



Illumio ASP 18.1 VEN Operations

Last Updated: 05/17/2018

Table of Contents

Product Version	5
About Illumio	5
Illumio ASP Training	5
Search Knowledge Base and Documentation.....	5
Illumio Support.....	5
Recommended Skills	5
How To Use This Guide	6
Related Documentation	6
Notational conventions	7
Overview of VEN Software Architecture and Description of Components	7
VEN Architectural Diagram	7
Description of VEN Components	8
Management Interfaces for the PCE and VEN.....	10
Cycle of Common VEN Tasks	11
illumio-ven-ctl General Syntax.....	12
illumio-ven-ctl: Linux Two Dashes, Windows One Dash	12
Set the PATH Environment Variable	12
VEN Installation/Uninstallation - See VEN Deployment Guide.....	12
VEN Activation/Deactivation - See VEN Deployment Guide	13
Verify VEN Version Number	13
VEN Startup	13
VEN Shutdown	13
VEN Status	14
VEN Disable/Enable.....	14
VEN Suspend/Unsuspend	15
Linux - Before Suspending, Backup iptables/NAT rules	15

Suspend/Unsuspend Commands	16
Results of Suspending/Unsuspending	16
Upgrading the VEN	17
Upgrading All VENs via PCE	17
Linux VEN Upgrade	17
Preserve Custom User Name.....	17
Set Non-default Data Directory before Upgrade	17
Running the Upgrade	18
Windows VEN Upgrade	18
Diagnostics and Troubleshooting.....	18
Enable Windows Application Layer Enforcement (ALE).....	18
Linux ignored_interface Inhibits PCE Policy Updates.....	19
Connectivity Checking Tool.....	19
Troubleshooting Steps	19
Appendix - Basic Theory of VEN Operations.....	21
VEN Installation and Uninstallation	21
Linux Pairing Script for VEN Repo: pair	21
RPM Installation.....	22
Packages and Kernel Modules.....	22
Windows Pairing Script pair.ps1	23
VEN-to-PCE Communications	24
Polling Intervals of VEN Operations	24
Heartbeat mechanism	25
Details about Isolation Mechanism	25
SecureConnect.....	26
Sampling Mode	26
Linux nf_conntrack_tcp_timeout_established set to 8 Hours	26
VEN Uninstallation	26
VEN Activation or Pairing	27
VEN Startup	28

VEN Shutdown	28
VEN Status	28
Workload Policy States.....	28
Automatic History of Firewall Changes.....	29
VEN Traffic Logging.....	29
Contents of Traffic Flow Logs	29
Querying flow log databases.....	29
Linux Database Query Examples	30
Window Database Query Examples.....	30
Summary of VEN and Useful OS commands	30
Revision History: Illumio ASP VEN Operations	31

Product Version

- Current PCE Version: 18.1.0 (Standard release)
- Current VEN Version: 18.1.0 (Standard release)

About Illumio

Copyright © 2013 - 2018 Illumio, Inc., 160 San Gabriel Drive, Sunnyvale, CA 94086

Illumio's products and services are built on our patented technologies. For information on Illumio's patents and patent applications, see <https://www.illumio.com/patents>.

Illumio ASP Training

Illumio offers a wide, focused training curriculum for Illumio Adaptive Security Platform (ASP), from beginning to advanced topics.

To see available courses, login to your [Illumio Support account](#) and select the **Training** tab.

Search Knowledge Base and Documentation

For useful short articles about Illumio ASP, login to your [Illumio Support account](#) and select the **Knowledge Base** or **Documentation** tabs.

Illumio Support

If you cannot find what you are looking for in this document or the support knowledge base and documentation, contact us at:

- support@illumio.com
- +1-888-631-6354
- +1-408-831-6354

Recommended Skills

Illumio recommends that you be familiar with the following:

- Your organization's security goals

- Solid understanding of Illumio ASP
- General computer system administration of Linux and Windows operating systems, including startup/shutdown, common processes or services
- Linux shell (bash), Windows PowerShell, or both
- Understanding TCP/IP networks, including protocols and well-known ports
- Familiarity with PKI certificates

How To Use This Guide

This document shows you how use `illumio-ven-ctl` and other commands to administer the Illumio Virtual Enforcement Node (VEN) on a managed Workload for operational tasks such as start/stop, suspend, and other functions on the VEN and with the Policy Compute Engine (PCE) in an on-premise deployment.

The *VEN Operations Guide* has several main divisions:

- [Overview of VEN Software Architecture and Description of Components](#)
- Command-line-oriented sections with syntax examples for `illumio-ven-ctl`, the key Illumio-provided script for managing the VEN.
- The [Basic Theory of VEN Operations](#), which describes the detailed effects of command usage, interactions of the VEN components, interactions with the Policy Compute Engine (PCE), communications between the VEN and the PCE, and other considerations.

This guide focuses on the commands installed by the recommended deployment model: single package installation directly on the workload, which installs the `illumio-ven-ctl` script for operational functions. The Illumio Repo model is detailed in the [VEN Deployment Guide](#).

Terminology: Activation or Pairing

These following terms indicate the same function: putting the workload under managed control by the PCE:

- The terms *activation/deactivation a VEN* is used for the single package deployment model, which is downloaded and installed on the workload and then the `illumio-ven-ctl` command.
- The term *pairing/unpairing a VEN* is used for the Illumio Repo deployment model and also in the PCE Web Console, which relies on the `pair` or `pair.ps1` script.

Related Documentation

Illumio ASP documentation is available from the [Support portal](#).

- *PCE Web Console* guide: working with Illumination, designing policy, creating labels, and provisioning and administering managed workloads
- *PCE Deployment* guide: requirements, planning, and installing the Policy Compute Engine (PCE)
- *PCE Operations* guide: *common* operational tasks on the Policy Compute Engine (PCE)
- *PCE REST API* guide: Programming Illumio ASP

- *VEN Deployment* guide: installing and activating the Virtual Enforcement Node (VEN) on workloads
- *VEN Operations* guide: administering the Virtual Enforcement Node (VEN) directly on managed workloads

Notational conventions

- *New term*: Newly introduced terminology is indicated by italics. Example: *activation code*.
- *Command-line examples* are in monospace. Example: `illumio-ven-ctl --activate`
- *Arguments* on command lines are *monospace italics*. Example: `illumio-ven-ctl --activate activation_code`

Overview of VEN Software Architecture and Description of Components

A *workload* with an installed VEN is a computer system you want to secure. A secured workload is known as a *managed workload*. You control the VEN's operations through the PCE user console or from the command-line on the VEN itself.

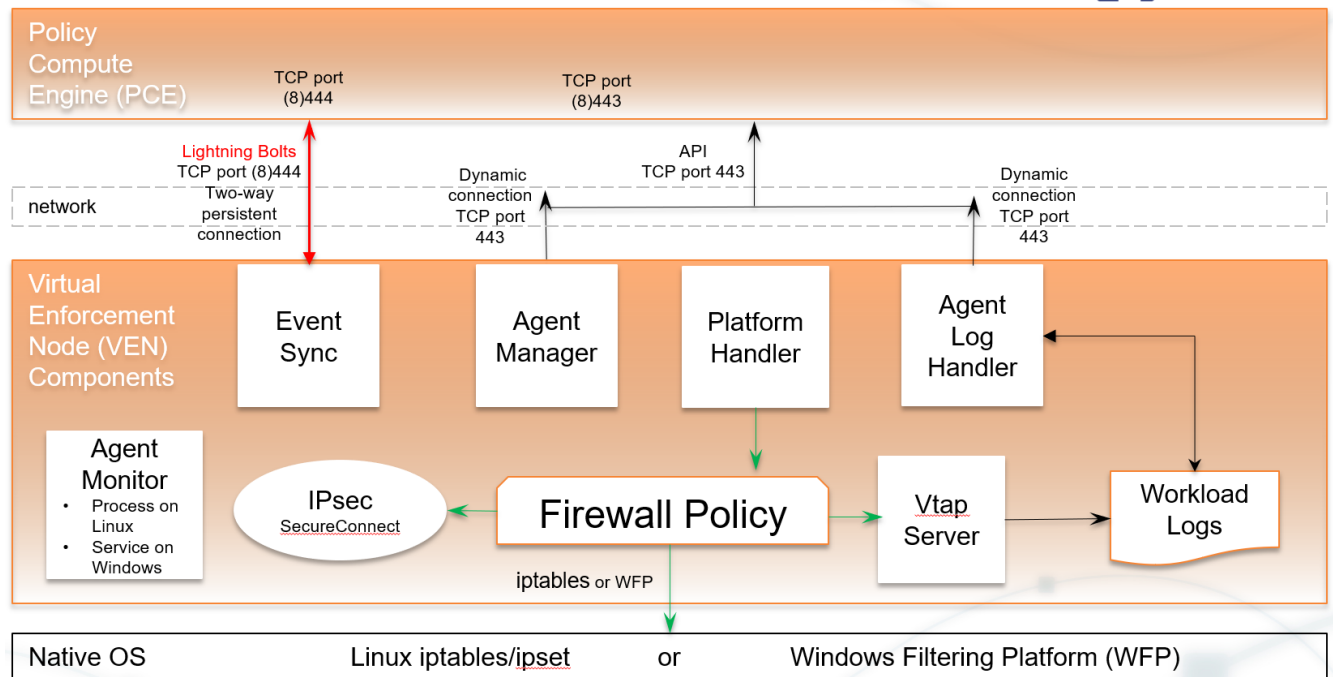
The VEN resides in the guest OS as a lightweight, multiple-process application with a minimal footprint.

- It interacts with the native networking interfaces to enforce policy received from the PCE.
- It operates periodically at maximum speed, remaining in the background as much as possible.
- It uses configurable operational modes to minimize the impact to workloads.
- It provides details of traffic flow data collected via logging, summarized by the VEN, and viewable in the PCE's Support Reports.

VEN Architectural Diagram

At startup, the VEN instantiates the following processes or services.

VEN Architecture



1. The VEN reports the workload's context (status and attributes) to the PCE.
2. The PCE computes a unique security policy for each managed workload and transmits it to the VEN.
3. The VEN receives the policy and programs native OS mechanisms on the workload:
 - a. Linux: iptables and ipsets
 - b. Windows, the Windows Filtering Platform (WFP), including WFP Optimization by default
4. The PCE lets those mechanisms enforce that policy.

Description of VEN Components

VEN Process	Description	Linux User	Windows User
AgentManager	<ul style="list-style-type: none"> • Manages uninstallation and upgrades • Mines the workload's system information, such as network interfaces, and listening processes, to send to the PCE • Sends heartbeats to the PCE. 	root	Administrator

VEN Process	Description	Linux User	Windows User
EventSync	Maintains a dynamic connection to the PCE to receive asynchronous notifications.	<ul style="list-style-type: none"> For SaaS customers: illumio For on-premise customers: ilo-ven 	LOCAL SERVICE
PlatformHandler	Handles: <ul style="list-style-type: none"> Firewall configuration via native OS mechanisms Tamper detection and protection Upgrades and uninstallation 	root	Administrator
AgentLogManager	Uploads logs to the PCE for events and traffic flows.	<ul style="list-style-type: none"> For Illumio Secure Cloud customers: illumio For on-premise customers: ilo-ven 	LOCAL SERVICE
vtapServer	Receives traffic flow data logs and records them in a SQLite database.	root	LOCAL SERVICE
AgentMonitor	Monitors VEN processes or services and restarts them when necessary.	root	LOCAL SERVICE
IPsec	Illumio's optional SecureConnect configures Internet Protocol Security (IPsec), a set of protocols to enforce security for IP networks. IPsec can be configured to use cryptography.	root	LOCAL SERVICE

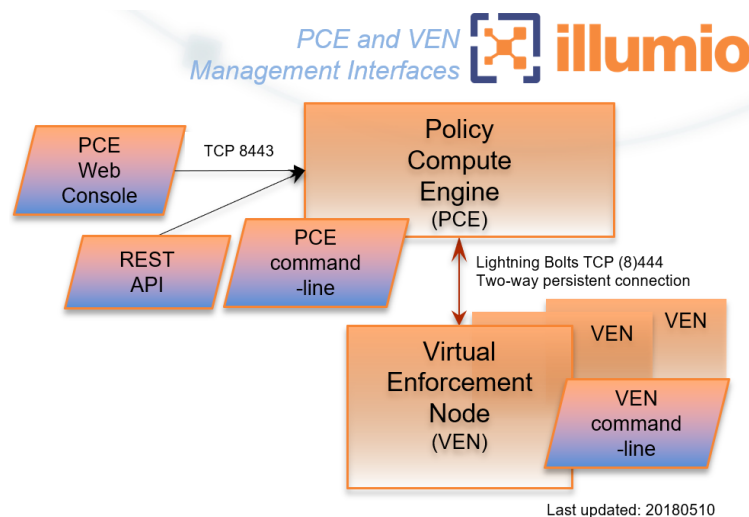
See these additional discussions with greater detail:

- [Basic Theory of VEN Operations:](#)
 - [Workload Policy States](#)

- [VEN-to-PCE Communications](#)
- [VEN Traffic Logging](#)

Management Interfaces for the PCE and VEN

You can manage the PCE and the VEN via several interfaces.



Interface	Notes	See Also
PCE Web Console	With the PCE Web Console, you can perform many of the same functions described in this guide.	PCE Web Console guide at Documentation
Command-line on the PCE	Use of the command-line directly on the PCE. One of the primary management tools on the command-line is the <code>illumio-pce-ctl</code> script.	<code>illumio-pce-ctl</code> in the PCE Operations guide at Documentation
REST API	<p>With the Illumio ASP REST API, you can perform many of managerial functions The endpoint for REST API requests is the PCE Leader node itself, not the workload; the REST API does not communicate directly with the VEN.</p> <p>One use of the REST API is to automate the management of large groups of workloads, rather than each workload individually.</p>	REST API guide at Documentation

Interface	Notes	See Also
Command-line on the VEN workload	Use of the command-line directly on the VEN workload. One of the primary management tools on the command-line is the <code>illumio-ven-ctl</code> script.	<code>illumio-ven-ctl</code> in the VEN Operations guide at Documentation



Cycle of Common VEN Tasks



illumio-ven-ctl General Syntax

The `illumio-ven-ctl` is a primary tool for managing VENs on individual workloads. The script varies slightly by platform.

illumio-ven-ctl: Linux Two Dashes, Windows One Dash

Platform	Command	Notes
Linux	<code>illumio-ven-ctl</code>	 Parameters for the script are preceded by a <i>double</i> hyphen: <code>--option1 var --option2 var ...</code>
Windows	<code>illumio-ven-ctl.ps1</code>	In Windows PowerShell, the <code>.ps1</code> extension is <i>optional</i> .  Parameters for the script are preceded by a <i>single</i> hyphen: <code>-option1 var -option2 var ...</code>

Set the PATH Environment Variable

For easier invocation of the script, you might want to set your PATH variable to the directory where the VEN commands are located:

- Linux: default location is `/opt/illumio/bin`
- Windows: default location is `C:\Program Files\Illumio`

VEN Installation/Uninstallation - See VEN Deployment Guide

How you install, uninstall, and upgrade the VEN is highly configurable. This extensive topic is detailed in the companion guide *VEN Deployment Guide*. See [Related Documentation](#).

VEN Activation/Deactivation - See VEN Deployment Guide

Because VEN activation frequently accompanies deployment, it is detailed in the companion guide *VEN Deployment Guide*. See [Related Documentation](#).

Verify VEN Version Number

You can verify the version of the VEN software in several different ways:

- In the PCE Web Console
- On the Workload itself.
 - Windows: Any of the following:
 - Examine the columns in **Uninstall or change a program** or Task Manager
 - Examine the **Properties > Details** tab of the `venAgentMgr.exe` or `venPlatformHandler.exe`
- With the REST API, the `agent-version` key and value are returned in the payload of every response.

VEN Startup

Via system boot files, the VEN starts when the workload is booted. The VEN can also be started manually.

Platform	Command	Notes
Linux	<ul style="list-style-type: none"> • <code>/etc/init.d/illumio-firewall</code> • <code>/etc/init.d/illumio-ven-ctl start</code> 	<ul style="list-style-type: none"> • Installs ipset kernel module if necessary, sets iptables/ipsets to desired state. • Initializes and starts the daemon processes needed for VEN operation.
Windows	None needed	The Service Control Manager (SCM) starts all VEN services at boot.

VEN Shutdown

At shutdown, the VEN sends a “goodbye” message to the PCE. The PCE marks the Workload as offline and initiates a policy recomputation. After the new policy is distributed throughout the network, the Workload without the VEN is effectively isolated from the network.

Linux Workload Shutdown

- `illumio-ven-ctl stop` stops all VEN processes.
- The VEN sends a “goodbye” message to the PCE.

Windows Workload Shutdown

- Service Control Manager (SCM) stops all VEN services
- The VEN sends a “goodbye” message to the PCE

VEN Status

To see the status of the VEN on the workload, run this command.

```
$ illumio-ven-ctl status
```

VEN Disable/Enable

If you want to install the VEN but activate it at a later time, you can disable the VEN after you first install it.

For example, you can load the VEN on machine image and disable the VEN. See considerations regarding preparing a “Golden Master” in the [VEN Deployment guide](#).

Platform	Action		Notes
Linux	<ul style="list-style-type: none"> • Enable • Disable 	<pre>\$ illumio-ven-ctl enable</pre> <pre>\$ illumio-ven-ctl disable</pre>	
Windows	<ul style="list-style-type: none"> • Enable • Disable 	<pre>\$ illumio-ven-ctl.ps1 enable</pre> <pre>\$ illumio-ven-ctl.ps1 disable</pre>	When you disable the VEN, all Illumio-based filters are removed from the Windows Filtering Platform (WFP).

VEN Suspend/Unsuspend

Suspending a VEN isolates a VEN on a workload so you can troubleshoot possible communication issues to determine the cause of any anomalous behavior.

- When you suspend a VEN, any rules programmed into the workload's iptables (including Custom iptables rules) or Windows Filtering Platform (WFP) firewalls are removed completely and all VEN software processes are shut down. The VEN's connectivity and policy sync status are changed to **Suspended**.
- Workloads communicating with the suspended VEN continue to have rules programmed into iptables or WFP.
- You can unpair a workload while its VEN is suspended.
- With the PCE Web Console you can change the policy state of the workload while the VEN is suspended. When the VEN is unsuspended, the new policy state is applied.

Linux - Before Suspending, Backup iptables/NAT rules


Before you suspend a Linux VEN, back up the workload's custom iptables rules or NAT rules. For information about backing up and restoring the iptables, see <http://sharadchhetri.com/2014/02/22/how-to-backup-and-restore-iptables-on-linux-systems/>.

After a workload is suspended, you need to restore the rules on the workload because all custom iptables or NAT rules are removed from the workload. At the time of suspension, the VEN informs the PCE that it is in suspended state.

If the PCE does not receive this notification, you must mark the workload as "suspended" in the PCE web console. See the [PCE Web Console guide](#).

If you do not mark the VEN as suspended in the PCE, after one hour the PCE assumes the Workload is offline and removes it from policy, which effectively isolates the workload from the network.

Suspend/Unsuspend Commands

Platform	Action	Command	Notes
Linux	<ul style="list-style-type: none"> Suspend Unsuspend 	<pre>\$ illumio-ven-ctl suspend Suspending the VEN... The VEN has been suspended. PCE was notified.</pre> <pre>\$ illumio-ven-ctl unsuspend Unsuspending the VEN... The VEN has been unsuspended. PCE was notified.</pre>	 Be sure to backup your configuration as described in Linux - Before Suspending, Backup iptables/NAT rules.
Windows	<ul style="list-style-type: none"> Suspend Unsuspend 	<pre>PS C:\Program Files\Illumio> illumio-ven-ctl.ps1 suspend Suspending the VEN... The VEN has been suspended. PCE was notified.</pre> <pre>PS C:\Program Files\Illumio> illumio-ven-ctl.ps1 unsuspend Unsuspending the VEN... The VEN has been unsuspended. PCE was notified.</pre>	

Results of Suspending/Unsuspending

- The workload still appears in the PCE in the workloads list page and Illumination map.
- The workload can only be unpaired from the PCE.
- An organization event (server_suspended) is logged. This event is exportable in Common Event Format (CEF) and Log Event Extended Format (LEEF). This event has a severity of WARNING.
- Heartbeats or other communication are not expected, but if received, communication is logged by the PCE.
- If the PCE is rebooted, the VEN remains suspended.
- Any custom iptables rules are removed and must be reconfigured manually.

- If SecureConnect has been enabled on the VEN, it is not disabled.

When a VEN is unsuspended:

- The PCE is informed that the VEN is no longer suspended and is able to receive policy from the PCE.
- If existing rules affect the unsuspended workload, the PCE reprograms those rules.
- An organization event (`server_unsuspended`) is logged. The event has a severity of `WARNING`. The event is exportable in Common Event Format (CEF) and Log Event Extended Format (LEEF).
- The workload revert sto its policy state prior to Suspended.
- Custom iptables rules are configured back into the iptables.

Upgrading the VEN

There are several mechanisms for upgrading the VEN.

Upgrading All VENs via PCE

Run this command on one of the Leader nodes in your deployment to upgrade all of the VENs in your environment:

```
$ ./illumio_pce/illumio-pce-ctl ven-upgrade
Reading /opt/pce_config/etc/runtime_env.yml.
VEN upgrade initiated.
```

Linux VEN Upgrade

Preserve Custom User Name

If you installed the VEN with your own user name, for upgrade you need to specify that same user name with the `VEN_NONPRIV_USER` environment variable. See details in the [VEN Deployment guide](#).

Set Non-default Data Directory before Upgrade

If you previously installed the VEN to different installation and data directories with the `VEN_DATA_DIR` environment variable, you need to specify the same value for `VEN_DATA_DIR` before upgrade. See details in the [VEN Deployment guide](#).

Running the Upgrade

To start the upgrade:

```
$ /opt/illumio/admin/upgrade
or
$ /opt/illumio/admin/upgrade -y # The -y option suppresses the confirmation prompt.
```

A record of the upgrade is stored in `/opt/illumio/log/upgrade.log`.

Windows VEN Upgrade

1. Change directories to the following path:
PS> **cd %ProgramFiles(x86)%\Illumio\admin**
2. Start the upgrade:
PS> **illumio-ven-ctl.ps1 upgrade**
3. When asked if you want to upgrade the VEN, type **yes** and then press **Enter** . You can also suppress this prompt. See **yes** parameter above.

A record of the upgrade is stored in `C:\ProgramData\Illumio\log\install.log`

Diagnostics and Troubleshooting

This section describes some important System administration considerations on Windows, a useful tools, and a generalized set of steps for troubleshooting.

Enable Windows Application Layer Enforcement (ALE)

If you have disabled Windows Application Layer Enforcement (ALE), you need to re-enable it.

The ALE is a Windows component to examine all packets and decides which packets should be sent to the TCP/IP stack. (For more information about ALE itself, see [https://msdn.microsoft.com/en-us/library/windows/desktop/bb451830\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/bb451830(v=vs.85).aspx).)

ALE is enabled by default. If you disable it, all packets are sent to the TCP/IP stack, which marks them as illegal and drop them. You also lose visibility into the packets. Such packets might be exploited for a Denial of Service (DOS) attack on the workload.

Linux ignored_interface Inhibits PCE Policy Updates

Transitioning an enforced VEN's interface in and out of `ignored_interface` might drop the dynamic, long-lived connections maintained by the AgentManager component between the VEN and the PCE. (See the description of the AgentMonitor in [Description of VEN Components](#).)

When a VEN interface is placed in `ignore_interface` list, `conntrack` is disabled. (The `conntrack` table on Linux stores information about network connections.) If the connection on TCP port 8444 to the PCE is reinitialized, any arriving packets from the PCE are dropped, because the packets do not have any state in `conntrack`.

The [VEN heartbeat](#) eventually restores connections, but meanwhile the VEN does not implement any policy sent via lightning bolt from the PCE.

Connectivity Checking Tool

A connectivity checking tool for workloads is available in the installation package and online. See the documentation and download at [VEN Connectivity Checking Tool](#).

Troubleshooting Steps

Follow these steps to identify the cause of workload connectivity issues. If a workload is unreachable or cannot reach other workloads/PCE, follow these steps to troubleshoot.

1. Determine if all workloads are unable to communicate or just a subset of the workloads are reported as disconnected. If PCE reports that all workloads are offline, check if PCE is reachable from workloads.
2. If a subset of workloads are down, check if there are differences in network configuration between those and the workloads that are connected, and if they are contributing to PCE being unreachable.
3. Check if any workloads unable to communicate are located behind NAT devices, firewalls, or remote data centers. See [Connectivity Checking Tool](#).
4. For on-premise Illumio ASP deployments, ensure TCP port 443 and 444 on workloads are allowed to open to the PCE.
5. If running in a public cloud instance:
 6.
 - a. For AWS, ensure security groups permit TCP ports 443 and 444.
 - b. For Azure, ensure that Endpoints are configured to allow traffic.
7. Check the status of the Illumio-specific processes and ensure that they are running and active:
 - On Linux: run `/opt/illumio/bin/agent_status -a` or `illumio-ven-ctl status`
 - On Windows: execute `get-service` in the PowerShell
 - a. Ensure the following processes are running and active:
 - On Linux: AgentManager, EventSync, IPSec, PlatformHandler, AgentLogManager, VtapServer, AgentMonitor
 - On Windows: venAgentLogMgrSvc, venEventSyncSvc, venPlatformHandler, venVtapServerSvc, ilowfp

8. Review Illumio log files to find any errors generated by the system (sudo required):

- Logs in Data_Dir/log directory
- To look for any errors in the log files, execute `grep -ir ERROR *`

9. To check for firewall updates, view `platform.log` file. Look for logs related to firewall updates; for example:

```
2014-07-26T22:20:41Z INFO:: Enforcement mode is: XXXX
2014-07-26T22:20:41Z INFO:: Is fw update yes
2014-07-26T22:20:41Z INFO:: Is ipset update yes
2014-07-26T22:20:41Z INFO:: saved fw-json
```

10. Check `/heartbeats/events/evsync.log` for logs related to update messages from the PCE. The following are example heartbeats:

```
2014-07-26T22:43:12Z Received HELLO from EventService.
2014-07-26T22:43:12Z Sent ACK to EventService.
Events - f/w updates etc.
014-07-26T22:34:11Z Received EVENT from EventService.
2014-07-26T22:34:11Z Added EVENT from EventService to PLATFORM handler thread message queue
```

```
iptables-save | grep 443 | grep allow_out
-A tcp_allow_out -d 54.185.43.60/32 -p tcp -m multiport --dports 443 -m conntrack --ctstate NEW -j
NFLOG --nflog-prefix "0x800000000000025f " --nflog-threshold 1
-A tcp_allow_out -d 54.185.43.60/32 -p tcp -m multiport --dports 443 -m conntrack --ctstate NEW -j
ACCEPT
-A tcp_allow_out -d 204.51.153.0/27 -p tcp -m multiport --dports 443 -m conntrack --ctstate NEW -j
NFLOG --nflog-prefix "0x8000000000000265 " --nflog-threshold 1
-A tcp_allow_out -d 204.51.153.0/27 -p tcp -m multiport --dports 443 -m conntrack --ctstate NEW -j
ACCEPT
iptables-save | grep 444 | grep allow_out
-A tcp_allow_out -d 54.185.43.60/32 -p tcp -m multiport --dports 444 -m conntrack --ctstate NEW -j
NFLOG --nflog-prefix "0x8000000000000266 " --nflog-threshold 1
-A tcp_allow_out -d 54.185.43.60/32 -p tcp -m multiport --dports 444 -m conntrack --ctstate NEW -j
ACCEPT
```

Run the following commands on the workload to get a copy of the logs and configured firewall settings.

Linux

- `iptables-save`
- `ipset -L`

Windows

- Navigate to **Start -> Control Panel -> System and Security -> Windows Firewall -> Advanced Settings**
- Navigate across inbound and outbound rules to look for relevant firewall/filtering configuration.

Appendix - Basic Theory of VEN Operations

The section describes in greater detail the effects, behaviors, and other aspects of how VEN operations work.

VEN Installation and Uninstallation

Linux Pairing Script for VEN Repo: pair

Below is an example of Linux pairing script with annotation. The pairing script works with the VEN Repo model of deployment, not the single-package installation (the preferred deployment model), which uses `illumio-ven-ctl --activation` option. Both mechanisms accomplish the same purpose: bring a workload under management by Illumio ASP.

```
rm -fr /opt/illumio/scripts && \
umask 027 && mkdir -p /opt/illumio/scripts && \
curl https://repo.bigcompany.com/scp2HSPFGj2C82BVDYJf2BCXlzsGWX03/pair -o /opt/illumio/scripts/pair && \
chmod +x /opt/illumio/scripts/pair && \
/opt/illumio/scripts/pair \
--repo-host repo.bigcompany.com --repo-dir scp2HSPFGj2C82BVDYJf2BCXlzsGWX03/ --repo-https-port 443 \
--management-server scp2.bigcompany.com:443 \
--activation-code 0123456789abcdef
```

1. Removes any existing `/opt/illumio/scripts` directory
2. Changes umask to 027 to prevent the group-write and others-read,write,execute permissions as it creates `/opt/illumio/scripts` directory
3. Uses curl to download the pair script from repo.illumio.com and store it under `/opt/illumio/scripts`.
4. Changes script permissions to allow execution
5. Runs the `/opt/illumio/scripts/pair` with the following command line options:
6. Line 6 `--repo-host`, `--repo-dir`, and `--repo-port` to check for VEN software updates
7. Line 7 `--management-server`: to communicate with the PCE
8. Line 8 `--activation code` to authenticate the VEN to the PCE and authorize the VEN to pair

The pair script installs the VEN packages on the application workload and pairs the VEN with the PCE. The output of pair is captured in `/var/log/illumio_install.log`.

The script then performs the following:

- a. For yum-based OSes, updates `releaserver` to release name in `/etc/yum.conf` using `sed`

- b. Removes VEN files leftover from a previous failed installation
- c. Downloads the Illumio package signing GnuPG public key from repo using curl
- d. Stores the key in /tmp and make it root accessible
- e. Installs the key into rpm or apt-key commands
- f. On Red Hat, disables the subscription manager plugin (set enabled=0 in /etc/yum/pluginconf.d/subscription-manager.conf)
- g. Creates the Illumio yum repo at /etc/yum.repos.d/illumio.repo or apt-get repo at /etc/apt/sources.list.d/illumio_repo.list
- h. Populates the repo file with repo URI and other information
- i. Installs dependencies followed by illumio-agent-control and illumio-agent-vtapserver packages from Illumio repo
- j. Places upgrade hold on VEN packages so that they don't get upgraded automatically
- k. Checks if ipset kernel module is installed (if not, the process fails)
- l. Runs /opt/illumio/bin/init_Platform script with "start" option
- m. Generates activation file /opt/illumio/etc/agent_activation.cfg
- n. Invokes /opt/illumio/bin/agent_status to activate the VEN
- o. Restores releaserver=latest in /etc/yum.conf

RPM Installation

RPM installation performs the following operations:

- Creates the `ilo-ven` user and group, unless a custom username is specified at install.
- Starts the Illumio security service to manage the following:
 - a. In [Idle state](#), Illumio security service does nothing.
 - b. Loads kernel modules: `ip_tables`, `iptables_filter`, `nf_conntrack`, `nf_conntrack_ipv4`, `nf_conntrack_ftp`, `ipt_LOG`, `ip_set`, `ip6_tables`, `ip6table_filter`, `nf_conntrack_ipv6`, `ip6t_LOG`
 - c. Sets `net.netfilter.nf_conntrack_tcp_timeout_established` to 8 hours (28,800 seconds). See also [Linux nf_conntrack_tcp_timeout_established set to 8 Hours](#)
 - d. Disables the system firewall service `iptables`
 - e. Stops system firewall service `iptables`
 - f. Saves existing `iptables` rules if any
 - g. Loads `iptables` rules computed from PCE firewall policy
 - h. Starts the VEN components described in [Description of VEN Components](#).

Packages and Kernel Modules

Some packages, such as Illumio's SecureConnect StrongSwan for enforcing IPsec, are included as part of the VEN package. Other packages are installed on the host itself if they are not already present.

If the following packages are not installed on the workload, via RPM dependencies the VEN installation downloads and install them.

1. `curl`: Used for HTTPS client functionality

2. dnstools: Used for DNS client functionality
3. uuid-runtime: Used for generating UUIDs
4. ipset: Used for ipset functionality
5. libnftnl: Used for communicating with the Net Filter module
6. libcap2: Used for selectively enabling/disabling capabilities
7. libgmp10: Used for multi-precision arithmetic
8. bind-utils: Used for DNS client functionality
9. iptables and iptables-ipv6: Used for iptables functionality
10. apt-transport-https (for apt-based OS): Used for HTTPS transport for apt

If the following kernel modules are not installed, the VEN downloads and installs them:

- ipset

Windows Pairing Script pair.ps1

Below is an example of Windows pairing script.

```
Set-ExecutionPolicy -Scope process remotesigned -Force;
Start-Sleep -s 3;
(New-Object System.Net.WebClient).DownloadFile("https://repo.illum.io/ scp2HSPFGj2C82BVDYJf2BCXlzsGWX03/
pair.ps1", "$pwd\Pair.ps1"); .\Pair.ps1
-repo-host repo.illum.io
-repo-dir scp2HSPFGj2C82BVDYJf2BCXlzsGWX03/
-repo-https-port 443 -management-server scp2.illum.io:443
-activation-code 0123456789abcdef;
Set-ExecutionPolicy -Scope process undefined -Force;
```

In the above example, the Windows pairing script performs the following:

- Changes execution policy of the host PowerShell process to remotesigned.
- Using .NET framework WebClient class, downloads pair.ps1 from VEN repository and stores it in the current directory
- Runs the pair.ps1 script with the following command line options:
 - repo host, repo directory, and repo port: Used by the VEN to check for VEN software updates
 - management server: Used by the VEN to communicate with the PCE
 - activation code: Used by the PCE to authenticate and authorize the VEN during pairing process
- The pair script installs the VEN packages on the application workload and pairs the VEN with the PCE. The output of pair.ps1 is captured in %TMP%\illumio.log. The script performs the following steps:
 - Retrieves VEN MSI package from repo using .NET framework WebClient class
 - Launches msixexec.exe to install the downloaded package
 - Generates agent_activation.cfg file with PCE information
 - Retrieves agent activation status and displays it

MSI Installation

The MSI installation performs the following:

1. Creates VEN registry key under HKLM\Software\Illumio
2. Adds Illumio code signing certificate to “Trusted Publisher” store for Computer
3. Registers VEN Event (Event Tracing for Windows, or ETW) providers
4. Installs Ilowfp (kernel mode driver) and the processes and services described in [VEN Architectural Diagram](#) as auto-start at boot and then starts them.

VEN-to-PCE Communications

For background, see the diagram and details in [Overview of VEN Software Architecture and Description of Components](#).

The VEN communicates with the PCE on (8)443 using HTTPS. The VEN uses Transport Level Security (TLS) to connect to the PCE. The PCE certificate must be trusted by the VEN before communication can be set up.

The VEN communicates with PCE and sends following details:

1. Traffic logs
2. Network interfaces
3. Processes
4. Open ports
5. The PCE communicates with VEN and sends following details
6. Firewall policy
7. Lightning bolts with action to perform (such as send support report)

Polling Intervals of VEN Operations

The following table shows the intervals of common VEN operations. The [PCE Operations Guide](#) includes more details about these functions and their effects.

Function	Polling Interval	Note
Firewall policy updates	Real-time if lightning bolts are enabled	
Active service reporting	Every 30 seconds	

Function	Polling Interval	Note
Interface reports and changes	Every 5 minutes	
Firewall and traffic flow Log	Every 10 minutes	
Heartbeat	Every 5 minutes	If the PCE does not receive three consecutive heartbeats, an event is written to the PCE's event log.
Dead-peer interval	60 minutes = 12 heartbeats	
VEN tampering detection	10 minutes	

Heartbeat mechanism

The VEN sends a heartbeat message every five minutes to the PCE to inform the PCE that it is up and running. If the VEN cannot connect to the PCE (either because the PCE is down or because of a network issue), the VEN continues to enforce the last-known-good policy while it tries to reconnect with the PCE.

After missing two heartbeats, the VEN enters a diminished state. In the diminished state, the VEN ignores all the asynchronous commands received as lightning bolts from the PCE, except the commands for software upgrade and support reports. After the connectivity to the PCE is restored, the VEN comes out of the diminished state after two successful heartbeats.

If the VEN fails to communicate with the PCE because of failed authentication, the VEN enters a state called *lost agent*. In the lost agent state, the VEN only attempts to connect with the PCE every four hours. The PCE logs a message in the Organization Events to inform the user that the VEN needs to be uninstalled or reinstalled manually on this Workload. If the authentication failure was temporary, after first successful connection to the PCE, the VEN exits the lost agent state .

Details about Isolation Mechanism

When the VEN on a workload is stopped, the PCE detects that the workload is offline. The PCE recomputes the policy for all the peer workloads. In the new policy, the peer workloads are not allowed to communicate with the workload where the VEN is stopped.

If the workload goes offline abruptly (for example, due to a power outage), the PCE stops receiving heartbeats from the workload. After one hour, the PCE marks the workload as offline and recomputes policies for the peer workloads to isolate the offline workload.

SecureConnect

The VEN uses the StrongSwan suite to provide IPsec encryption between the host and a communicating Workload. StrongSwan is installed as part of the VEN installation. StrongSwan is used to perform the Internet Key Exchange (IKE) v2 handshake. The actual encryption of IP packets is done natively by the OS.

Sampling Mode

If the VEN receives a sustained amount of high traffic per second from many individual connections, the VEN enters Sampling Mode to reduce load. Sampling Mode is a protection mechanism to ensure that the VEN does not contribute to the consumption of CPU. In Sampling Mode, not every flow is reported. Instead, flows are periodically sampled and logged.

After CPU usage on the VEN decreases, Sampling Mode is disabled and each connection is reported to the VEN. The entry and exit from sampling-mode is automatically performed by VEN depending on the load on VEN.

Linux `nf_conntrack_tcp_timeout_established` set to 8 Hours

For VENs installed on Linux workloads, Illumio uses `conntrack` to manage the `nf_conntrack_tcp_timeout_established` variable.

By default, as soon as the VEN is installed, it sets the `nf_conntrack_tcp_timeout_established` value to 8 hours (28,800 seconds). This frequency is to manage workload memory by removing unused connections from the table and thereby increase performance.

If you change this setting via `sysctl`, it is reverted the next time the workload is rebooted or the next time the VEN's configuration file is read.

VEN Uninstallation

During uninstallation, the VEN performs the following steps.

Linux	Windows
<ul style="list-style-type: none"> • Unpairs from the PCE • Restores the host firewall state to the requested or open state if no state is specified. Possible values of the state are: <ul style="list-style-type: none"> • Open: All ports are open after VEN uninstalls • Saved: Restore the firewall to its state just before the VEN was installed • Uninstalls the illumio-agent-control and illumio-agent-vtap packages <ul style="list-style-type: none"> • Removes program and data files • Removes Illumio repo and gpg files and packages 	<ul style="list-style-type: none"> • Unpairs the VEN from the PCE • Sends a “deactivate” message to PCE • Stops all VEN services • Unregisters services from Service Control Manager • Restores Windows Firewall to requested state <ul style="list-style-type: none"> • Open: All ports are open after VEN uninstalls • Saved: Restore the firewall to its state just before the VEN was installed • Removes Program Files and ProgramData directories • Removes VEN registry keys • Removes Certificate • Unregisters VEN Event provider

VEN Activation or Pairing

The terms *activation/deactivation* of a *VEN* applies to the single-package installation directly on the workload, but the term *pairing/unpairing* a *VEN* is used for the Illumio Repo model of deployment and also in the PCE Console UI. These two terms indicate the same function.

A workload pairs with a PCE before it can become part of illumio distributed security system. Pairing can be performed using one of these methods:

- A pairing key
- A PKI certificate
- A Kerberos service principal name (SPN)

An activation key or pairing key is used only at initial pairing. During pairing, an Agent Token is generated and stored in a local file on workload, and the hash of the token is stored on PCE. Only the agent-token is used in VEN-to-PCE communication from that point onwards.

The VEN communicates with PCE with HTTPS over Transport Layer Security (TLS). The Agent Token is used by VEN to uniquely authenticate itself to PCE. In addition, a Clone Token is generated by the VEN. If an Agent Token

is mistakenly or maliciously reused on another workload, the Clone Token is used to detect the condition and disambiguate the hosts. The Clone Token is periodically rotated. Agent Token is never rotated.

VEN Startup

See the commands [VEN Startup](#).

For a description of the VEN architecture and software components, see [Overview of VEN Software Architecture and Description of Components](#).

VEN Shutdown

See the commands in [VEN Shutdown](#).

For a description of the VEN architecture and the software components, see [Overview of VEN Software Architecture and Description of Components](#).

VEN Status

The VEN status contains information related to the current state of VEN connectivity, the most recently provisioned policy changes that affect the workload, any potential firewall tampering, and any issues related to SecureConnect functionality.

See the commands in [VEN Status](#).

Workload Policy States

After activation, the VEN can be in one of the following policy states. The VEN policy state determines how the rules received from PCE affect a Workload's network communication.

You change the policy state of the VEN via settings in the PCE or the REST API.

1. **Build:** The VEN inspects all open ports on a Workload and reports to the PCE the flow of traffic between it and other Workloads. In this state, the PCE displays the flow of traffic to and from the Workload, providing insight into the data center and the applications running in it. No traffic is blocked in this state. The Build state is useful when firewall policies are not yet known. This state can be used for discovering the Application flows in the organization and then generating a security policy that governs all desired communication.
2. **Enforced:** All Ruleset Rules are enforced on the workload. Any traffic flows not explicitly allowed by the Rules from the PCE are blocked.

3. **Idle:** The VEN does not take control of the Workload's iptables (Linux) or firewall (Windows), but uses Workload network analysis to provide to the PCE relevant details about the Workload, such as the Workload's IP address, operating system, and traffic flows. This snapshot is taken every four hours. Idle state is used for installing and activating VENs on Workloads without changing the iptables/ipsets on the Workloads. A pairing profile can be used to pair Workloads in Idle state.
4. **Test:** No traffic is blocked in this state. In Test, you can visualize all of the traffic that would be blocked if you enforced Ruleset Rules on the Workloads.

Automatic History of Firewall Changes

Changes to the firewall on a workload are historically recorded for audit trail. Up to 10 changes to the firewall history are saved. The history is viewable via the PCE Support Reports. For more details on Support Reports, see the PCE Operations guide and PCE Web Console guide in [Related Documentation](#).

VEN Traffic Logging

The VEN captures logs of its operation and traffic flow summaries locally on the workload. There are several different application log files, each with one backup.

Application logs are rotated from primary to backup when their size reaches 15 MB. Application log files are limited to 390MB total for all logs. Application log files are preserved at reboot, because application logs are stored in files on a workload

Contents of Traffic Flow Logs

The VEN stores traffic flow summaries, rather than each individual traffic flow. For each connection, the traffic flow summary includes:

- Source IP
- Destination IP
- Destination Port
- Protocol
- Number of connections

Querying flow log databases

The `sqlite` command-line tool comes with the VEN, which you can use to query the flow log databases.

Linux Database Query Examples

- Non-aggregated accepted flows
`/opt/illumio/bin/sqlite /opt/illumio/log/flow.db "select * from flow_view"`
- Non-aggregated dropped flows
`/opt/illumio/bin/sqlite /opt/illumio/log/flow.db "select * from drop_flow_view"`
- Aggregated accepted flows
`/opt/illumio/bin/sqlite /opt/illumio/log/flowsum.db "select * from flow_view"`
- Aggregated dropped flows
`/opt/illumio/bin/sqlite /opt/illumio/log/flowsum.db "select * from drop_flow_view"`

Window Database Query Examples

- Non-aggregated accepted flows
`"c:\Program Files\Illumio\bin\sqlite.exe" c:\ProgramData\Illumio\log\flow.db "select * from flow_view"`
- Non-aggregated dropped flows
`"c:\Program Files\Illumio\bin\sqlite.exe" c:\ProgramData\Illumio\log\flow.db "select * from drop_flow_view"`
- Aggregated accepted flows
`"c:\Program Files\Illumio\bin\sqlite.exe" c:\ProgramData\Illumio\log\flowsum.db "select * from flow_view"`
- Aggregated dropped flows
`"c:\Program Files\Illumio\bin\sqlite.exe" c:\ProgramData\Illumio\log\flowsum.db "select * from drop_flow_view"`

Summary of VEN and Useful OS commands

Below is a short description of the VEN command-line tools you commonly use for various operations and some useful native OS commands.

Syntax for the Illumio-provided commands is detailed throughout this guide, in the [VEN Deployment guide](#), and in the help of the commands themselves.

Platform	Command	Description
Linux	illumio-firewall	Illumio shell script to start the policy-based firewall at boot.
	illumio-ven-ctl	Illumio shell script to control VEN control VEN settings and functions
	agent_status	Alternative to <code>illumio-ven-ctl status</code>
	pair	Script to pair with the PCE
	ps	Native OS command to list all system processes
	chkconfig	Native OS command to update and query runlevel information for system services
Windows	illumio-ven-ctl.ps1	Illumio PowerShell script to control VEN settings and functions
	pair.ps1	Illumio PowerShell script to pair with the PCE
	Get-Service	Native OS command to display system services
	tasklist /svc	Native OS command to display system services
	wf.msc	Native OS command to manage the Windows firewall

Revision History: Illumio ASP VEN Operations

Date	Description
2018-05-10	<ul style="list-style-type: none"> • Updated for Illumio ASP version 18.1 • Addition of theory of VEN operations • Details of upgrade VENs on PCE Virtual Appliance now moved to PCE Virtual Appliance Guide • Reorganization and miscellaneous corrections throughout; removal of section numbering. • Addition of glossary of common terminology • Start of revision history