help for using the CMS or SWA

CMS and SWA each provides a link to help documentation in the top menu bar.

In the CMS, click the help icon to access the CMS Knowledgebase. From the home page of the Knowledgebase you can download additional help PDFs, browse for information using the expandable left-hand menu, or type a search term in the search bar.

In SWA, clicking **Help** > **User Guide** displays a PDF file that describes the controls shown in each SWA screen.

CMS configuration change does not take effect

Configuration changes in the CMS do not take effect until they are committed and any uncommitted changes can be lost when you logout. Always remember to commit your changes (**Commit** > **Commit**), before you log out.

Defense device not reachable from CMS

Adding a Defense device to the CMS doesn't automatically mean the device is reachable from the CMS, you are just telling the CMS what device to look for. A variety of different problems could prevent the CMS from communicating with the device.

In the CMS, click **Network** > **Devices** to display the Devices table. The **Deployment State** column shows connectivity information for each managed device. The following states indicate the device is not connected:

- **Connection refused** – The CMS successfully sent a request to the device but the device refused to send a response.
- **Connection timed out** – The CMS attempted a connection but the attempt timed out.
- **Authentication failed** – The CMS attempted a connection but the authentication credentials on the CMS did not match the credentials on the device.

Most issues can be remedied by performing the following checks:

- Does the device have power?
- Is the device management port connected to the network?
- Does the CMS have connectivity to the network on which the devices management interface is connected?
- Does your firewall allow the connection?
- Does the CMS have the correct IP address for that device?
- Is the device in the correct Authentication Group in the CMS? Do those credentials need to be updated?

The Defense device shows out-of-sync in the CMS

If a device is "out of sync" it means that the Policy on the device does not match the corresponding Policy in the CMS. You may occasionally see a device become out of sync after the device has been restarted, or after you perform a software upgrade. In the CMS, click **Network** > **Devices** display the Devices table and look in the **Deployment State** column to see which device is out of sync, and what action is required:

- **Sync required** – The device is connected but its Policy configuration does not match the current Policy committed to the CMS. The device could have become out of sync if it was unavailable when a change was committed in the CMS or if you have replaced a connected device with a new version (with the same IP address). In the Devices table, click ••• the action button and select **Sync Device** to push the Policy changes to the device.
- **Force sync required** – The device is connected but there has been an unexpected error in the Policy configuration. In the Devices table, click ••• the action button and select **Force Sync Device** to wipe the old Policy from the device and replace it with the current version stored in the CMS.
- **Not in cluster** – The device is not in a Cluster. Go to **Network**> **Clusters**, add the device to an existing Cluster or create a new Cluster for it.
- **Initial sync pending** – The device is new and the CMS has not yet sent its Policy configuration. Wait a few minutes and check again.

vNTD device showing as not-licensed

You must have at least 10Gbps available license capacity for the Defense device to automatically license and connect to the CMS. If you don't, you will have to create some space by delicensing an old vNTD or buying additional license capacity from your Corero representative. You can then license the device manually:

1. In the CMS, use the left-hand menu to navigate to **Network** > **Devices**.
2. On the Devices table, locate the vNTD you want to license.
3. In the Actions column, click ••• and select **License**.

> **Tip:** If you need to delicense a vNTD, in the Actions column, click ••• and select **Delicense**. The delicense option is only available for currently licensed vNTDs. When you delicense a vNTD (or add it to the CMS when there isn't enough license capacity available), it enters the not-licensed state. In the not-licensed state, the devices do not send any information via syslog message (except the device status), and cannot function as Detection Engines for the TDD.

Remote Device added to the CMS Devices table instead of the SWA

If you accidentally add a Remote Device (e.g. a router) to the Devices table in the CMS (**Network** > **Devices**), it will appear with a status message of `unexpected device type`. Remote Devices cannot be stored in this table, it is only for vNTDs. Delete the Remote Device from the table and instead add it to the Remote Devices screen in the SWA Web UI (**Mitigation** >**Remote Devices**).

Cannot add a new vNTD to a CMS Cluster

To use SmartWall Network Threat Defense virtual editions (vNTDs) you need to have a license for them. If you do not have a vNTD license, or you have already allocated your full license capacity, when you add a new vNTD to the CMS you will be unable to add it to a Cluster. In the CMS, click **Network** > **Devices** to display the Devices table and check the **Deployment State** column to see if the vNTD is listed as **not-licensed**. To license this vNTD you need to contact your Corero representative for additional license capacity and upload the new license file. Alternately, you can choose to delicense another vNTD. Once you have the available license capacity, at the Devices table click ••• the action button next to the unlicensed vNTD, and select **License**.

SWA doesn't show any data from the CMS

SWA receives data from the managed Defense and Bypass devices via the CMS. If SWA is not receiving data, it will not show any information on the Overview scren. Open the CMS in a browser and navigate to **System** > **Analytics & Syslog**. On the Servers table, check the details of your SWA application to make sure you have the correct IP address and the right port number (the default should be 9997). If you need to make a change click the edit button and remember to commit your changes (**Commit** > **Commit**).

If the CMS is configured correctly to send data to SWA, but you still don't see all the expected data in SWA, it may be that your devices are not reachable from the CMS. In the CMS, click **Network** > **Devices** to display the Devices table and check the **Deployment State** column to see if the devices are connected. If they aren't connected, follow the checks in the above method: Defense device not reachable from CMS.

Remote Device Info table (System > Health) is showing warning against new router

On the **System** > **Health** screen, the **Remote Device Info** table won't show green ticks against a router until a filter has been sent and successfully received by the router. If you want to verify the connection before sending an active filter, you can use the method below to send a dummy filter to your connected routers:

1. Open the SWA web UI in a browser.
2. Navigate to the **Dashboards** screen.
3. Click on **Flexible Configuration Tool** to open the dashboard.
4. From the **Action** drop-down, select **Detect**.
5. In the **Destination Host or CIDR** field, type **1.1.1.1/32** and press enter.
6. From the **Protocol** drop-down, select **IP** or **UDP**.
7. At the bottom right of the green area, click **Add**.
8. Click **Add** to send the filter to your routers.
9. Navigate to **System** > **Health** screen. The **Remote Device Info** table should now show green ticks against the routers.

SWA doesn't show any telemetry data from a router

If you are expecting to see telemetry from a router but none is appearing in the SWA, you must check the router has been successfully added to the SWA and CMS application.

First, check the SWA. On the Health screen (**System**>**Health**), if you cannot see the router in the **Remote Devices Info** table, there has most likely been a mistake made when adding the routers.

Check the Autonomics alert has the correct hostnames for the routers: **Alerts** > **Real-Time Juniper 3** >**Edit** > **Edit Alert** > **Corero Autonomic Response**. Then under Device Names, make sure the router hostnames are spelled identically to how they appear in the router and in the SWA Remote Devices table. Also check that they are all separated by a single comma (no spaces).

Check the SWA to make sure the correct IP addresses and access credentials are stored for each router. Open the SWA Web UI > **Mitigation** > **Remote Devices** and click **Edit** next to each router to check the stored information. The name shown must be the hostname for the device, check it is identical to how it is displayed on the router. The password is obfuscated so you may need to re-enter it.

Finally, check the configuration on the forwarding plane of the Juniper Networks MX Series router. A good place to start is to check the router is reachable and recieving traffic as expected. The following three troubleshooting commands can help you begin to diagnose an issue, for more information see the Juniper documentation for your router.

- `show chassis fpc` – Display status information
- `ping 128.0.0.16 routing-instance __juniper_private1__` – Check the connection between the forwarding plane and control plane
- `monitor interface traffic` – Display real-time statistics about interfaces

> **Caution:** Be careful to get the order of words correct in `monitor interface traffic`. Typing `monitor traffic interface` may start a TCP dump.

Telemetry traffic is only showing for one of my connected routers

If you have some of your routers in a remote subnet from the SWA, the telemetry traffic may not be recognized on the secondary interface. To fix this, you need to add a static route from the SWA to each router on the remote subnet. You can do this in the SWA pCLI, for each router:

1. Open the SWA pCLI and log in as the admin user.
2. Type the command: `setup routes`
3. Type `I` to insert a new route.
4. Enter the following information:
    - Destination IPv4 Address – The IP address of the telemetry interface on the router.
    - Network Mask – 255.255.255.255
    - Gateway – The IP of the next hop router from your SWA
5. Type `A` to accept the change.

Traffic is entering the network, but the Defense device does not seem to do anything with it

If the tables and charts in SWA show inbound traffic entering the network, but there's no evidence that rules are being triggered on the SWA, there may be no DDoS attacks occurring. However, if the traffic appears abnormal but the system is not responding to it as expected, you should first check the system is healthy:

1. Open the SWA Web UI and log in.
2. Navigate to **System** > **Health**.
3. Check all table rows are showing as expected with **green ticks** to indicate good health. If there are any errors, warning, or information messages, you can investigate to see if these are the route of the issue.

If your system is in good health, another possibility is that you may have accidentally changed the necessary defense policy settings required to trigger mitigations. Check the CMS defense policy defaults are still in place. You can contact your support representative for more information.

> **Note:** If the TDD Flex-Rules show a revision number higher than 1, you may have accidentally edited the filter definition . This can disrupt the TDD system's ability to mitigate attack traffic. Contact your support representative for a copy of the original filter definition if you're concerned.

Mitigations are not performing the actions I expect

For the TDD mitigations to work as described, your CMS Operating Mode must be set to Mitigate. If the Operating Mode is in Monitor you will see the following behavior:

- Block mitigations > accept action on router
- Detect mitigations > accept action on router
- Redirect mitigations > act as disabled mitigations
- Policer mitigations > act as disabled mitigations
- Ignore and disabled work as expected

To change the Operating Mode to Mitigate

1. Open the CMS web UI in a browser and log in.
2. Use the left-hand menu to navigate to **Network** > **Operating Modes**.
3. Use the **Global Defense Mode** drop-down to change the default mode to **mitigate**.
4. If you want to save the new configuration, and push your changes to any affected Defense devices, click 

. Then, on the pop-up dialog, click **Commit** to push the changes (alternatively, you can click **Discard** to discard any uncommitted changes).

> **Note:** The Monitor mode can be used for testing new mitigations and is the default mode for new installations.

CMS shows uncleared alarms

In the CMS application, click on  the Alarm icon in the Status bar and open the Alarm Center. Uncleared alarms are listed, describing issues that require your attention.

Lost administrative user credentials

If you have lost the admin user credentials for a vNTD, vCMS or vSWA virtual machine, you can reset the username/password to their default values (admin/smartwall) without redeploying the VM.

> **Note:** Resetting the administrative username and password will not affect any other user credentials.

1. Create a password reset ISO file. You can use the following command to create a password reset ISO file in Linux. These commands assume you already have the package containing the mkisofs command installed; if you don't, you should download this first using the Linux package manager.
   ```
   >reset_pw
    mkisofs -input-charset utf-8 -quiet -o reset.iso reset_pw
   ```
2. Transfer the ISO file to a datastore which can be accessed by the virtual machine.
3. On the host where the virtual machine is deployed, set the CD-ROM drive for the virtual machine to the datastore ISO file . Ensure that the device status is connected.
4. Restart the guest virtual machine. When the virtual machine reboots, it should display the following message: `Username/password reset to default`
5. Login with the username: `admin` and the password: `smartwall`. After logging in, you can change the username and/or password using the `setup aaa` command in the pCLI.
6. Disconnect the CD-ROM from the virtual machine (e.g. in VMware you can do this by clicking **Edit virtual machine settings**, opening the **CD/DVD drive 1** settings, and deselecting **Connected**). If you don't, the username and password will be reset to default every time you restart the VM.

Downloading diagnostic packages

During troubleshooting, customer support may ask you for diagnostic packages from the affected systems. You can download then through your browser in the following locations:

- **For SWA** – Open the SWA in a browser. **System** > **Settings** > **Diagnostic Package** > **Download**
- **For CMS** – Open the CMS in a browser. **System** > **Diagnostics**. Under **Download file from CMS appliance**, select a **source** package type and click **Download File**.
- **For a vNTD** – Open the CMS in a browser. **System** > **Diagnostics**. Under **Download file from a device**, select a source **package** type and the specific **device**. Click **Download File**.

After restarting my server, the Corero applications haven't come back up

By default, Corero VMs are not configured to automatically start on ESXi server boot. Corero recommends you set the VMs to automatically start by editing the virsh configuration.

To configure the host to auto-start VMs after a restart

1. Log into the server hosting your Corero VMs.
2. Use the following command for each VM, replacing `<vmName>` with the name of the VM: `virsh autostart <vmName>`

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or Partner Support Service support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the JTAC User Guide located at https://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf.
- Product warranties—For product warranty information, visit https://www.juniper.net/support/warranty/.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

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

- Find CSC offerings: https://www.juniper.net/customers/support/
- Search for known bugs: https://prsearch.juniper.net/
- Find product documentation: https://www.juniper.net/documentation/
- Find solutions and answer questions using our Knowledge Base: https://kb.juniper.net/
- Download the latest versions of software and review release notes: https://corero.force.com/support
- Search technical bulletins for relevant hardware and software notifications: https://kb.juniper.net/InfoCenter/
- Join and participate in the Juniper Networks Community Forum: https://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: https://www.juniper.net/cm/

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: https://entitlementsearch.juniper.net/entitlementsearch/

Creating a Service Request with JTAC

You can create a service request with JTAC on the Web or by telephone.

- Visit https://myjuniper.juniper.net.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see https://support.juniper.net/support/requesting-support/.

Requesting Licenses

The system requires a TDD license key, plus keys for each vNTD, to become fully operational. Juniper devices do not require license keys to support the solution. To obtain the keys, please contact the Corero Customer Services team by one of the following methods:

- Email: Support.Portal@corero.com
- Web: https://corero.force.com/support
- Telephone: Dial +1.978.212.1500 -> Select Option 2

Appendix A – Deploying a vNTD for High Sampled Traffic Rates

The following instructions cover how to deploy a vNTD for deployments expecting sampled traffic rates of 1.2Gbps or higher.

> **Note:** The CMS and SWA are always deployed in the same way, regardless of the expected traffic rate. Only the vNTD requires the modified method below. For the CMS and SWA use the method in the main document.

For high performance rates the vNTD requires dedicated hardware resources to achieve performance:

- 7 available cores for the vNTD
- CPU Core Pinning
- Huge Pages – Optional, but encouraged to improve performance for any deployments approaching line rate sampled traffic (e.g. 9Gbps mixed packet sizes)
- PCI Passthrough

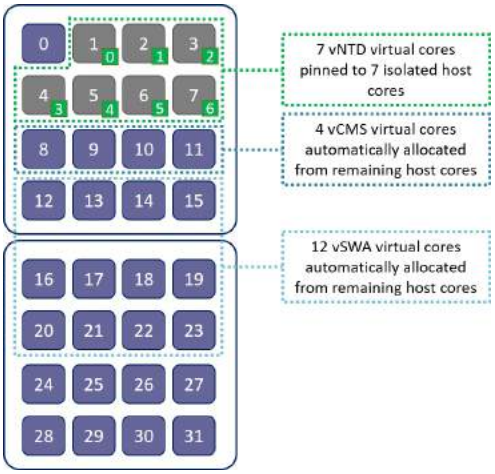
If you do not expect a sampled traffic rate of 1.2Gbps or higher, you should use the standard deployment steps for a vNTD.

> **Caution:** Hugepages improve performance by changing how memory is allocated on the host. On **Debian** hosts, use of Huge Pages can expose a known bug in memory accounting in the Debian LibVirt system that can lead to vNTD failure. When deploying a vNTD on Debian it is recommended you avoid huge pages. If you require huge pages for line rate, there is a safe method available in the deployment instructions below.

Prerequisites

- Prepare the disk image for deployment:
  - Save and export the vNTD zip file (*vntd_9.7.5.nnn-kvm.zip*). This creates a folder containing the required qcow2 disk image.
  - Modify the disk image to use preallocation=falloc (the equivalent of thick provisioning the disk to ensure correct storage allocation) and rename the file for ease of deployment. Use the following command: `qemu-img convert -f qcow2 <disk image file name> -O qcow2 <newFileName>-disk1.qcow2 -o preallocation=falloc`
- Identify your expected sampled traffic rate, and decide if you need hugepages and additional host optimization to reach 10Gbps
- To avoid miss-allocation of cores, CPU Core Pinning must be completed in 3 places and identify the same cores in each place. Decide on the 7 host cores you want to allocate to the vNTD and use the same cores in every time the `<coreRange>` is required:
  - When configuring the Host OS isolated CPUs using grub
  - When you deploy the vNTD VM (in the virt-install command or the virsh XML template)
  - Post VM deployment, by pinning the cores using virsh

Caution: Incorrectly allocating cores can lead to cores being isolated by the Host OS which are then unavailable to any of the VMs including the vNTD.



Note: You can replace the `<coreRange>` variable in the grub command and vNTD deployment methods with a range of 7 cores (e.g. `1-7`) or with 7 individual cores (e.g. `2,4,6,8,10,12,14`). However, you must use the same value for `<coreRange>` in every place it appears in the methods below (e.g. if `isolcpus=1-7` when you configure the host OS, then in the virt-install deployment command `cpuset=1-7`).

Configure the KVM Host

> **Caution:** This process isolates CPU cores for use with the vNTD. If you follow this method and later chose to reconfigure your host to use a virtIO vNTD or for another purpose, you MUST remove the core isolation from your kernel boot parameters or you will be unable to use the isolated cores.

Enable Intel Virtualization features

Make sure your host has the following features enabled:

- VT-d
- VT-x

Isolate CPU cores to enable pinning, and optionally enable Huge Pages

> **Caution:** You will see errors if you complete the following steps but, after deploying the vNTD, you do not complete the passthrough process by pinning the cores (instructions after each deployment method).

1. You need to add the following kernel boot parameters to `/etc/default/grub`:
   - `intel_iommu=on iommu=pt` – Enables passthrough (PCI Passthrough).
   - `isolcpus=<coreRange>` – Specify the cores you want to isolate for the vNTD to use exclusively. This parameter isolates the specified CPUs and their hyperthreads from the kernel.

     Caution: You **MUST** then specify the same cores when deploying the vNTD and when pinning the cores after deployment.

   - `intel_idle.max_cstate=1` – Ensures the CPU doesn't enter any power saving phases.
   - `transparent_hugepage=never` – Disables transparent hugepages to prevents the kernel from swapping pages in and out.
   - `default_hugepagesz=1G hugepagesz=1G hugepages=24` – Enables 1G huge pages. In the kernel boot parameter, the number required is 24 as Linux splits it across the sockets (assuming a dual socket host).

     Caution: Hugepages improve performance by changing how memory is allocated on the host. For **Debian** hosts using hugepages can expose a bug in the libvirt system. If you don't want to use huge pages, do not include `default_hugepagesz=1G hugepagesz=1G hugepages=24`.

   The following is an example modification, using all the possible parameters but you should check the documentation for your specific operating system before making modifications to the grub file:

```
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="crashkernel=auto rd.lvm.lv=centos/root rd.lvm.lv=centos/swap
rhgb quiet intel_iommu=on iommu=pt default_hugepagesz=1G hugepagesz=1G
hugepages=20 isolcpus=<coreRange> transparent_hugepage=never intel_idle.max_
cstate=1"
```

2. Save `/etc/default/grub`.
3. Identify if your server uses BIOS or EFI as a partition scheme. A quick way to do this is to check if you have this file on your system: `/sys/firmware/efi`. If you do, it's EFI and if not it's BIOS.
4. Run the appropriate command to rebuild grub with those new parameters:
   - If using BIOS as a partition scheme: `grub2-mkconfig --output=/boot/grub2/grub.cfg`
   - If using EFI as a partition scheme: `grub2-mkconfig -o /boot/efi/EFI/centos/grub.cfg`
5. Reboot the host server.

(Optional) Performance Optimization suggestions

In addition to the PCI Passthrough steps detailed above, you can perform some additional performance optimization on your server for line rate sampled traffic to the vNTD. These suggestions are only necessary in rare circumstances for sampled traffic to the vNTD which reaches close to line rate 10Gbps (with a sample rate of 1:1000 that would come from 10Tbps of network traffic).

> **Caution: THE FOLLOWING SUGGESTIONS ARE NOT COMPATIBLE WITH EVERY SYSTEM**. You should use them as guidance for adjusting your own system settings.

- You may need to configure the BIOS settings to disable power saving features and configure maximum PCIe settings. The following suggestions may help:
  - Disable any CPU power saving features:
    - Disable P State Control
    - Disable C State Control (only allow C states 0/1)
    - Disable T State Control
    - Disable HWPM State Control
  - Configure PCIe Settings
    - Maximum Payload – 2018 bytes
    - Maximum Read Request – 4096 bytes
- If your system uses hugepages, you may need to modify `/lib/systemd/system/dev-hugepages.mount` and ensure the `[Mount]` section is set as below:

```
[Mount]
What=hugetlbfs
Where=/dev/hugepages
Type=hugetlbfs
```

- You may need to edit the `/etc/sysctl.conf` file to allow the vNTD to use 100% of the CPU and the guest memory:
  - `kernel.nmi_watchdog = 0` – Disables watchdog to avoid unnecessary interrupts
  - `kernel.numa_balancing = 0` – Disables balancing to avoid sending anything to a second socket
  - `kernel.sched_rt_runtime_us = -1` – Allows the vNTD to use 100% of the CPU
  - `kernel.hung_task_panic = 0` – Prevents the kernel from thinking 100% CPU usage means a process has hung
  - `vm.nr_hugepages = 24` – Where `24` is the number of sockets in the server multiplied by 12 (in this case 2 sockets). This is to ensure all 12GB of guest memory is local to the socket it's running on. To check that all sockets have 12GB you can run the following command: `cat /sys/devices/system/node/node*/meminfo | fgrep Huge`

- You may need to move all interrupts to core 0 and set scaling_governor to performance for all cpus. The following command are an example of what you may need to run after you boot the machine:
  `for f in /proc/irq/*/smp_affinity; do echo 1 > $f; done`
  `echo performance | tee /sys/devices/system/cpu/cpu*/cpufreq/scaling_governor`
  To check this has worked as expected, you should see only `performance` returned multiple times when you run the following command: `cat /sys/devices/system/cpu/cpu*/cpufreq/scaling_governor`

> **Note:** To make this change persistent, you may need to add it to a startup file (e.g. RC2.local).

Isolate sampled traffic NICs to prepare for PCI Passthrough

> **Tip:** Most deployments require two NICs, if your deployment requires a different amount you can modify the process.

This process assigns the NIC hardware directly to the vNTD. Part of it must be completed prior to vNTD deployment and part of it must be completed after.

To remove a NIC device from the host OS

> **Caution:** The two NICs you detach are recommended to be on the same socket as the `<coreRange>` you isolated for the vNTD. This is to avoid possible performance problems if using inter-processor communication to access the NICs.

1. Ensure any other VMs which are using the PCI devices are stopped. **If you do not stop the VMs before detaching the PCI devices, the OS will crash.**
2. View a list of the NICs on this host using the following command: `ip addr`
   If you're not sure which NICs you need, you can check the speed of you NICs to help identify the correct ones. For example, this command returns the speed available on a NIC called enp4s0f3: `ethtool enp4s0f3 | grep Speed`

   > **Tip:** If you're confident with PCI addresses, you can work out the ones you need from the location and type of NIC and skip step 3 below. For example, the NIC name `enp4s0f3` tells you this is EtherNet PCI bus 4 Slot 0 Function 3 which means its PCI address should be `pci_0000_04_00_3`. You should always double check this on the host before using that PCI address to detach the NICs.

3. To find the PCI address of the NICs you're interested in (e.g. `pci_0000_04_00_3`), use the command `virsh nodedev-list --tree`
   Expand the PCI addresses until you locate one containing your NIC name as part of it's long PCI number (e.g. you could find `enp4s0f3` within `net_enp4s0f3_00_12_c0_02_d5_9a` under the PCI address `pci_0000_04_00_3`) .

Note: Only detach a top-level PCI parent device if you want to detach all its child devices.

4. Open virsh using the following command: `virsh`
5. Detach the NIC using the following command: `nodedev-detach<pciAddress>`

Tip: Keep a note of the addresses of the PCI nodes you detached to avoid mistakes during vNTD deployment and configuration. If you accidentally detach the wrong node, you can use the following command to reattach the node: `nodedev-reattach<pciAddress>`

Deploy the vNTD for high sampled traffic rates

To deploy a vNTD using virt-install

Before you run the install command, **libvirtd** must be running. Run the required command (below), replacing the following information:

- *<vmName>* – A name for this VM.
- *<qcow2DiskName1>* – The file name of the qcow2 disk (e.g. `new-swa-disk1.qcow2`).

  Tip: If you didn't save the disk files in the same location you're creating the VM, you can add a file path to the file name field (e.g. `/home/kvmfiles/vswa_9.7.5.133-030-disk1.qcow2`)

- *<diskType>* – The type of disk you want to use: `ide`, `virtio`, `sata`.
- *<managementInterface>* – The name of your management interface (e.g. eth0).
- <coreRange> – The 7 cores you isolated when you configured the host.

  Caution: You **MUST** specify the same cores you isolated when you configured the host. These are the same cores you will need to pin after deployment.

- *<pciAddress1>* – The pci address (e.g. pci_0000_04_00_1) of one of the NIC devices you removed from the host.
- *<pciAddress2>* – The pci address (e.g. pci_0000_04_00_2) of the other NIC device you removed from the host.

  Caution: When you run the commands below, they should not contain line breaks. Copying and pasting directly from the PDF may create unnecessary line breaks. Always paste into a text editor first and edit the commands.

vNTD:

Note: Hugepages improve performance by changing how memory is allocated on the host. For **Debian** hosts using hugepages can expose a bug in the libvirt system:

- To not use huge pages – Remove the following fields from the end of the command: `--memorybacking locked=yes,size=1,unit=GiB,nosharepages=yes`
- To use huge pages safely– Remove the above fields then add the following fields: `--memtune hard_limit=67108864`

```
virt-install -n <vmName> -r 12288 --os-type=linux --os-variant=rhel7 --disk
<qcow2DiskName1>,device=disk,format=qcow2,bus=<diskType> -w
source=<managementInterface>,type=direct,source_mode=bridge,model=virtio --import
--vnc --noautoconsole --cpu host-passthrough --vcpus
sockets=1,threads=1,cores=7,placement=static,vcpus=7,cpuset=<coreRange> --
```

```
memballoon=none --host-device=<pciAddress1> --host-device=<pciAddress2> --
memorybacking locked=yes,size=1,unit=GiB,nosharepages=yes
```

> **Tip:** The command above creates a new bridge for the management interface. If you already have a bridge (and its assigned interface) configured on your host, you can replace the `-w` networking command with the following syntax which enables you to specify the bridge you want to use rather than the interface: `-w source=<BridgeName>,type=bridge, model=virtio`

To deploy a vNTD using virsh

> **Caution:** Deployments using virsh do not support a serial console.

1. Create a new XML file and give it the name you want to use for the vNTD VM (`<vmName>`.xml).
2. From the appendix in this guide, copy the vNTD template into your XML file and replace the green placeholders with the necessary information for your new VM.

   > **Caution:** When you replace the `[coreRange]` placeholder in the **vNTD: High Sampled Traffic Rates** XML template, you **MUST** specify the same cores you isolated when you configured the host. These are the same cores you will need to pin after deployment.

3. Type the following command to create a VM from the xml file: `virsh define <vmName>.xml`
4. Type the following command to start the VM : `virsh start <vmName>`

Pinning vNTD virtual cores to isolated host cores (to complete PCI Passthrough)

Note: If you are using PCI Passthrough, you **must** complete this method before running traffic to your vNTD.

Caution: You **MUST** pin the same cores you isolated when you configured the host and specified when you deployed the vNTD.

Verifying the vNTD is deployed correctly

After deploying the vNTD using the method above, and the vCMS and vSWA using the main deployment in this document, you can verify the following areas before you use the system.

VM deployment

After deploying the vNTD, you can verify the virtual machine deployed correctly. Use the command `virsh list --all` to see all VMs currently running. You should see the new VM in that list.

Core allocation

During the deployment process, you had to handle CPU core pinning in 3 places and identify the same cores in each place:

- When configuring the Host OS isolated CPUs using grub
- When you deploy the vNTD VM (in the virt-install command or the virsh XML template)
- Post VM deployment, by pinning the cores using virsh

If any of these processes did not receive the correct core range, it could lead to incorrectly allocated cores. This can cause cores to be isolated by the Host OS and then be unavailable to any of the VMs, including the vNTD.

To check there are no isolated cores which haven't been pinned by the vNTD, perform the following method:

1. Check which cores were isolated from the host by looking at the settings in `cat/proc/cmdline`
2. Open the vNTD XML to check the core numbers pinned in vcpupin are the numbers you isolated during host configuration:
   `virsh dumpxml <vmName>`
3. Check the cores that the vNTD is using are definitely pinned by running the command below and checking that each core listed for the vNTD has a single host core associated with it, not a range of cores (e.g. for core 0 on the vNTD you see something like `0: 1` not `0: 1-31`).
   `virsh vcpupin <vmName>`

Memory capacity

The VMs require a minimum amount of memory allocated to them to perform effectively: 8GB for CMS, 12GB for SWA or vNTD. You can check your VMs have sufficient memory using the following command for each VM:

`virsh dominfo <vmName>`

The `Max memory` shown in the output is the memory size the VM has been configured with (in KiB).

Ordering of network interfaces

All interfaces on the vNTD are treated as PCI devices and each interface has a PCI-ID value which identifies the interface. The management interface (eth0) is assigned to the interface with the lowest PCI-ID value.

1. Access the vNTD pCLI. Note: The vNTD application may take 2 minutes to initialize when first installed.
2. Type the command: `show nic`
3. Identity the PCI Address shown for the Management interface.
4. Return to the host.
5. Type the following command to open the VM XML file: `virsh edit <vmName>`
6. Check the PCI address of the management interface matches the one from the pCLI.

If your PCI addresses match, then the management interface is allocated as expected. If they do not match, then your expected management interface did not have the lowest PCI-ID value. Edit the VM XML file to use the correct management interface PCI ID.

Appendix B – XML Templates for KVM Virsh Deployments

> **Caution:** Copying text out of a PDF can result in unexpected line breaks and character loss. If possible, use the online help version of the documentation. Alternately, first copy this set of commands into a plain text word processor (e.g. notepad) and check none of the hyphens or spaces have been removed and that no additional returns have been added.

These templates are for use with virsh deployments for normal sampled traffic rates and virsh deployments for high sampled traffic rates. The green placeholders must be replaced with the relevant information for your system. You may need to make other changes to the templates for your deployment needs, so you must check them carefully before deployment.

> **Caution:** Placeholder text is surrounded by square brackets which must be removed when you replace the placeholder with information. For example, `[vmName]` could be replaced by `myvNTD` (with no brackets).

vNTD: Normal Sampled Traffic Rates

```
<domain type='kvm'>
  <name>[vmName]</name>
  <memory unit='KiB'>12582912</memory>
  <currentMemory unit='KiB'>12582912</currentMemory>
  <os>
    <type arch='x86_64' machine='pc-i440fx-rhel6.0.0'>hvm</type>
    <boot dev='hd'/>
  </os>
  <features>
    <acpi/>
    <apic/>
  </features>
  <cpu mode='host-passthrough'>
    <topology sockets='1' cores='2' threads='1'/>
  </cpu>
  <clock offset='utc'/>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>restart</on_crash>
  <pm>
```

```
  <suspend-to-mem enabled='no'/>
  <suspend-to-disk enabled='no'/>
</pm>
<devices>
  <emulator>/usr/libexec/qemu-kvm</emulator>
  <disk type='file' device='disk'>
    <driver name='qemu' type='qcow2'/>
    <source file='[disk1FilePath]'/>
    <target dev='sda' bus='sata'/>
    <address type='drive' controller='0' bus='0' target='0' unit='0'/>
  </disk>
  <controller type='pci' index='0' model='pci-root'/>
    <controller type='sata' index='0'>
    <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'/>
  </controller>
  <controller type='usb' index='0'>
    <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x2'/>
  </controller>
  <interface type='bridge'>
    <source dev='[mgmtInterface]' mode='bridge'/>
    <model type='virtio'/>
    <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
  </interface>
  <interface type='bridge'>
    <source dev='[ExternalInterface]' mode='bridge'/>
    <model type='virtio'/>
    <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'/>
  </interface>
  <interface type='bridge'>
    <source dev='[InternalInterface]' mode='bridge'/>
    <model type='virtio'/>
    <address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x0'/>
  </interface>
  <serial type='pty'>
```

```
      <target port='0'/>
    </serial>
    <console type='pty'>
      <target type='serial' port='0'/>
    </console>
    <input type='mouse' bus='ps2'/>
    <input type='keyboard' bus='ps2'/>
    <graphics type='vnc' port='-1' autoport='yes'>
      <listen type='address'/>
    </graphics>
    <video>
      <model type='qxl' ram='65536' vram='65536' vgamem='16384' heads='1'
primary='yes'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x0'/>
    </video>
    <memballoon model='none'/>
  </devices>
</domain>
```

vNTD: High Sampled Traffic Rates

Caution: For **Debian** hosts:

- To not use huge pages – Remove the following section from the XML template before saving your XML file:

```
<memoryBacking>
     <hugepages>
         <page size='1048576' unit='KiB'/>
     </hugepages>
     <nosharepages/>
     <locked/>
</memoryBacking>
```

- To use huge pages – Add the following section to the XML template before saving your XML file:

```
<memtune>
    <hard_limit unit='KiB'>67108864</hard_limit>
</memtune>
```

```
<domain type='kvm'>
  <name>[vmName]</name>
  <memory unit='KiB'>12582912</memory>
  <currentMemory unit='KiB'>12582912</currentMemory>
  <memoryBacking>
    <hugepages>
      <page size='1048576' unit='KiB'/>
    </hugepages>
    <nosharepages/>
    <locked/>
  </memoryBacking>
  <vcpu placement='static' cpuset='[coreRange]'>7</vcpu>
  <os>
    <type arch='x86_64' machine='pc-i440fx-rhel6.0.0'>hvm</type>
    <boot dev='hd'/>
  </os>
  <features>
    <acpi/>
    <apic/>
```

```
</features>
<cpu mode='host-passthrough'>
  <topology sockets='1' cores='7' threads='1'/>
</cpu>
<clock offset='utc'/>
<on_poweroff>destroy</on_poweroff>
<on_reboot>restart</on_reboot>
<on_crash>restart</on_crash>
<pm>
  <suspend-to-mem enabled='no'/>
  <suspend-to-disk enabled='no'/>
</pm>
<devices>
  <emulator>/usr/libexec/qemu-kvm</emulator>
  <disk type='file' device='disk'>
    <driver name='qemu' type='qcow2'/>
    <source file='[disk1FilePath]'/>
    <target dev='sda' bus='sata'/>
    <address type='drive' controller='0' bus='0' target='0' unit='0'/>
  </disk>
  <controller type='pci' index='0' model='pci-root'/>
    <controller type='sata' index='0'>
    <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'/>
  </controller>
  <controller type='usb' index='0'>
    <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x2'/>
  </controller>
  <interface type='bridge'>
    <source dev='[mgmtInterface]' mode='bridge'/>
    <model type='virtio'/>
    <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
  </interface>
  <serial type='pty'>
    <target port='0'/>
```

```
    </serial>
    <console type='pty'>
      <target type='serial' port='0'/>
    </console>
    <input type='mouse' bus='ps2'/>
    <input type='keyboard' bus='ps2'/>
    <graphics type='vnc' port='-1' autoport='yes'>
      <listen type='address'/>
    </graphics>
    <video>
      <model type='qxl' ram='65536' vram='65536' vgamem='16384' heads='1'
primary='yes'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x0'/>
    </video>
    <hostdev mode='subsystem' type='pci' managed='yes'>
      <driver name='vfio'/>
      <source>
        <address domain='0x0000' bus='0x01' slot='0x00' function='0x2'/>
      </source>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x0c' function='0x0'/>
    </hostdev>
    <hostdev mode='subsystem' type='pci' managed='yes'>
      <source>
        <address domain='0x0000' bus='0x01' slot='0x00' function='0x3'/>
      </source>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x0d' function='0x0'/>
    </hostdev>
    <memballoon model='none'/>
  </devices>
</domain>
```

vCMS:

```
<domain type='kvm'>
  <name>[vmName]</name>
  <memory unit='KiB'>8388608</memory>
  <currentMemory unit='KiB'>8388608</currentMemory>
  <resource>
    <partition>/machine</partition>
  </resource>
  <os>
    <type arch='x86_64' machine='pc-i440fx-rhel7.0.0'>hvm</type>
    <boot dev='hd'/>
  </os>
  <features>
    <acpi/>
    <apic/>
  </features>
  <cpu>
    <topology sockets='1' cores='4' threads='1'/>
  </cpu>
  <clock offset='utc'>
    <timer name='rtc' tickpolicy='catchup'/>
    <timer name='pit' tickpolicy='delay'/>
    <timer name='hpet' present='no'/>
  </clock>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>restart</on_crash>
  <pm>
    <suspend-to-mem enabled='no'/>
    <suspend-to-disk enabled='no'/>
  </pm>
  <devices>
    <emulator>/usr/libexec/qemu-kvm</emulator>
    <disk type='file' device='disk'>
```

```
      <driver name='qemu' type='qcow2'/>
      <source file='[disk1FilePath]'/>
      <backingStore/>
      <target dev='hda' bus='ide'/>
      <alias name='ide0-0-0'/>
      <address type='drive' controller='0' bus='0' target='0' unit='0'/>
    </disk>
    <controller type='usb' index='0' model='ich9-ehci1'>
      <alias name='usb'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x7'/>
    </controller>
    <controller type='usb' index='0' model='ich9-uhci1'>
      <alias name='usb'/>
      <master startport='0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x0'
multifunction='on'/>
    </controller>
    <controller type='usb' index='0' model='ich9-uhci2'>
      <alias name='usb'/>
      <master startport='2'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x1'/>
    </controller>
    <controller type='usb' index='0' model='ich9-uhci3'>
      <alias name='usb'/>
      <master startport='4'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x2'/>
    </controller>
    <controller type='pci' index='0' model='pci-root'>
      <alias name='pci.0'/>
    </controller>
    <controller type='ide' index='0'>
      <alias name='ide'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x1'/>
    </controller>
```

```
    <controller type='virtio-serial' index='0'>
      <alias name='virtio-serial0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'/>
    </controller>
    <interface type='bridge'>
      <source dev='[mgmtInterface]' mode='bridge'/>
      <model type='virtio'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
    </interface>
    <serial type='pty'>
      <source path='/dev/pts/3'/>
      <target port='0'/>
      <alias name='serial0'/>
    </serial>
    <console type='pty' tty='/dev/pts/3'>
      <source path='/dev/pts/3'/>
      <target type='serial' port='0'/>
      <alias name='serial0'/>
    </console>
    <channel type='unix'>
      <source mode='bind' path='/var/lib/libvirt/qemu/channel/target/domain-26-
vcms/org.qemu.guest_agent.0'/>
      <target type='virtio' name='org.qemu.guest_agent.0' state='disconnected'/>
      <alias name='channel0'/>
      <address type='virtio-serial' controller='0' bus='0' port='1'/>
    </channel>
    <input type='tablet' bus='usb'>
      <alias name='input0'/>
      <address type='usb' bus='0' port='1'/>
    </input>
    <input type='mouse' bus='ps2'>
      <alias name='input1'/>
    </input>
    <input type='keyboard' bus='ps2'>
```

```
      <alias name='input2'/>
    </input>
    <graphics type='vnc' port='5904' autoport='yes' listen='127.0.0.1'>
      <listen type='address' address='127.0.0.1'/>
    </graphics>
    <video>
      <model type='cirrus' vram='16384' heads='1' primary='yes'/>
      <alias name='video0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x0'/>
    </video>
    <memballoon model='virtio'>
      <alias name='balloon0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x06' function='0x0'/>
    </memballoon>
  </devices>
  <seclabel type='dynamic' model='selinux' relabel='yes'>
    <label>system_u:system_r:svirt_t:s0:c176,c698</label>
    <imagelabel>system_u:object_r:svirt_image_t:s0:c176,c698</imagelabel>
  </seclabel>
  <seclabel type='dynamic' model='dac' relabel='yes'>
    <label>+107:+107</label>
    <imagelabel>+107:+107</imagelabel>
  </seclabel>
</domain>
```

vSWA:

```
<domain type='kvm'>
  <name>[vmName]</name>
  <memory unit='KiB'>12582912</memory>
  <currentMemory unit='KiB'>12582912</currentMemory>
  <resource>
    <partition>/machine</partition>
  </resource>
```

```
<os>
  <type arch='x86_64' machine='pc-i440fx-rhel7.0.0'>hvm</type>
  <boot dev='hd'/>
</os>
<features>
  <acpi/>
  <apic/>
</features>
<cpu>
  <topology sockets='1' cores='12' threads='1'/>
</cpu>
<clock offset='utc'>
  <timer name='rtc' tickpolicy='catchup'/>
  <timer name='pit' tickpolicy='delay'/>
  <timer name='hpet' present='no'/>
</clock>
<on_poweroff>destroy</on_poweroff>
<on_reboot>restart</on_reboot>
<on_crash>restart</on_crash>
<pm>
  <suspend-to-mem enabled='no'/>
  <suspend-to-disk enabled='no'/>
</pm>
<devices>
  <emulator>/usr/libexec/qemu-kvm</emulator>
  <disk type='file' device='disk'>
    <driver name='qemu' type='qcow2'/>
    <source file='[disk1FilePath]'/>
    <backingStore/>
    <target dev='hda' bus='ide'/>
    <alias name='ide0-0-0'/>
    <address type='drive' controller='0' bus='0' target='0' unit='0'/>
  </disk>
  <disk type='file' device='disk'>
```

```
      <driver name='qemu' type='qcow2'/>
      <source file='[disk2FilePath]'/>
      <backingStore/>
      <target dev='hdb' bus='ide'/>
      <alias name='ide0-0-1'/>
      <address type='drive' controller='0' bus='0' target='0' unit='1'/>
    </disk>
    <controller type='usb' index='0' model='ich9-ehci1'>
      <alias name='usb'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x7'/>
    </controller>
    <controller type='usb' index='0' model='ich9-uhci1'>
      <alias name='usb'/>
      <master startport='0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x0'
multifunction='on'/>
    </controller>
    <controller type='usb' index='0' model='ich9-uhci2'>
      <alias name='usb'/>
      <master startport='2'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x1'/>
    </controller>
    <controller type='usb' index='0' model='ich9-uhci3'>
      <alias name='usb'/>
      <master startport='4'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x2'/>
    </controller>
      <controller type='pci' index='0' model='pci-root'>
      <alias name='pci.0'/>
    </controller>
    <controller type='ide' index='0'>
      <alias name='ide'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x1'/>
    </controller>
```

```
    <controller type='virtio-serial' index='0'>
      <alias name='virtio-serial0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'/>
    </controller>
    <interface type='bridge'>
      <source dev='[mgmtInterface]' mode='bridge'/>
      <model type='virtio'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
    </interface>
    <interface type='bridge'>
      <source dev='[TelemetryInterface]' mode='bridge'/>
      <model type='virtio'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'/>
    </interface>
    <serial type='pty'>
      <source path='/dev/pts/5'/>
      <target port='0'/>
      <alias name='serial0'/>
    </serial>
    <console type='pty' tty='/dev/pts/5'>
      <source path='/dev/pts/5'/>
      <target type='serial' port='0'/>
      <alias name='serial0'/>
    </console>
    <channel type='unix'>
      <source mode='bind' path='/var/lib/libvirt/qemu/channel/target/domain-6-
vswa/org.qemu.guest_agent.0'/>
      <target type='virtio' name='org.qemu.guest_agent.0' state='disconnected'/>
      <alias name='channel0'/>
      <address type='virtio-serial' controller='0' bus='0' port='1'/>
    </channel>
    <input type='tablet' bus='usb'>
      <alias name='input0'/>
      <address type='usb' bus='0' port='1'/>
```

```
    </input>
    <input type='mouse' bus='ps2'>
      <alias name='input1'/>
    </input>
    <input type='keyboard' bus='ps2'>
      <alias name='input2'/>
    </input>
    <graphics type='vnc' port='5905' autoport='yes' listen='127.0.0.1'>
      <listen type='address' address='127.0.0.1'/>
    </graphics>
    <video>
      <model type='cirrus' vram='16384' heads='1' primary='yes'/>
      <alias name='video0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x0'/>
    </video>
    <memballoon model='virtio'>
      <alias name='balloon0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x06' function='0x0'/>
    </memballoon>
  </devices>
  <seclabel type='dynamic' model='selinux' relabel='yes'>
    <label>system_u:system_r:svirt_t:s0:c414,c963</label>
    <imagelabel>system_u:object_r:svirt_image_t:s0:c414,c963</imagelabel>
  </seclabel>
  <seclabel type='dynamic' model='dac' relabel='yes'>
    <label>+107:+107</label>
    <imagelabel>+107:+107</imagelabel>
  </seclabel>
</domain>
```