Pivotal

PRODUCT DOCUMENTATION

PCF IPsec Add-On®

Version 1.5

User's Guide

Rev: 03

Table of Contents

Table of Contents 2
IPsec Add-on for PCF 3
Installing the IPsec Add-on for PCF 4
Upgrading the IPsec Add-on for PCF 10
Troubleshooting the IPsec Add-on for PCF 11
Rotating IPsec Certificates 17
Renewing Expired IPsec Certificates 19
Release Notes 21

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.

Note: If you apply the IPsec add-on to your PCF deployment, you cannot remove IPsec without removing and reinstalling the entire deployment.

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 to each BOSH-deployed virtual machine (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.

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

Refer to the following topics for more information about the IPsec add-on:

- Installing the IPsec Add-on for PCF
- Rotating IPsec Certificates
- Renewing Expired IPsec Certificates
- Troubleshooting the IPsec Add-on for PCF
- Upgrading the IPsec Add-on for PCF
- Release Notes

Installing the IPsec Add-on for PCF

Page last updated:

This topic describes how to prepare your network for IPsec, create an IPsec manifest, 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 1.7 or later
- Set the MTU for your IaaS in the 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](https://wiki.openstack.org/wiki/Neutron/ML2) plugin provider, or empirically test the correct MTU for your environment.

Limitations

Before installing the IPsec add-on, review the following list of limitations:

- During installation, the IPsec add-on may disrupt connectivity. For minimal downtime in new deployments, Pivotal recommends installing IPsec immediately after Ops Manager, and before installing the Elastic Runtime tile. For existing deployments, there will be downtime when applying IPsec. For a cluster of approximately 40 VMs, you can expect approximately one hour of downtime. Use v1.6.x or later to avoid downtime.
- IPsec may affect the functionality of other service tiles. As a result, Pivotal recommends deploying 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 Elastic Runtime subnet. Some service tiles may not support IPsec, and must be placed in a non-IPsec subnet.
- For IPsec, Pivotal recommends any Ubuntu stemcells for vSphere and OpenStack, and HVM stemcells for AWS. These stemcells are available on Pivotal Network. If you are using PV stemcells obtained from bosh.io, see Preventing Packet Loss in the Troubleshooting topic to adjust MTU values.
- The IPsec add-on v1.5 does not work on Windows.

Configure Network Security

Refer to the appropriate section below for your IaaS network configuration details.

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 Preventing Packet Loss section of the Troubleshooting the IPsec Add-on for PCF topic.

vSphere

Confirm that your network allows the following protocols:

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

Azure

1. Confirm that your network allows the following protocols:

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. Set up 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. Set up 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

Create the IPsec Manifest

Follow these steps to create the IPsec manifest for your deployment:

1. Create an IPsec manifest file `ipsec-addon.yml` starting with the code below as a template.

```
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
      - 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-----
          MIIFCTCCAvGgAwIBAgIBATANBgkqhkiG9w0BAQsFADAUMRIwEAYDVQQDEwl0ZXN0
          ...
          -----END CERTIFICATE-----
        - |
          -----BEGIN CERTIFICATE-----
          MIIFCTCCAvGgAwIBAgIBATAAYDVQQDEwl0ZXN0NBgkqhkiG9w0BAQsFADAUMRIwE
          ...
          -----END CERTIFICATE-----
       prestart_timeout: 30
```

2. Replace the properties listed in the file as follows:
   - `releases: - version:` Specify the version number of your IPsec download from Pivotal Network.
   - `jobs: - name:` Do not change the name of this job. It must be `ipsec`.
   - `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 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 must be listed under no-ipsec.

     > **Note**: In GCP, if you are using 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:` Paste in the signed certificate for use by all your instance VMs. You must use one of the CAs in the ca_certificates property to sign this certificate. For a development or test environment, you can use a self-signed certificate. See the Generate a Self-Signed Certificate section of this topic for more information.
   - `instance_private_key:` Paste in the private key that corresponds to the instance_certificate above. The key must not use a pass phrase.
   - `ca_certificates:` Paste 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 certificate rotation, you need two CA certificates.
   - `prestart_timeout:` You can optionally modify the 30 second default prestart timeout value. The value limits the number of seconds allowed for IPsec to start before failing the attempt.

> **Note:** To modify the configuration in an existing deployment, you must update the manifest file and redeploy.

Download and Deploy the IPsec Add-on

⚠ **WARNING:** Adding the IPsec add-on to an existing deployment results in app downtime. Use IPsec Add-on for PCF v1.6.x or later to avoid app downtime.

After deploying Ops Manager, perform the following steps to download and deploy the IPsec add-on:

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 manifest 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. 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**.

```
$ 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'
```

7. Upload your release.

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

8. Optionally, from the command line, confirm that the upload of the IPsec software binary completed. You should see the IPsec binary file.

```
$ bosh releases
```

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

```
$ bosh update runtime-config PATH/ipsec-addon.yml
```

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

```
$ bosh runtime-config
Acting as user 'admin' on 'micro'

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

addons:
 name: ipsec-addon
  jobs:
  - name: ipsec
    release: ipsec
...
```

11. If you have already deployed Elastic Runtime and are adding IPsec to an existing deployment, scale down the number of Consul instances to 1. See the Scaling Elastic Runtime topic for more information.

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

Note: The following step cannot be undone. After you deploy IPsec, you cannot roll it back.

13. Click **Apply Changes**.

14. If you scaled down the number of Consul instances in your existing Elastic Runtime deployment, scale the number of instances back to the previous value.

15. Secure the sensitive information in the `ipsec-addon.yml` file. Pivotal recommends encrypting the file and/or moving it to a secure location.

16. If you have not installed the Elastic Runtime tile, deploy Elastic Runtime by following the installation instructions for your IaaS. See the Installing Pivotal Cloud Foundry topic for more information.

Verify Your IPsec Installation

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

1. Run `bosh vms` to list the job VMs in your deployment. Choose the job name and index of any VM. The exact VM does not matter, because installing the IPsec add-on loads IPsec on all VMs deployed by Ops Manager.

2. Run `bosh ssh JOB-NAME/INDEX` to open a SSH connection into the VM.

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
...
```

Generate a Self-Signed Certificate

Note: Use a self-signed certificate only for development or test environments. Do not use a self-signed certificate for a production deployment. The scripts below generate private keys in a `certs` subdirectory.

To generate a self-signed certificate for your IPsec manifest, you can use either `openssl` or `certstrap`. Follow the instructions for your preferred method below.

Rerunning the scripts overwrites your current keys and the certificates.

Generate a Self-Signed Certificate with OpenSSL

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. 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.

Generate a Self-Signed Certificate with Certstrap

1. Download and install Go.

2. Download the `certstrap` bash script.

3. Change into the directory where you downloaded the script:

```
$ cd ~/workspace
```

4. Change the permissions of the script:

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

5. Run the script:

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

Upgrading the IPsec Add-on for PCF

Page last updated:

This topic describes how to upgrade the IPsec add-on.

Upgrading the IPsec Add-On

1. Retrieve the latest runtime config.

```
$ bosh runtime-config > PATH_TO_SAVE_THE_RUNTIME_CONFIG
```

2. Change the release version.

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

3. Update the runtime config.

```
$ bosh update runtime-config PATH_TO_SAVE_THE_RUNTIME_CONFIG
```

4. Upload your new release.

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

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

6. Click **Apply Changes**.

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 applications 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 applications 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:

   ```
   $ bosh ssh JOB INDEX
   ```

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

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

3. Run `bosh cloudcheck` to detect the missing monit job VM.

   ```
   # 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 descritor 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-----
        MIIEMDCCAhigAwIBAgIRAIvrBY2TttU/LeRhO+V1t0YwDQYJKoZIhvcNAQELBQAw
        ...
        -----END CERTIFICATE-----
      instance_private_key: |
        -----BEGIN EXAMPLE RSA PRIVATE KEY-----
        MIIEogIBAAKCAQEAtAkBjrzr5x9g0aWgyDEmLd7m9u/ZzpK7UScfANLaN7JiNz3c
        ...
        -----END EXAMPLE RSA PRIVATE KEY-----
      ca_certificates:
        - |
        -----BEGIN CERTIFICATE-----
        MIIEUDCCArigAwIBAgIJAJVLBeJ9Wm3TMA0GCSqGSIb3DQEBCwUAMB0xGzAZBgNV
        BAMMElBDRiBJUHNlYyBBZGRPbiBDQTAeFw0xNjA4MTUxNzQwNDVaFw0xOTA4MTUx
        ...
        -----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.

Rotating IPsec Certificates

Page last updated:

This topic describes the process Pivotal recommends to increase deployment security by rotating certificates in the IPsec manifest. Your organizational security policy may specify how often you should apply these changes.

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

- The first procedure describes rotating the instance certificate and instance private key specified in your IPsec manifest, and requires updating BOSH. This procedure does not include rotating the certificate authority (CA) certificate.
- The second procedure 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. Run the following command:

```
$ bosh update runtime-config PATH/MYPRODUCT_ipsec-addon.yml
```

Note: The following step results in a few minutes of application 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 below to rotate the CA certificate, the 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-----
    ...
    -----END CERTIFICATE-----
 - |
    -----BEGIN CERTIFICATE-----
    ...
    -----END CERTIFICATE-----
    .
    .
    .
```

3. Run the following command:

```
$ bosh update runtime-config PATH/MYPRODUCT_ipsec-addon.yml
```

Note: The following step results in a few minutes of application downtime.

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

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

6. Update the instance certificate and the private key fields in the YAML file with new values from above.

7. Run the following command:

   ```
   $ bosh update runtime-config PATH/MYPRODUCT_ipsec-addon.yml
   ```

   Note: The following step results in a few minutes of application downtime.

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. Run the following command:

    ```
    $ bosh update runtime-config PATH/MYPRODUCT_ipsec-addon.yml
    ```

    Note: The following step results in a few minutes of application downtime.

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 soon-to-be-expiring 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 expiring IPsec certificates, do the following:

1. Retrieve the latest runtime config by running the following command:

   ```
   bosh runtime-config > PATH-TO-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+V1t0YwDQYJKoZIhvcNAQELBQAw
           ...
           -----END CERTIFICATE-----
          instance_private_key: |
           -----BEGIN EXAMPLE RSA PRIVATE KEY-----
           EXAMPLExRSAxPRIVATExKEYxDATAxEXAMPLExRSAxPRIVATExKEYxDATA
           ...
           -----END EXAMPLE RSA PRIVATE KEY-----
          ca_certificates:
           - |
             -----BEGIN CERTIFICATE-----
             ExampleAvGgAwIBAgIBATANBgkqhkiG9w0BAQsFADAUMRIwEAYDVQQDEwl0ZXN0
             ...
             -----END CERTIFICATE-----
   ```

4. Update the runtime config by running the following command:

   ```
   bosh update runtime-config 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.

Release Notes

Page last updated:

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

v1.5.37

- **Bug Fixes**: This release corrects an issue that was discovered when the no-IPsec subnet configuration is empty.

New Features

- **StrongSwan**: This version of the IPsec add-on now uses StrongSwan version 5.5.0
- **Gentle Shutdown**: This version includes additional shutdown processing logic with the goal of ensuring other jobs are allowed time to exit cleanly at shutdown. Shutdown will be delayed until other jobs have exited, up to a deployer-configurable timeout value.

v1.5.31

New Features

- **StrongSwan**: This version of the IPsec add-on now uses StrongSwan version 5.4.0
- **Supported IAASes**: This version of the IPsec add-on adds support for OpenStack. It has been tested with OpenStack version 8.0.1, using Elastic Runtime version 1.7.9, and Ops Manager version 1.7.1 and greater. The IPsec add-on is now supported on AWS, vSphere, and OpenStack.
- **Bug Fixes**: This latest release corrects an issue that was discovered that could cause communication between hosts located in the IPsec subnet, and the no-IPsec subnet, to fail. This release includes a more robust implementation of the processing logic that controls the StrongSwan configuration which ensures that communication in a heterogeneous deployment (containing both IPsec and no-IPsec subnets) should always work as intended.

Known Issues

- **Spurious Configuration Warning**: As part of the upgrade to StrongSwan version 5.4.0, this version of the IPsec add-on will emit a sequence of spurious configuration warning messages. The messages will appear 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 plug-ins. 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 plug-ins. Any unused plug-ins are not loaded. The newest version of StrongSwan now issues this warning message when it detects that situation. The actual list of plug-ins in use has been determined to be appropriate for use of StrongSwan in the PCF environment. This warning is expected, and should be ignored.
- **Certificate Verification**: There is a known issue with the CA certificate validation. The IPsec add-on supports certificate rotation with minimal downtime. The host instance certificate can be rotated at any time by doing a deployment. In addition, the CA certificate that is used to verify trust in the host certificates can be rotated with minimal downtime by doing multiple deployments.

  However, because all VMs typically share the same instance certificate, they will trust each other without relying upon the CA certificate. The CA certificate is not actually needed until the operator does a deployment to rotate the instance certificate(s). While that deployment is running, some of the VMs will have received a new instance certificate, while other VMs are still operating using the prior instance certificate. During this time, while the instance certificates are different, the validation of the peer instance certificate will rely upon the common CA certificate in order to establish trust in the counterparty.

  If the CA certificate is malformed, or otherwise invalid, this problem will remain latent until the time when the instance certificate is being rotated. It is only during that deployment when the operator will discover that the CA certificate is not valid. Of course, as long as the CA certificate is valid, there is no problem.

It is recommended that operators use a tool such as OpenSSL to verify that the CA certificate they are choosing to configure is in fact valid, and contains the appropriate details for proper end-entity authentication of the VM in the deployment (such as subjectName, issuerName, and validity dates, etc).

Operators can use their favorite certificate management tool to confirm that their certificate matches what they expect. Using OpenSSL, one can issue the command:

```
$ openssl x509 -in myCA.crt -text
```

If this command produces valid output, then the certificate will be OK when configured for IPsec.

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