

IPsec Add-On for PCF®

Documentation

Version 1.8

Published: 5 Feb 2019

Table of Contents

Table of Contents	2
IPsec Add-on for PCF	3
Troubleshooting the IPsec Add-on for PCF	4
Release Notes	10
Installing the IPsec Add-on for PCF	12
Upgrading the IPsec Add-on for PCF	22
Uninstalling the IPsec Add-on for PCF	24
Checking Certificate Dates	25
Rotating Active IPsec Certificates	27
Renewing Expired IPsec Certificates	29

IPsec Add-on for PCF

Page last updated:

This guide describes the IPsec Add-on for PCF, which secures data transmissions inside [Pivotal Cloud Foundry](#) (PCF). Topics covered in this guide include IPsec Add-on for PCF installation and configuration, troubleshooting, and certificate rotation.

Your organization may require IPsec if you transmit sensitive data.

Overview

The IPsec Add-on for PCF provides security to the network layer of the OSI model with a [strongSwan](#) implementation of IPsec. The IPsec Add-on provides a strongSwan job in FIPS mode to each BOSH-deployed VM.

IPsec encrypts IP data flow between hosts, between security gateways, and between security gateways and hosts. The IPsec Add-on for PCF secures network traffic within a Cloud Foundry deployment and provides internal system protection if a malicious actor breaches your firewall.

Product Snapshot

The following table provides version and version-support information about the IPsec Add-on for PCF.

Element	Details
Version	v1.8.31
Release date	April 27, 2018
Compatible Ops Manager version(s)	v1.10.x, v1.11.x, v1.12.x, v2.0.x, and v2.1.x
Compatible Elastic Runtime version(s)	v1.10.x, v1.11.x, and v1.12.x
Compatible Pivotal Application Service (PAS)* version(s)	v2.0.x and 2.1.x
IaaS support	vSphere, GCP, AWS, Azure, and Openstack

* As of PCF v2.0, *Elastic Runtime* is renamed *Pivotal Application Service (PAS)*.

IPsec Implementation Details

The IPsec Add-on for PCF implements the following cryptographic suite:

Key Agreement (Diffie-Hellman)	IKEv2 Main Mode
Bulk Encryption	AES128GCM16
Hashing	SHA2 256
Integrity/Authentication Tag	128 bit GHASH ICV
Digital Signing	RSA 3072/4096
Peer Authentication Method	Public/Private Key

Limitation

IPsec Add-on for PCF has the following limitations:

- Due to a [known issue](#) in Windows Server OS, apps hosted on PAS for Windows cannot route traffic when deployed with the IPsec add-on for PCF.
- Pivotal recommends configuring IPsec to use a self-signed certificate to sign instance certs. Using a certificate signed by a public or third-party CA is not recommended.

Troubleshooting the IPsec Add-on for PCF

Page last updated:

This topic provides instructions to verify that strongSwan-based IPsec works with your Pivotal Cloud Foundry (PCF) deployment and general recommendations for troubleshooting IPsec.

Verify that IPsec Works with PCF

To verify that IPsec works between two hosts, you can check that traffic is encrypted in the deployment with `tcpdump`, perform the ping test, and check the logs with the steps below.

1. Check traffic encryption and perform the ping test. Select two hosts in your deployment with IPsec enabled and note their IP addresses. These are referenced below as `IP-ADDRESS-1` and `IP-ADDRESS-2`.

- a. SSH into `IP-ADDRESS-1`.

```
$ ssh IP-ADDRESS-1
```

- b. On the first host, run the following, and allow it to continue running.

```
$ tcpdump host IP-ADDRESS-2
```

- c. From a separate command line, run the following:

```
$ ssh IP-ADDRESS-2
```

- d. On the second host, run the following:

```
$ ping IP-ADDRESS-1
```

- e. Verify that the packet type is ESP. If the traffic shows `ESP` as the packet type, traffic is successfully encrypted. The output from `tcpdump` will look similar to the following:

```
03:01:15.242731 IP IP-ADDRESS-2 > IP-ADDRESS-1: ESP(spi=0xcfdbb261,seq=0x3), length 100
```

2. Open the `/var/log/daemon.log` file to obtain a detailed report, including information pertaining to the type of certificates you use, and to verify an established connection exists.
3. Navigate to your Installation Dashboard, and click **Recent Install Logs** to view information regarding your most recent deployment. Search for “ipsec” and the status of the IPsec job.
4. Run `ipsec statusall` to return a detailed status report regarding your connections. The typical path for this binary: `/var/vcap/packages/strongswan-x.x.x/sbin`. `x.x.x` represents the version of strongSwan packaged into the IPsec.

If you experience symptoms that IPsec does not establish a secure connection, return to the [Installing the IPsec Add-on for PCF](#) topic and review your installation.

If you encounter issues with installing IPsec, refer to the [Troubleshooting IPsec](#) section of this topic.

Troubleshoot IPsec

IPsec Installation Issues

Symptom

Unresponsive apps or incomplete responses, particularly for large payloads

Explanation: Packet Loss

IPsec packet encryption increases the size of packet payloads on host VMs. If the size of the larger packets exceeds the maximum transmission unit (MTU) size of the host VM, packet loss may occur when the VM forwards those packets.

If your VMs were created with an Amazon PV stemcell, the default MTU value is 1500 for both host VMs and the application containers. If your VMs were created with Amazon HVM stemcells, the default MTU value is 9001. Garden containers default to 1500 MTU.

Solution

Implement a 100 MTU difference between host VM and the contained application container, using one of the following approaches:

- Decrease the MTU of the application containers to a value lower than the MTU of the VM for that container. In the Elastic Runtime tile configuration, click **Networking** and modify **Applications Network Maximum Transmission Unit (MTU) (in bytes)** before you deploy. Decrease it from the default value of 1454 to 1354.
- Increase the MTU of the application container VMs to a value greater than 1500. Pivotal recommends a headroom of 100. Run `ifconfig NETWORK-INTERFACE mtu MTU-VALUE` to make this change. Replace NETWORK-INTERFACE with the network interface used to communicate with other VMs For example: `$ ifconfig NETWORK-INTERFACE mtu 1600`

Symptom

Unresponsive apps or incomplete responses, particularly for large payloads

Explanation: Network Degradation

IPsec data encryption increases the size of packet payloads. If the number of requests and the size of your files are large, the network may degrade.

Solution

Scale your deployment by allocating more processing power to your VM CPU or GPUs, which, additionally, decreases the packet encryption time. One way to increase network performance is to compress the data prior to encryption. This approach increases performance by reducing the amount of data transferred.

IPsec Runtime Issues

Symptom

Errors relating to IPsec, including symptoms of network partition. You may receive an error indicating that IPsec has stopped working.

For example, this error shows a symptom of IPsec failure, a failed `clock_global-partition` :

```
Failed updating job clock_global-partition-abf4378108ba40fd9a43 > clock_global-partition-abf4378108ba40fd9a43/0
(ddb1fbfa-71b1-4114-a82c-fd75867d54fc)
(canary): Action Failed
get_task: Task 044424f7-c5f2-4382-5d81-57bacefbc238
result: Stopping Monitored Services: Stopping service
ipsec: Sending stop request to Monit: Request failed,
response: Response{ StatusCode: 503, Status: '503 Service Unavailable' } (00:05:22)..
```


Explanation: Asynchronous `monit` Job Priorities

When a monit stop command is issued to the NFS mounter job, it hangs, preventing a shutdown of the PCF cluster.

This is not a problem with the IPsec add-on release itself. Rather, it is a known issue with the NFS mounter job and the monit stop script that can manifest itself after IPsec is deployed with PCF v1.7.

This issue occurs when monit job priorities are asynchronous. Because the order of job shutdown is arbitrary, it is possible that the IPsec job will be

stopped first. After this happens, the network connectivity for that VM goes away, and the NFS mounter job loses visibility to the associated storage. This causes the NFS mounter job to hang, and it blocks the monit stop from completing. See the [Monit job Github details](#) for further information.

 **Note:** This issue affects deployments using CF v231 or earlier, but in CF v232 the release uses an nginx blobstore instead of the NFS blobstore. The error does not exist for PCF deployments using CF releases greater than CF v231. The error also does not apply to PCF deployments that use WebDAV as their Cloud Controller blobstore.

Solution

1. BOSH `ssh` into the stuck instance by running one of the following commands:
 - **For Ops Manager v1.10 or earlier:**
`bosh ssh VM-INDEX`
 - **For Ops Manager v1.11 or later:** `bosh2 -e BOSH-ENVIRONMENT -d DEPLOYMENT-NAME ssh VM-INDEX`

2. Authenticate as root and use the `sv stop agent` command to kill the BOSH Agent:

```
$ sudo su
# sv stop agent
```

3. Run the following command to detect the missing monit job VM.
 - **For Ops Manager v1.10 or earlier:**
`bosh cloudcheck`
 - **For Ops Manager v1.11 or later:** `bosh2 -e ENVIRONMENT-NAME -d DEPLOYMENT-NAME cloud-check`

For example,

```
# bosh cloudcheck
VM with cloud ID `vm-3e37133c-bc33-450e-98b1-f86d5b63502a' missing:

- Ignore problem
- Recreate VM using last known apply spec
- Delete VM reference (DANGEROUS!)
```

4. Choose `Recreate VM using last known apply spec`.
5. Continue with your deploy procedure.

Symptom

- App fails to start with the following message:

```
FAILED
Server error,
status code: 500,
error code: 10001,
message: An unknown error occurred.
```

The Cloud Controller log shows it is unable to communicate with Diego due to `getaddrinfo` failing.

- Deployment fails with a similar error message: `diego_database-partition-620982d595434269a96a/0 (a643c6c0-bc43-411b-b011-58f49fb61a6f)' is not running after update. Review logs for failed jobs: etcd`

Explanation: Split Brain `consul`

This error indicates a “split brain” issue with Consul.

Solution

Confirm this diagnosis by checking the `peers.json` file from `/var/vcap/store/consul_agent/raft`. If it is null, then there may be a split brain. To fix this problem, follow these steps:

1. Run `monit stop` on all Consul servers:
2. Run `rm -rf /var/vcap/store/consul_agent/` on all Consul servers.
3. Run `monit start consul_agent` on all Consul servers one at a time.
4. Restart the consul_agent process on the Cloud Controller VM. You may need to restart consul_agent on other VMs, as well.

Symptom

You see that communication is not encrypted between two VMs.

Explanation: Error in Network Configuration


The IPsec BOSH job is not running on either VM. This problem could happen if both IPsec jobs crash, both IPsec jobs fail to start, or the subnet configuration is incorrect. There is a momentary gap between the time when an instance is created and when BOSH sets up IPsec. During this time, data can be sent unencrypted. This length of time depends on the instance type, IAAS, and other factors. For example, on a t2.micro on AWS, the time from networking start to IPsec connection was measured at 95.45 seconds.

Solution

Set up a networking restriction on host VMs to only allow IPsec protocol and block the normal TCP/UDP traffic. For example, in AWS, configure a network security group with the minimal networking setting as shown below and block all other TCP and UDP ports.

Additional AWS Configuration

Type	Protocol	Port Range	Source
Custom Protocol	AH (51)	All	10.0.0.0/16
Custom Protocol	ESP (50)	All	10.0.0.0/16
Custom UDP Rule	UDP	500	10.0.0.0/16

 **Note:** When configuring a network security group, IPsec adds an additional layer to the original communication protocol. If a certain connection is targeting a port number, for example port 8080 with TCP, it actually uses IP protocol 50/51 instead. Due to this detail, traffic targeted at a blocked port may be able to go through.

Symptom

You see unencrypted app messages in the logs.

Explanation: `etcd` Split Brain

Solution

1. Check for split brain etcd by connecting with BOSH `ssh` into each etcd node:

```
$ curl localhost:4001/v2/members
```

2. Check if the members are consistent on all of etcd. If a node has only itself as a member, it has formed its own cluster and developed "split brain." To fix this issue, SSH into the split brain VM and run the following commands:

a. `$ sudo su -`

b. `# monit stop etcd`

c. `# rm -r /var/vcap/store/etcd`

d. `# monit start etcd`

3. Check the logs to confirm the node rejoined the existing cluster.

Symptom

IPsec deployment fails with `Error filling in template 'pre-start.erb'`

```
Error 100: Unable to render instance groups for deployment. Errors are:
- Unable to render jobs for instance group 'consul_server-partition-f9c4b18fd83cf3114d7f'. Errors are:
  - Unable to render templates for job 'ipsec'. Errors are:
    - Error filling in template 'pre-start.erb' (line 12: undefined method 'each_with_index' for #)
- Unable to render jobs for instance group 'nats-partition-f9c4b18fd83cf3114d7f'. Errors are:
  - Unable to render templates for job 'ipsec'. Errors are:
    - Error filling in template 'pre-start.erb' (line 12: undefined method 'each_with_index' for #)
```

Explanation: Typographical or syntax error in deployment descriptor YAML syntax

Solution

Check the deployment descriptor YAML syntax for the CA certificates entry:

```
releases:
- {name: ipsec, version: 1.0.0}

addons:
- name: ipsec-addon
jobs:
- name: ipsec
  release: ipsec
properties:
  ipsec:
    ipsec_subnets:
    - 10.0.1.1/20
    no_ipsec_subnets:
    - 10.0.1.10/32 # bosh director
  instance_certificate: |
    -----BEGIN CERTIFICATE-----
    MIIEMDCCAhigAwIBAgIRAIvrbY2TitU/LcRhO+V1t0YwDQYJKoZIhvcNAQELBQAw
    ...
    -----END CERTIFICATE-----
  instance_private_key: |
    -----BEGIN EXAMPLE RSA PRIVATE KEY-----
    MIIEEgIBAAKCAQEAtAkBjr5x9g0aWgyDEmLd7m9u/ZzpK7UScfANLaN7JiNz3c
    ...
    -----END EXAMPLE RSA PRIVATE KEY-----
  ca_certificates:
  - |
    -----BEGIN CERTIFICATE-----
    MIIEUDCCAriGAWIBAgIJAJVLBeJ9Wm3TMA0GCSqGSIb3DQEBCwUAMB0xGzAZBgNV
    BAMMEIBDRiBJUHNiYyBBZGRPbiBDQTAeFw0xNjA4MTUxNzQwNDVaFw0xOTA4MTUx
    ...
    -----END CERTIFICATE-----
```

In the example above, the values that appear after the `ca_certificates` key are contained within a list and are not just a single certificate. This entry must be followed by a line starting with `- |`, and ending with `|`. The lines following this contain the PEM encoded certificate(s).

The error message shown above indicating a problem with the `each_with_index` method provides a hint that the `- |` YAML syntax sequence is missing. Use this syntax even in situations where there is only one CA certificate, for example a list of one entry.

Symptom

Complete system outage with no warning.

Explanation: IPsec Certificates Might Have Expired

Expired IPsec certificates can cause a sudden system outage. For example, the self-signed certificates generated by the script provided in the installation instructions have a lifetime of 365 days. IPsec certificates expire if you do not rotate them within their lifetime.

Solution

Renew expired IPsec certificates. To avoid future downtime due to expired IPsec certificates, set a calendar reminder to rotate the certificates before they expire.

For how to renew certificates, see [Renewing Expired IPsec Certificates](#). For how to rotate them, see [Rotating IPsec Certificates](#).

Symptom

BOSH shows a VM in a failing state. On the failing VM, `monit summary` shows `ipsec` with a status of `Does not exist`.

Explanation: IPsec stopped/crashed and Monit Cannot Automatically Bring it Back Up.

IPsec is required to be the first process to start and the last process to stop. As a result, the start and stop scripts are located in `pre-start` and `post-stop`, which is a concept to BOSH, but not to Monit. Monit is not able to bring IPsec back up automatically because it does not know what `pre-start` is. After you run the `pre-start` manually, then Monit is able to detect IPsec as healthy.

Solution

1. From the failing VM, run the following command as root: `/var/vcap/jobs/ipsec/bin/pre-start`
2. Run `monit summary`. If the restart is successful, Monit shows `ipsec` with a status of `Running`.

Release Notes

Page last updated:

This topic contains release notes for the IPsec Add-on for PCF.

v1.8.31

Release Date: April 27, 2018

Features included in this release:

- Options to configure syslog messages that warn about upcoming IPsec certificates expiry.

Fixed issues in this release:

- When the host VM has more than one network interface available, the `leftsubnet` is always configured with the internal IP address. Previously, the `leftsubnet` was configured with the first available interface, which may or may not be the internal address.

v1.8.14

Release Date: January 24, 2018

Features included in this release:

- A new manifest property named `dpdaction` has been added. This property controls the IPsec response upon detecting a “dead” network peer. The default value is `restart`.

Known issues in this release:

- When using IKEv1, Pivotal recommends that you set the manifest property `dpdaction` to `none`.

v1.8.12

Release Date: January 19, 2018

Features included in this release:

- An arbitrary length certificate chain is now supported for both Linux and Windows.
- A warning message is generated in the IPsec stdout log file when the `optional` flag is set to true.
- A new manifest property has been added, `optional_warn_interval`, to control the message interval for the optional-is-true warning.
- The default `log_level` is now set to -1 for Linux.
- The golang dependency has been updated to v1.9.2.
- An error is reported if the host IP address is in neither the IPsec nor the no-IPsec subnet.
- If, upon shutdown, the IPsec job is not the last monit job running, a warning is logged.
- A new manifest property has been added, `ike_version`. Accepted values are `ikev1` or `ike` (for IKEv2).
- Additional certificate verification has been added so that an error is reported if the supplied instance certificate and CA certificate are not related.

Known issues in this release:

- Pivotal does not recommend using IKEv1 because of security and performance limitations.
- Only use IKEv1 for deployments that require it: mixed environments containing both Linux and Windows VMs.

v1.8.3

Release Date: October 11, 2017

Features included in this release:

- Updates strongSwan to 5.6.0
- Updates OpenSSL to 1.0.2l
- Updates OpenSSL FIPS to 2.0.16
- Log level is now configurable.
- Key exchange is now configurable.
- Instance certificate is validated with the CA certificate on start.

Fixed issues in this release:

- Stop script timeout is configurable.

Known issues in this release:

- **IKEv1 on Windows:** Windows uses IKEv1 for Key exchange. IKEv2 does not support multiple root certificates, and therefore does not support certificate rotation. An issue has been filed with Microsoft.
- **Spurious Configuration Warning:** As part of the upgrade to StrongSwan v5.4.0, this version of the IPsec add-on may emit a sequence of spurious configuration warning messages. The messages are similar to the following:

```
!! Your strongswan.conf contains manual plugin load options for charon.  
!! This is recommended for experts only, see  
!! http://wiki.strongswan.org/projects/strongswan/wiki/PluginLoad
```

These messages are both expected and harmless. As a caution to end users, the StrongSwan software now emits a warning message when it detects that the installation includes a manually configured set of plugins. As a matter of security hygiene best practices, the IPsec add-on has always used a manual (explicit) configuration and loads a restricted set of StrongSwan plugins. Any unused plugins are not loaded. The newest version of StrongSwan now issues this warning message when it detects that situation. The actual list of plugins in use has been determined to be appropriate for use of StrongSwan in the PCF environment. This warning is expected and should be ignored.

- **MTU Sizing:** Use 1354 on OpenStack. Keep the default on AWS and vSphere.

Installing the IPsec Add-on for PCF

Page last updated:

This topic describes how to prepare your network for IPsec, create an IPsec runtime config, and add IPsec to your deployment.

Prerequisites

To complete the IPsec installation, verify that you have satisfied the following prerequisites before you begin:

- Google Cloud Platform (GCP), vSphere, Azure, Amazon Web Services (AWS), or OpenStack as your IaaS
- Pivotal Cloud Foundry (PCF) operator administration rights
- BOSH deployed through Ops Manager v1.8 or later
- Set the MTU for your IaaS in the Pivotal Application Service (PAS) or Elastic Runtime tile, under **Networking**. Pivotal recommends MTU values of 1354 on GCP, 1438 on Azure, and the default values on AWS and vSphere. For OpenStack, follow the recommendations of your [Neutron/ML2](#) plugin provider, or empirically test the correct MTU for your environment.

Best Practices

- IPsec may affect the functionality of other service tiles. As a result, Pivotal recommends deploying PAS (or Elastic Runtime) and each service tile to different isolated subnets. Alternatively, you can minimally deploy all service tiles to a single isolated subnet, apart from the PAS (or Elastic Runtime) subnet. Some service tiles do not support IPsec and must be placed in a non-IPsec subnet.
- For IPsec on Linux VMs, Pivotal recommends any Ubuntu stemcells for vSphere, OpenStack, and HVM stemcells for AWS. These stemcells are available on [Pivotal Network](#). If you use PV stemcells obtained from [bosh.io](#), see the [Packet Loss](#) section of the Troubleshooting the IPsec Add-on for PCF topic to adjust MTU values.
- For IPsec on Windows VMs, Pivotal recommends the Windows 2012R2 stemcells for AWS, GCP, or Azure available on [Pivotal Network](#).

Step 1: Configure Network Security

Perform the steps in the appropriate section below to configure your IaaS network security.

Google Cloud Platform

To configure your Google Cloud Platform (GCP) environment for IPsec, perform the following steps:

1. Navigate to the **Networking** section of the GCP Console.
2. Click **Firewall rules**.
3. Click **Create Firewall Rule**.
4. For **Name**, enter `ipsec`.
5. For **Network**, select the network where Ops Manager is deployed. For example, `opsmgr`.
6. For **Source filter**, select **Allow from any source (0.0.0.0/0)**.
7. For **Allowed protocols and ports**, enter `udp:500; ah; esp`.
8. Click **Create**.
9. Adjust the MTU value to `1354` by performing the procedure in the [Packet Loss](#) section of the Troubleshooting the IPsec Add-on for PCF topic.

vSphere

Confirm that your network allows the protocols listed in the table below.

Protocol Name	Protocol Number	Port(s)
AH	51	Any
ESP	50	Any
UDP	17	500

Azure

1. Confirm that your network allows the protocols listed in the table below.

Protocol Name	Protocol Number	Port(s)
AH	51	Any
ESP	50	Any
UDP	17	500


2. Adjust the MTU value to `1438`. For instructions, see [Explanation: Packet Loss](#).

AWS


To configure your AWS environment for IPsec, perform the following steps:

1. Navigate to **EC2 Dashboard > Security Groups**.
2. Select the Security Group with the description **PCF VMs Security Group** and click **Edit**.
3. Create the following **Inbound Rules**.

Type	Protocol Name	Protocol Number	Port Range	Source
Custom Protocol	AH	51	All	10.0.0.0/16
Custom Protocol	ESP	50	All	10.0.0.0/16
Custom UDP Rule	UDP	17	500	10.0.0.0/16

 **Note:** The default **PCF VMs Security Group** is typically specified with a subnet of `10.0.0.0/16`. If your PCF subnet is deployed to a different CIDR block, adjust the source as needed.

OpenStack

 **Note:** The following network configuration is optimized for Mirantis OpenStack, but other OpenStack distributions have a similar workflow.

To configure your Mirantis OpenStack environment for IPsec, perform the following steps:

1. Navigate to **Project / Access & Security**.
2. Select the security group and click **Manage Rules**.
3. Create the following **Ingress and Egress Rules**. Adjust the source CIDR as needed for your environment.

Protocol Name	Protocol Number	Port Range	Source
ESP	50	Any	0.0.0.0/0
AH	51	Any	0.0.0.0/0
UDP	17	500	0.0.0.0/0


Step 2: Create the IPsec Manifest

To add IPsec to VMs in your deployment, you must create a runtime config file named `ipsec-addon.yml` that configures IPsec add-on properties for Linux VMs, Windows VMs, or both. Perform the following steps:

1. Create an IPsec runtime config file `ipsec-addon.yml`, with the code below as a template.

```
releases:
  \- name: ipsec
     version: 1.X.X
addons:
```

1. Add properties to the `ipsec-addon.yml` file as described below for [Linux VMs](#) and [Windows VMs](#).

 **Note:** Enabling IPsec for Windows adds IPsec security to Windows VMs that users can create after installing the [PAS for Windows 2012R2](#) tile.

Add Linux VM Support to Your Runtime Config

Perform the following steps to add IPsec to Linux VMs in your deployment:

1. Add the following YAML under `addons:` to your `ipsec-addon.yml` file:

```
releases:
- name: ipsec
  version: 1.X.X
addons:
- name: ipsec-addon
  jobs:
  - name: ipsec
    release: ipsec
  include:
    stemcell:
    - os: ubuntu-trusty
  properties:
    ipsec:
      optional: false
      ipsec_subnets:
      - 10.0.1.1/20
      no_ipsec_subnets:
      - 10.0.1.10/32 # bosh director
      - 10.0.1.4/32 # ops manager
      instance_certificate: |
        -----BEGIN CERTIFICATE-----
        MIIEMDCCAhiGAWIBAgIRAIvrbY2TttU/LeRhO+V1t0YwDQYJKoZIhvcNAQELBQAw
        ...
        -----END CERTIFICATE-----
      instance_private_key: |
        -----BEGIN EXAMPLE RSA PRIVATE KEY-----
        EXAMPLExRSAXPRIVATExKEYxDATAxEXAMPLExRSAXPRIVATExKEYxDATA
        ...
        -----END EXAMPLE RSA PRIVATE KEY-----
      ca_certificates:
      - |
        -----BEGIN CERTIFICATE-----
        MIIFCTCCAvGgAwIBAgIBATANBgkqhkiG9w0BAQsFADAUMRIwEAYDVQQDEw10ZXN0
        ...
        -----END CERTIFICATE-----
      - |
        -----BEGIN CERTIFICATE-----
        MIIFCTCCAvGgAwIBAgIBATAAYDVQQDEw10ZXN0NBgkqhkiG9w0BAQsFADAUMRIwE
        ...
        -----END CERTIFICATE-----
      prestart_timeout: 30
      esp_proposals: aes128gcm16!
      ike_proposals: aes128-sha256-modp2048!
      log_level: 1
      ike_version: ike
      optional_warn_interval: 1
      force_udp_encapsulation: false
      instance_certificate_info_period: 30
      instance_certificate_warn_period: 14
      instance_certificate_error_period: 7
      instance_certificate_critical_period: 3
```

2. Replace the values listed in the template as follows:

- `releases: - version` : Specify the version number of your IPsec download from Pivotal Network.
- `optional` : This value makes IPsec enforcement optional. To add IPsec to an existing PAS (or Elastic Runtime) deployment, set this flag to `true` . After IPsec has been successfully installed, set this flag back to `false` and redeploy.

⚠ WARNING: Communication between existing components fails if you try to add IPsec to an existing deployment without setting `optional` to `true` .

- `ipsec_subnets` : List the subnets that you want to be encrypted. You can include the entire deployment or a portion of the network. Encrypt any network that handles business-sensitive data.
- `no_ipsec_subnets` : List the IP address of your BOSH Director and Ops Manager VM, along with any other IP addresses in your PCF deployment that you want to communicate with without encryption. Pivotal recommends that you list the subnets that are used for PCF managed services. Subnets for PCF managed services that do not support IPsec (such as an Pivotal Ops Manager) must be listed under `no_ipsec_subnets` .

💡 Note: If you have an external load balancer such as F5, add it to the `no_ipsec_subnets` property. If you want to include it in the `ipsec_subnet` , you must configure it manually.

⚠ WARNING: IPs that are not in `ipsec_subnets` or `no_ipsec_subnets` have no default behavior and cannot communicate with other internal VMs. You must specify internal IPs in `ipsec_subnets` or `no_ipsec_subnets` .

⚠ WARNING: In GCP, if you use the default router for DNS instead of the Google public DNS at `8.8.8.8` , you must add the IP address of the default router in your subnet to `no_ipsec_subnets` . For example, `10.0.0.1/32` .

- `instance_certificate` : Copy in the signed certificate that will be used by all your instance VMs. You must use one of the CAs in the `ca_certificates` property to sign this certificate. Pivotal recommends that you use a self-signed certificate. For more information, see [Generate a Self-Signed Certificate](#).
- `instance_private_key` : Copy in the private key that corresponds to the `instance_certificate` above. This key must not use a pass phrase.
- `ca_certificates` : Copy in CA certificates for the instance VM to trust during the validation process. In most cases, you only need the CA certificate used to sign the instance certificate. During CA credential rotation, you need two CA certificates.

IPsec v1.8.12 and later supports the CA certificate chain. Concatenate the contents of the root and the intermediate certificates as one of the list items in `ca_certificates`, with the root CA at the top:

```
ca_certificates:
- |
  -----BEGIN CERTIFICATE-----
  ... <ROOT>
  -----END CERTIFICATE-----
  -----BEGIN CERTIFICATE-----
  ... <INTERMEDIATE 1 ISSUED BY THE ROOT CERT ABOVE>
  -----END CERTIFICATE-----
  -----BEGIN CERTIFICATE-----
  ... <INTERMEDIATE 2 ISSUED BY THE INTERMEDIATE 1 ABOVE
  ... AND SIGNS THE INSTANCE CERT>
  -----END CERTIFICATE-----
- |
  -----BEGIN CERTIFICATE-----
  ... <SECOND ROOT>
  -----END CERTIFICATE-----
```

💡 Note: The root and the intermediate certificates cannot have the same subjectName. This is also called the common name, and is set with `CN=` . The root **must** be the first certificate of the chain.

- `prestart_timeout` : You can modify the 30-second default prestart timeout value. This value limits the number of seconds allowed for IPsec to start before failing the attempt.
- `log_level` : You can specify the IKE daemon numerical log level, ranging from `-1` to `4` . For more information, see [Logger Configuration](#) in the strongSwan documentation.
- `optional_warn_interval` : The interval in hours of warning when `optional` property is set to true. It prints the warning message `{Date} - IPsec is set to "Optional"` in the file `/var/vcap/sys/log/ipsec/ipsec.stdout.log` for Linux.
- `force_udp_encapsulation` : Available on Linux-only deployments. If set to `true` it forces UDP encapsulation for ESP packets.

WARNING: Setting this property to `true` in mixed deployments causes the deployment to fail. If you do not have a Linux-only deployment, you must set `force_udp_encapsulation` to `false`.

- `instance_certificate_info_period`: If the instance certificate expires during the set period in days, the IPsec release writes an `[INFO]` message in the logs.
- `instance_certificate_warn_period`: If the instance certificate expires during the set period in days, the IPsec release writes a `[WARN]` message in the logs.
- `instance_certificate_error_period`: If the instance certificate expires during the set period in days, the IPsec release writes an `[ERROR]` message in the logs.
- `instance_certificate_critical_period`: If the instance certificate expires during the set period in days, the IPsec release writes a `[CRITICAL]` message in the logs.

Add Windows VM Support to Your Runtime Config

To add IPsec to Windows Server 2012R2 VMs in your deployment, follow these steps:

1. Modify the `ipsec-addon.yml` created during the previous section to add the properties indicated in **bold** below under the `ipsec` key.

```
- name: ipsec-addon
  ...
  properties:
    ipsec:
      .
      .
      .
      ike_version: ikev1
      dpdaction: none
```

1. Add the following YAML under `addons:` to your `ipsec-addon.yml` file. Add it under the `ipsec-addon` section for Linux, if you included one [above](#).

```
- name: ipsec-windows-addon
  jobs:
    - name: ipsec-win
      release: ipsec
  include:
    stemcell:
      - os: windows2012R2
  properties:
    ipsec:
      optional: false
      ipsec_subnets:
      - 10.0.1.1/20
      no_ipsec_subnets:
      - 10.0.1.10/32 # bosh director
      - 10.0.1.4/32 # ops manager
      instance_certificate: |
      -----BEGIN CERTIFICATE-----
      MIEMDCCAhigAwIBAgIRAIvrBY2TttU/LerhO+V1t0YwDQYJKoZIhvcNAQELBQAw
      ...
      -----END CERTIFICATE-----
      instance_private_key: |
      -----BEGIN EXAMPLE RSA PRIVATE KEY-----
      EXAMPLExRSAxPRIVATExKEYxDATAxEXAMPLExRSAxPRIVATExKEYxDATA
      ...
      -----END EXAMPLE RSA PRIVATE KEY-----
      ca_certificates:
      - |
      -----BEGIN CERTIFICATE-----
      MIIFCTCCAvGgAwIBAgIBATANBgkqhkiG9w0BAQsFADAUMRIwEAYDVQQDEw10ZXN0
      ...
      -----END CERTIFICATE-----
      - |
      -----BEGIN CERTIFICATE-----
      MIIFCTCCAvGgAwIBAgIBATAAYDVQQDEw10ZXN0NBgkqhkiG9w0BAQsFADAUMRIwE
      ...
      -----END CERTIFICATE-----
      quick_mode_proposals:
      - encryption: AESGCM128
        hash: AESGMAC128
      main_mode_proposals:
```



```
- encryption: AES128
  hash: SHA256
  keyexchange: DH14
```

1. Replace the values listed in the template as follows: * `ipsec_subnets` : Copy and paste the value from `ipsec_subnets` for Linux. * `no_ipsec_subnets` : Copy and paste the value from `no_ipsec_subnets` for Linux. * `instance_certificate` : Copy and paste the value from `instance_certificate` for Linux. * `instance_private_key` : Copy and paste the value from `instance_private_key` for Linux. * `ca_certificates` : Copy and paste the value from `ca_certificates` for Linux. * `optional` : Copy and paste the value from `optional` for Linux.

WARNING: Communication between existing components fails if you try to add IPsec to an existing deployment without setting `optional` to `true`.

Optional: Custom Linux/Windows Mixed Deployment Proposals

A default proposal set is already selected for the `ipsec-addon.yml`. If you want to use different proposals, modify the `ipsec-addon.yml` using the following table:

1. Select the encryption type from the first row.
2. Copy the properties from that row into `ipsec-addon.yml` accordingly. See the `ipsec-addon.yml` file example above.

Encryption Type	Linux (ipsec-addon)		Windows (ipsec-win-addon)	
	ike_proposals	esp_proposals	main_mode_proposals	quick_mode_proposals
128 Bit Encryption	<code>aes128-sha256-modp2048!</code>	<code>aes128gcm16!</code>	<pre>- encryption: AES128 hash: SHA256 keyexchange: DH14</pre>	<pre>- encryption: AESGCM128 hash: AESGMAC128</pre>
256 Bit Encryption	<code>aes256-sha256-modp2048!</code>	<code>aes256gcm16!</code>	<pre>- encryption: AES256 hash: SHA256 keyexchange: DH14</pre>	<pre>- encryption: AESGCM256 hash: AESGMAC256</pre>

- `ike_proposals` : You can modify the IKE (Main Mode) encryption and integrity algorithms, and the Diffie-Hellman group. The default, `aes128-sha256-modp2048!`, is 128 bit AES-CBC for encryption, SHA2_256_128 HMAC for integrity, and Group 14 for Diffie-Hellman.
- `esp_proposals` : You can modify the ESP (Quick Mode) encryption and integrity algorithms. The default, `aes128gcm16!`, is 128 bit AES-GCM with 128 bit ICV for both encryption and integrity.
- `main_mode_proposals` : This is an array of Main Mode algorithms for encryption, integrity, and key exchange. This value must match the list specified in `ike_proposals` for Linux. See the table for proposal sets for both Linux and Windows. The default entry that matches the Linux default is:

```
- encryption: AES128
  hash: SHA256
  keyexchange: DH14
```

- `quick_mode_proposals` : This is an array of Quick Mode algorithms for encryption and integrity. This value must match the list specified in `esp_proposals` for Linux. See the table for proposal sets for both Linux and Windows. The default entry that matches the Linux default is:

```
- encryption: AESGCM128
  hash: AESGMAC128
```


Step 3: Download and Deploy the IPsec Add-on

To download the IPsec binary, add your IPsec runtime config to your BOSH manifest, and deploy the IPsec add-on, follow the [procedure](#) below.

Assumption about Ops Manager Versions

The procedure below assumes the following about your Ops Manager, BOSH CLI, and runtime config.

Ops Manager Version	BOSH CLI Version	Runtime Config	More information
1.10 and earlier	CLI v1	single, default, runtime config file	<ul style="list-style-type: none"> BOSH CLI v1
1.11 and later	CLI v2+	runtime config in multiple, named files, so that <code>ipsec</code> can be managed separately	<ul style="list-style-type: none"> BOSH CLI v2 Configs - bosh

 **Breaking Change:** If you are using PCF v1.11 or later, you must use named runtime configs. If you have not already split your runtime config into multiple named files, do so before upgrading the IPsec Add-on for PCF. For general information about named runtime config files, see [Configs](#).

Procedure

1. Download the IPsec add-on software binary from the [Pivotal Network](#) to your local machine.
2. Copy the software binary to your Ops Manager instance.

```
$ scp -i PATH-TO-PRIVATE-KEY ipsec-release.tar.gz ubuntu@YOUR-OPS-MANAGER-VM-IP:
```

3. Copy the IPsec runtime config file to your Ops Manager instance.

```
$ scp -i PATH-TO-PRIVATE-KEY ipsec-addon.yml ubuntu@YOUR-OPS-MANAGER-VM-IP:
```

4. SSH into Ops Manager.

```
$ ssh -i PATH-TO-PRIVATE-KEY ubuntu@YOUR-OPS-MANAGER-VM-IP
```

5. On the Ops Manager VM, navigate to the software binary location in your working directory.

```
$ cd PATH-TO-BINARY
```

6. Log in to the BOSH Director.

- **For Ops Manager v1.10 or earlier:**

- i. On the Ops Manager VM, target the internal IP address of your BOSH Director. When prompted, enter your BOSH Director credentials. To retrieve your BOSH Director credentials, navigate to Ops Manager, click the **Credentials** tab, and click **Link to Credential** next to **Director Credentials**. For example:

```
$ bosh --ca-cert /var/tempest/workspaces/default/root_ca_certificate target YOUR-BOSH-DIRECTOR-INTERNAL-IP
Target set to 'p-bosh'
Your username: director
Enter password: *****
Logged in as 'director'
```

- **For Ops Manager v1.11 or later:**

- i. On the Ops Manager VM, create an alias in the BOSH CLI for your Ops Manager Director IP address. For example:

```
$ bosh2 alias-env my-env -e 10.0.0.3
```

- ii. Log in to the BOSH Director, specifying the newly created alias. For example:

```
$ bosh2 -e my-env log-in
```

7. Upload your release, specifying the path to the tarballed IPsec binary, by running one of the following commands:

- **For Ops Manager v1.10 or earlier:**

```
$ bosh upload release PATH-TO-BINARY/BINARY-NAME.tar
```

- **For Ops Manager v1.11 or later:**

```
$ bosh2 -e my-env upload-release PATH-TO-BINARY/BINARY-NAME.tar
```

8. List the releases by running one of the following commands, and confirm that the IPsec binary file appears:

- For Ops Manager v1.10 or earlier:

```
$ bosh releases
```

- For Ops Manager v1.11 or later:

```
$ bosh2 -e my-env releases
```

9. Download your current runtime config and save as `bosh-manifest.yml` by running one of the following commands:

- For Ops Manager v1.10 or earlier:

```
$ bosh runtime-config > bosh-manifest.yml
```

- For Ops Manager v1.11 or later:

```
$ bosh2 -e my-env runtime-config > bosh-manifest.yml
```

10. For Ops Manager v1.10 or earlier:

Append the contents of your IPsec manifest `ipsec-addon.yml` to `bosh-manifest.yml`.

11. Update your runtime configuration to include the IPsec add-on.

- For Ops Manager v1.10 or earlier:

```
$ bosh update runtime-config PATH/bosh-manifest.yml
```

- For Ops Manager v1.11 or later:

```
$ bosh2 -e my-env update-runtime-config --name=ipsec PATH/bosh-manifest.yml
```

12. Verify your runtime configuration changes match what you specified in the IPsec manifest file.

- For Ops Manager v1.10 or earlier:

```
$ bosh runtime-config
```

- For Ops Manager v1.11 or later:

```
$ bosh2 -e my-env runtime-config --name=ipsec
```

For example:

```
$ bosh2 -e my-env runtime-config --name=ipsec
Acting as user 'admin' on 'micro'

releases:
- {name: ipsec, version: 1.0.0}

addons:
name: ipsec-addon
jobs:
- name: ipsec
  release: ipsec
...
- name: ipsec-win # if using Windows
  release: ipsec
...
```

13. If you have already deployed PAS (or Elastic Runtime) or are adding IPsec to an existing deployment:

- Set the `optional` flag to `true`.
- Navigate to your **Installation Dashboard** in Ops Manager.
- Click **Apply Changes**
- Wait for the installation to complete.
- Set the `optional` flag to `false`.
- Update the runtime config.

- For Ops Manager v1.10 or earlier:

```
$ bosh update runtime-config PATH/bosh-manifest.yml
```

- For Ops Manager v1.11 or later:

```
$ bosh2 -e my-env update-runtime-config --name=ipsec PATH/bosh-manifest.yml
```

- Navigate to your **Installation Dashboard**.
- Click **Apply Changes**.

14. If the PAS (or Elastic Runtime) tile is not yet installed:

- Navigate to your **Installation Dashboard** in Ops Manager.
- Click **Apply Changes**
- Deploy PAS (or Elastic Runtime) by following the installation instructions for your IaaS. For more information, see [Installing Pivotal Cloud Foundry](#).

15. The `bosh-manifest.yml` and `ipsec-addon.yml` files contain sensitive information. When the deployment process is completed, be sure to remove any unneeded copies of these files from the local workstation. Pivotal recommends that any archival copies of manifest files to be retained should be appropriately secured via encryption and/or logical access controls.

Step 4: Verify Your IPsec Installation

After installing IPsec and deploying PAS (or Elastic Runtime), perform the following steps to verify your IPsec installation:

1. List the job VMs in your deployment by running one of the following commands:

- For Ops Manager v1.10 or earlier:

```
bosh vms
```

- For Ops Manager v1.11 or later:

```
bosh2 -e BOSH-ENVIRONMENT vms
```

2. Open an SSH connection into the VM, using the job name and index of any VM found above, by running one of the following commands:

- For Ops Manager v1.10 or earlier:

```
bosh ssh JOB-NAME/INDEX
```

- For Ops Manager v1.11 or later:

```
bosh2 -e BOSH-ENVIRONMENT -d DEPLOYMENT-NAME ssh JOB-NAME/INDEX
```



Note: The exact VM does not matter, because installing the IPsec add-on loads IPsec on all VMs deployed by Ops Manager.

3. Run `sudo su -` to enter the root environment with root privileges.

4. Run `monit summary` to confirm that your `ipsec` job is listed as a `bosh` job.

```
The Monit daemon 5.2.5 uptime: 18h 32m
...
Process 'ipsec'           running
System 'system_localhost' running
```

5. Run `PATH-TO-IPSEC/ipsec statusall` to confirm that IPsec is running. If IPsec is not running, this command produces no output.

```
$ /var/vcap/packages/strongswan-5.3.5/sbin/ipsec statusall
Status of IKE charon daemon (strongSwan 5.3.5, Linux 3.19.0-56-generic, x86_64):
uptime: 18 hours, since Mar 16 23:58:50 2016
malloc: sbrk 2314240, mmap 0, used 1182400, free 1131840
worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled: 206
loaded plugins: charon aes sha1 sha2 random nonce x509 revocation constraints pubkey pkcs1 pkcs7 pkcs8 pkcs12 pem gmp xcbc cmac hmac attr kernel-netlink socket-default stroke
Listening IP addresses:
10.10.5.66
Connections:
ipsec-10.10.4.0/24: %any...%any IKEv1/2
ipsec-10.10.4.0/24: local: [CN=test-cert-1-ca-1] uses public key authentication
ipsec-10.10.4.0/24: cert: "CN=test-cert-1-ca-1"
ipsec-10.10.4.0/24: remote: uses public key authentication
ipsec-10.10.9.0/24: child: 10.10.5.66/32 === 10.10.9.0/24 TRANSPORT
no-ipsec-10.10.4.1/32: %any...%any IKEv1/2
no-ipsec-10.10.4.1/32: local: uses public key authentication
no-ipsec-10.10.4.1/32: remote: uses public key authentication
no-ipsec-10.10.4.1/32: child: dynamic === 10.10.4.1/32 PASS
Shunted Connections:
no-ipsec-10.10.4.1/32: dynamic === 10.10.4.1/32 PASS
no-ipsec-10.10.5.1/32: dynamic === 10.10.5.1/32 PASS
no-ipsec-10.10.6.1/32: dynamic === 10.10.6.1/32 PASS
Routed Connections:
ipsec-10.10.9.0/24{6}: ROUTED, TRANSPORT, reqid 6
ipsec-10.10.9.0/24{6}: 10.10.5.66/32 === 10.10.9.0/24
ipsec-10.10.8.0/24{5}: ROUTED, TRANSPORT, reqid 5
ipsec-10.10.4.0/24{1}: 10.10.5.66/32 === 10.10.4.0/24
Security Associations (45 up, 0 connecting):
ipsec-10.10.4.0/24[459]: ESTABLISHED 13 seconds ago, 10.10.5.66[CN=test-cert-1-ca-1]...10.10.4.38[CN=test-cert-1-ca-1]
ipsec-10.10.4.0/24{1527}: 10.10.5.66/32 === 10.10.4.38/32
...
```

6. If you installed IPsec for Windows, follow these steps:

- a. From any Windows VM, open **Windows Firewall with Advanced Security**.
- b. Click **Connection Security Rules**.
- c. Confirm that you see rules for each `ipsec` and `no-ipsec` subnet that you listed in your manifest.

Generate a Self-Signed Certificate with OpenSSL

Follow these steps to generate a self-signed certificate for your IPsec manifest.

1. [Download](#) the `openssl-create-ipsec-certs.sh` bash script.

2. Navigate to the directory where you downloaded the script:

```
$ cd ~/workspace
```

3. Change the permissions of the script:

```
$ chmod u+x openssl-create-ipsec-certs.sh
```

4. Run the script:

```
$ ./openssl-create-ipsec-certs.sh
```

5. This generates four files in a new `certs` directory where the script is run:

- `pcf-ipsec-ca-cert.pem` — this value can be used as the CA Cert in the `ca_certificates` manifest field.
- `pcf-ipsec-ca-key.pem` — the key used to sign the generated CA Cert.
- `pcf-ipsec-peer-key.pem` — this value can be used as the instance private key in the `instance_private_key` manifest field.
- `pcf-ipsec-peer-cert.pem` — this value can be used as the instance certificate in the `instance_certificate` manifest field.

6. Because this certificate expires in 365 days, set a calendar reminder to rotate the certificate within the year. For instructions on changing certificates, see [Rotating IPsec Certificates](#).

Upgrading the IPsec Add-on for PCF

Page last updated:

This topic describes how to upgrade the IPsec Add-on for PCF.

Assumption about Ops Manager Versions

This topic assumes the following about your Ops Manager, BOSH CLI, and runtime config.

Ops Manager Version	BOSH CLI Version	Runtime Config	More information
1.10 and earlier	CLI v1	single, default, runtime config file	<ul style="list-style-type: none"> BOSH CLI v1
1.11 and later	CLI v2+	runtime config in multiple, named files, so that <code>ipsec</code> can be managed separately	<ul style="list-style-type: none"> BOSH CLI v2 Configs - bosh

Breaking Change: If you are using PCF v1.11 or later, you must use named runtime configs. If you have not already split your runtime config into multiple named files, do so before upgrading the IPsec Add-on for PCF. For general information about named runtime config files, see [Configs](#).

Upgrade IPsec Add-on

To upgrade the IPsec add-on to a later version, do the following:

1. Download the IPsec add-on software binary from the [Pivotal Network](#) to your local machine.
2. To copy the software binary to your Ops Manager VM, run the following command:

```
scp -i PATH-TO-PRIVATE-KEY ipsec-VERSION.tar.gz ubuntu@YOUR-OPS-MANAGER-VM-IP:
```

For example:

```
$ cp -i ~/.ssh/my-key.pem ~/Downloads/ipsec-1.8.14.tgz ubuntu@192.168.0.2:
```

3. SSH into the Ops Manager VM. For how to do this, see [SSH into Ops Manager](#).
4. Retrieve the latest runtime config by running one of the following commands:

For Ops Manager v1.10 or earlier:

```
bosh runtime-config > PATH-TO-SAVE-THE-RUNTIME-CONFIG
```

For example:

```
bosh runtime-config > /tmp/ipsec.yml
```

For Ops Manager v1.11 or later:

```
bosh2 -e BOSH-ENVIRONMENT runtime-config --name ipsec > PATH-TO-SAVE-THE-RUNTIME-CONFIG
```

For example:

```
bosh2 -e my-env runtime-config --name ipsec > /tmp/ipsec.yml
```

5. Upload the latest IPsec release:

For Ops Manager v1.10 or earlier:

```
bosh upload release PATH-TO-NEW-IPSEC-RELEASE
```

For example:

```
bosh upload release ~/ipsec-1.8.14.tgz
```

For Ops Manager v1.11 or later:

```
bosh2 -e BOSH-ENVIRONMENT upload-release PATH-TO-NEW-IPSEC-RELEASE
```

For example:

```
bosh2 -e my-env upload-release ~/ipsec-1.8.14.tgz
```

6. Edit the ipsec runtime config to set the new release version.

For example, edit the version in `/tmp/ipsec.yml` as follows:

```
releases:
- {name: ipsec, version: 1.8.14}
```

7. Update the runtime config:

For Ops Manager v1.10 or earlier:

```
bosh update runtime-config PATH-TO-SAVE-THE-RUNTIME-CONFIG
```

For example:

```
bosh update runtime-config /tmp/ipsec.yml
```

For Ops Manager v1.11 or later:

```
bosh2 -e BOSH-ENVIRONMENT update-runtime-config --name=ipsec PATH-TO-SAVE-THE-RUNTIME-CONFIG
```

For example:

```
bosh2 -e my-env update-runtime-config --name=ipsec /tmp/ipsec.yml
```

8. Navigate to your **Installation Dashboard** in Ops Manager.
9. Click **Apply Changes**.

Uninstalling the IPsec Add-on for PCF

Page last updated:

This topic describes how to uninstall IPsec from your deployment.

Uninstall the IPsec Add-On

1. Retrieve the latest runtime config by running one of the following commands:

- For Ops Manager v1.10 or earlier: `bosh runtime-config > PATH-TO-SAVE-THE-RUNTIME-CONFIG`
- For Ops Manager v1.11 or later: `bosh2 -e BOSH-ENVIRONMENT runtime-config > PATH-TO-SAVE-THE-RUNTIME-CONFIG`

2. Set the `optional` flag to `true` under IPsec properties.

3. Update the runtime config by running one of the following commands:

- For Ops Manager v1.10 or earlier: `bosh update runtime-config PATH/YOUR-RUNTIME-CONFIG.yml`
- For Ops Manager v1.11 or later: `bosh2 -e BOSH-ENVIRONMENT update-runtime-config --name=ipsec PATH-TO-SAVE-THE-RUNTIME-CONFIG`

4. Navigate to your **Installation Dashboard** in Ops Manager.

5. Click **Apply Changes**.

6. Wait for the installation to complete.

7. Remove IPsec from the runtime config.

8. Update the runtime config by running one of the following commands:

- For Ops Manager v1.10 or earlier: `bosh update runtime-config PATH/YOUR-RUNTIME-CONFIG.yml`
- For Ops Manager v1.11 or later: `bosh2 -e BOSH-ENVIRONMENT update-runtime-config --name=ipsec PATH-TO-SAVE-THE-RUNTIME-CONFIG`

9. Navigate to your **Installation Dashboard** in Ops Manager.

10. Click **Apply Changes**.

Checking Certificate Dates

Page last updated:

This topic describes how to check the expiration dates of IPsec certificates.

The following procedure describes how to download the runtime configuration file and extract the two IPsec certificates into temporary files. Then, the files are input to the OpenSSL tool. The OpenSSL tool decodes the certificates and displays the expiration dates.

Check Certificate Dates

Follow the steps below to determine the expiration dates of your IPsec certificates.

1. Log in to BOSH Director.
2. Run one of the following commands to download your runtime configuration YAML file:

- **For Ops Manager v1.10 or earlier:** `bosh runtime-config > PATH-TO-SAVE-THE-RUNTIME-CONFIG`
- **For Ops Manager v1.11 or later:** `bosh2 -e BOSH-ENVIRONMENT runtime-config --name=ipsec > PATH-TO-SAVE-THE-RUNTIME-CONFIG`

For example,

```
bosh runtime-config > /tmp/my-runtime-config.yml
```

3. Display the runtime configuration YAML file so that you can copy from it.

For example,

```
$ cat /tmp/my-runtime-config.yml
```

4. Identify the section of the file that contains IPsec properties, and locate the certificates:

```
addons:
- include:
  stemcell:
    - os: ubuntu-trusty
jobs:
- name: ipsec
  release: ipsec
  name: ipsec
  properties:
    ipsec:
      ca_certificates:
        - |
          -----BEGIN CERTIFICATE-----
          MIIE/TCCAUWgAwIBAgIBATANBgkqhkiG9w0BAQsFADAOMQwwCgYDVQQDEwNjYTEw
          HhcNMjYwNTI2MjI1MDMzWheNMjYwNTI2MjI1MDQyWjAOMQwwCgYDVQQDEwNjYTEw
          ...
          Axu2pbEoT1PrMd3HIAZ3AH8ZrMR3ScJKCW3wQFRX/Plj
          -----END CERTIFICATE-----
      instance_certificate: |
        -----BEGIN CERTIFICATE-----
        MIIEGTCCAAGGAgAwIBAgIQDlqK1V54BEknblVPXu5lzANBgkqhkiG9w0BAQsFADAOMQwwCgYDVQQDEwNjYTEwHhcNMjYwNTI2MjI1MDQyWjAOMQwwCgYDVQQDEwNjYTEw
        MQ4wDAYDVQQDEwVjZXI0MTCCASlwdQYJKoZIhveNAQEBBQADggEPADCCAQoCggEB
        ...
        4Q6P/cDn9QvW2QbbWkApP2uuMk04jWJV7p79CfX4pipPqiSofjFyFqsjjvir
        -----END CERTIFICATE-----
```

5. Copy the ca_certificate into a text file. Retain the header and footer, but delete the leading white space before the -----BEGIN CERTIFICATE----- and ---END CERTIFICATE----- lines.

For example,

```
-----BEGIN CERTIFICATE-----
MIIE/TCCAUWgAwIBAgIBATANBgkqhkiG9w0BAQsFADAOMQwwCgYDVQQDEwNjYTEw
HhcNMjYwNTI2MjI1MDMzWheNMjYwNTI2MjI1MDQyWjAOMQwwCgYDVQQDEwNjYTEw
...
Axu2pbEoT1PrMd3HIAZ3AH8ZrMR3ScJKCW3wQFRX/Plj
-----END CERTIFICATE-----
```

6. Save the file with the PEM extension, for example, `my-ipsec-ca-cert.pem`.

7. Run the following command:

```
openssl x509 -text -inform pem -in /PATH/FILENAME.pem | grep "Not After"
```

Where `/PATH/FILENAME.pem` is the path to and filename of the file you saved in the step above.

For example,

```
$ openssl x509 -text -inform pem -in /tmp/my-ipsec-ca-cert.pem | grep "Not After"
Not After : May 26 22:50:42 2026 GMT
```

If the PEM file is correctly formatted, the output shows a line with the `Not After` date.

If the PEM file is not correctly formatted, The output shows `unable to load certificate`.

8. Repeat steps 5–7 for the `instance_certificate`.

9. Review the `Not After` date and plan to replace the certificates accordingly.

Keep in mind the lead time to obtain new certificates and the time to perform a deployment to apply them.

For information, see [Rotating Active IPsec Certificates](#).

10. For security hygiene, delete three temporary files that you created: the downloaded copy of the `runtime-config.yml` which contains the private key and the two PEM files that contain the certificates.

Rotating Active IPsec Certificates

Page last updated:

This topic describes the process Pivotal recommends to increase deployment security by rotating certificates in the IPsec manifest.

Why You Need to Rotate Credentials


These are common reasons for rotating credentials:

- Your organizational security policy may specify how often you should apply these changes.
- Your certificates are going to expire. To find the expiration dates on your certificates, see [Checking Certificate Dates](#).

About the Procedures

There are two procedures for certificate rotation described in this topic:


- [Procedure 1](#) describes rotating the following certificates specified in your IPsec manifest:
 - The instance certificate and instance private key
This procedure requires updating BOSH. It does not include rotating the certificate authority (CA) certificate.
- [Procedure 2](#) describes rotating your CA certificate in addition to your instance certificate and instance private key. This procedure requires updating BOSH three times.

 **Note:** The rolling deploys during these procedures result in minimal deployment downtime.

Procedure 1: Rotate the Instance Certificate and Instance Private Key

Follow the steps below to rotate the instance certificate and instance private key.

1. Generate a new certificate and use your existing IPsec CA certificate to sign the new certificate.
2. Update the instance certificate and the private key fields in your `ipsec-addon.yml` file with new values from the previous step.
3. Update the runtime config by running one of the following commands:
 - **For Ops Manager v1.10 or earlier:** `bosh update runtime-config PATH-TO-SAVE-THE-RUNTIME-CONFIG`
 - **For Ops Manager v1.11 or later:** `bosh2 -e BOSH-ENVIRONMENT update-runtime-config --name=ipsec PATH-TO-SAVE-THE-RUNTIME-CONFIG`

 **Note:** This step results in a few minutes of app downtime.

4. Navigate to your Ops Manager interface in a browser, and click **Apply Changes**.

Procedure 2: Rotate the CA Certificate, the Instance Certificate, and Instance Private Key

Follow these steps to rotate the CA certificate, instance certificate, and instance private key.

1. Generate a new CA certificate.
2. Append the newly generated CA certificate under the existing certificate as a new yaml list element in your `ipsec-addon.yml`. For example:

```
ca_certificates:
- |
  -----BEGIN CERTIFICATE-----
  ... <ORIGINAL ROOT>
  -----END CERTIFICATE-----
```

```
- |
  -----BEGIN CERTIFICATE-----
  ... <NEW ROOT>
  -----END CERTIFICATE-----
  .
  .
  .
```

For **v1.8.12** and above: IPsec supports CA certificate chain.

Concatenate the contents of the root and the intermediate certificates as one of the list items in **ca_certificates** (the root CA is at the top).

```
ca_certificates:
- |
  -----BEGIN CERTIFICATE-----
  ... <ORIGINAL ROOT>
  -----END CERTIFICATE-----
- |
  -----BEGIN CERTIFICATE-----
  ... <NEW ROOT>
  -----END CERTIFICATE-----
  -----BEGIN CERTIFICATE-----
  ... <INTERMEDIATE 1 ISSUED BY THE NEW ROOT CERT ABOVE>
  -----END CERTIFICATE-----
  -----BEGIN CERTIFICATE-----
  ... <INTERMEDIATE 2 ISSUED BY THE INTERMEDIATE 1 ABOVE
  ... AND SIGNS THE NEW INSTANCE CERT>
  -----END CERTIFICATE-----
```



Note: The root and the intermediate certificates cannot have the same subjectName, (also called the common name and set with `CN=`). Also, the root certificate **must** be the first certificate of the chain.

3. Update the runtime config by running one of the following commands:

- For Ops Manager **v1.10** or earlier: `bosh update runtime-config PATH-TO-SAVE-THE-RUNTIME-CONFIG`
- For Ops Manager **v1.11** or later: `bosh2 -e BOSH-ENVIRONMENT update-runtime-config --name=ipsec PATH-TO-SAVE-THE-RUNTIME-CONFIG`



Note: This step results in a few minutes of app downtime.

4. Navigate to your Ops Manager interface in a browser, and click **Apply Changes**.

5. Generate a new certificate and use your new CA certificate to sign the new certificate.

6. Update the instance certificate and the private key fields in the your `ipsec-addon.yml` file with new values from above.

7. Repeat step 3 to update the runtime config.

8. Navigate to your Ops Manager interface in a browser, and click **Apply Changes**.

9. Delete the older CA certificate in the `ipsec-addon.yml` file.

10. Repeat step 3 to update the runtime config.

11. Navigate to your Ops Manager interface in a browser, and click **Apply Changes**.

Renewing Expired IPsec Certificates


Page last updated:

This topic describes the basic process that deployers may use to renew any already expired certificates contained in the IPsec manifest.

About Certificate Expiration

The IPsec Add-on relies upon X.509 certificates to secure the communications between communicating peers.

Like all certificates, the IPsec certificates have a finite lifetime and eventually expire. The certificates generated by the procedure provided in the installation instructions, [Generate a Self-Signed Certificate](#) have a default lifetime of one year. Regardless of their specific lifetime, all certificates must eventually be rotated, and so it is important for the operations team to plan accordingly and remember to rotate the IPsec certificates before they actually expire.

 **IMPORTANT:** Rotating the certificates while they are still valid ensures the maximum availability of the Cloud Foundry platform and avoids any unscheduled interruption in service.

Renew Expired IPsec Certificates

To renew expired IPsec certificates, do the following:

1. Retrieve the latest runtime config by running one of the following commands:
 - For Ops Manager v1.10 or earlier: `bosh runtime-config > PATH-TO-SAVE-THE-RUNTIME-CONFIG`
 - For Ops Manager v1.11 or later: `bosh2 -e BOSH-ENVIRONMENT runtime-config > PATH-TO-SAVE-THE-RUNTIME-CONFIG`
2. Generate a new set of certificates. For development or test environments, you can use self-signed certificates. For information about self-signed certificates, see [Generate a Self-Signed Certificate](#).
3. In the runtime `config.yml` file saved from step 1, update the `optional` field to `true` and update the certificate fields with new certificates. For more information about these fields, see the field descriptions under [Create the IPsec Manifest](#).

```
properties:
  ipsec:
    optional: true
    instance_certificate: |
      -----BEGIN CERTIFICATE-----
      EXAMPLEAhigAwIBAgIRAIvrBY2TttU/LeRhO+Vlt0YwDQYJKoZIhvcNAQELBQAw
      ...
      -----END CERTIFICATE-----
    instance_private_key: |
      -----BEGIN EXAMPLE RSA PRIVATE KEY-----
      EXAMPLExRSAXPRIVATExKEYxDATAxEXAMPLExRSAXPRIVATExKEYxDATA
      ...
      -----END EXAMPLE RSA PRIVATE KEY-----
    ca_certificates:
      - |
        -----BEGIN CERTIFICATE-----
        ExampleAvGgAwIBAgIBATANBgkqhkiG9w0BAQsFADAUMRIwEAYDVQQDEw10ZXN0
        ...
        -----END CERTIFICATE-----
```

4. Update the runtime config by running one of the following commands:
 - For Ops Manager v1.10 or earlier: `bosh update runtime-config PATH-TO-SAVE-THE-RUNTIME-CONFIG`
 - For Ops Manager v1.11 or later: `bosh2 -e BOSH-ENVIRONMENT update-runtime-config --name=ipsec PATH-TO-SAVE-THE-RUNTIME-CONFIG`
5. Navigate to your **Installation Dashboard** in Ops Manager.
6. Click **Apply Changes**.
7. Remove the `optional: true` set in step 3.
8. Repeat steps 4 to 6.

