Server Vdi Support

CB v4.2.4.150206.1225

February 06, 2015

Contents

Overview

Carbon Black provides support for Virtual Desktop Infrastructure (VDI) environments, in which client machines are frequently re-imaged. In these cases, the Carbon Black sensor a specific client needs to report to the server with the same sensor ID even though it is re-imaged to maintain the client event history.

Global Server VDI Support

Prerequisites

```
1. A Carbon Black enterprise server installation >= 4.1.2

2. Client image with the Carbon Black sensor ID set to 0
```

With the global VDI option, the client sensor must be configured to initially register with a sensor id of 0. This setting is global on the server - the server will attempt to correlate any sensor registration with an id of 0. The server will then attempt to correlate the client to an existing sensor registration, or register the client as a new sensor (if not seen before).

Sensor ID correlation is done via a plug-in that can be configured in the cb.conf configuration file. The Carbon Black enterprise server includes a default plug-in that correlates sensors based on the hostname of the client. Additional plug-ins can be created that correlate based on other characteristics, such as mac address, ip, etc.

1. Server configuration

   Edit the '/etc/cb/cb.conf' file and add the following configuration options:

   ```
   NewRegistrationCallbackModulePath=/usr/share/cb/plugins/default_new_sensor_registratio
   NewRegistrationCallbackClassName=DefaultNewRegistrationCallback
   ```

2. Restart cb-enterprise

   ```
   sudo service cb-enterprise restart
   ```

Group-based VDI Support

Prerequisites

```
1. A Carbon Black enterprise server installation >= 4.2
```

With the group VDI option, the server will attempt to correlate any sensor that is in a VDI-enabled group when that sensor registers with the server.

Sensor ID correlation is done via a plug-in that can be configured in the cb.conf configuration file. The Carbon Black enterprise server includes a default plug-in that correlates sensors based on the hostname of the client. Additional plug-ins can be created that correlate based on other characteristics, such as mac address, ip, etc.

1. Server configuration
   Edit the '/etc/cb/cb.conf' file and add the following configuration options:

   ```
   NewRegistrationCallbackModulePath=/usr/share/cb/plugins/default_new_sensor_registratio
   NewRegistrationCallbackClassName=DefaultNewRegistrationCallback
   ```

2. Restart cb-enterprise

   ```
   sudo service cb-enterprise restart
   ```

3. Group configuration
   Bring up the 'Edit Settings' dialog for the group that will be configured for VDI support.
   On the 'Advanced' Tab, check the 'VDI Behavior Enabled' checkbox.

VDI Support Plugin Explained

Since, client machines are commonly re-imaged in a VDI environment, Carbon Black provides VDI support by using a server-side plug in to correlate a new (or what appears to be new) sensor registration to an existing sensor registration. This ensures that the event history for a specific client is maintained as a single sensor, regardless of re-imaging.

The default plug-in provided with the Carbon Black server is shown below:

```
~~~~~
#
# /usr/share/cb/plugins/default_new_sensor_registration_callback.py
#

from cb.sensor.NewRegistrationCallback import NewRegistrationCallback
from cb.db.core_models import SensorRegistration


class DefaultNewRegistrationCallback (NewRegistrationCallback):

    @staticmethod
    def get_sensor_id(db_session, sensor_reg_request, logger):
        sensor_host_name = sensor_reg_request.ComputerId.ComputerName
        sensor_dns_name = sensor_reg_request.ComputerId.ComputerDnsName

        sensor = db_session.query(SensorRegistration) \
            .filter(SensorRegistration.computer_name == sensor_host_name) \
            .filter(SensorRegistration.computer_dns_name == sensor_dns_name) \
            .order_by(SensorRegistration.last_checkin_time.desc()) \
            .first()
```

```
        if sensor is not None:
            logger.debug("Found sensor id [%d] for hostname [%s @ %s]" % (sensor.id, senso
            return sensor.id
        else:
            logger.debug("Could not find a sensor id for hostname hostname [%s @ %s]" % (s
            return 0
~~~~~
```

This plug-in model allows a deployment to be customized to perform correlation on any sensor characteristics that the server is aware of. The default plug-in (above) performs correlation based on the client hostname. If a sensor cannot be correlated during registration, the server will treat the registration as a new registration.

The basic requirements for creating a custom plug-in are:

```
1) Create a python script that contains a class that is a subclass of 'cb.sensor.NewRegist

2) Override the static method 'get_sensor_id(db_session, sensor_reg_reguest, logger)' and

3) Configure the server to use the new plug-in

    ~~~~~
    NewRegistrationCallbackModulePath=(path to new plug-in)
    NewRegistrationCallbackClassName=(name of NewRegistrationCallback subclass)
    ~~~~~
```