Mandiant

DEPLOY MANDIANT SECURITY VALIDATION ON GOOGLE CLOUD

This document shows you how to set up a Mandiant Security Validation (MSV) environment (a Director and an Actor) in Google Cloud. The process requires uploading the installed OVA (Director) and tar (Actor) files to Google Cloud Storage. Using these files, a Google Compute Engine instance is created. Follow these steps to complete the process:

1. Prepare Google Cloud environment
2. Install the Director
3. Install and configure the MSV Actor on Google Cloud

1. Prepare Google Cloud environment

Follow these steps to prepare your Google Cloud environment to deploy MSV:

1. **Prerequisites**
2. **Upload files to Google Cloud bucket**
3. **Provision Director VM with OVA**
4. **Provision Rocky Linux VM for Actor**
5. **Set up the Linux machine**

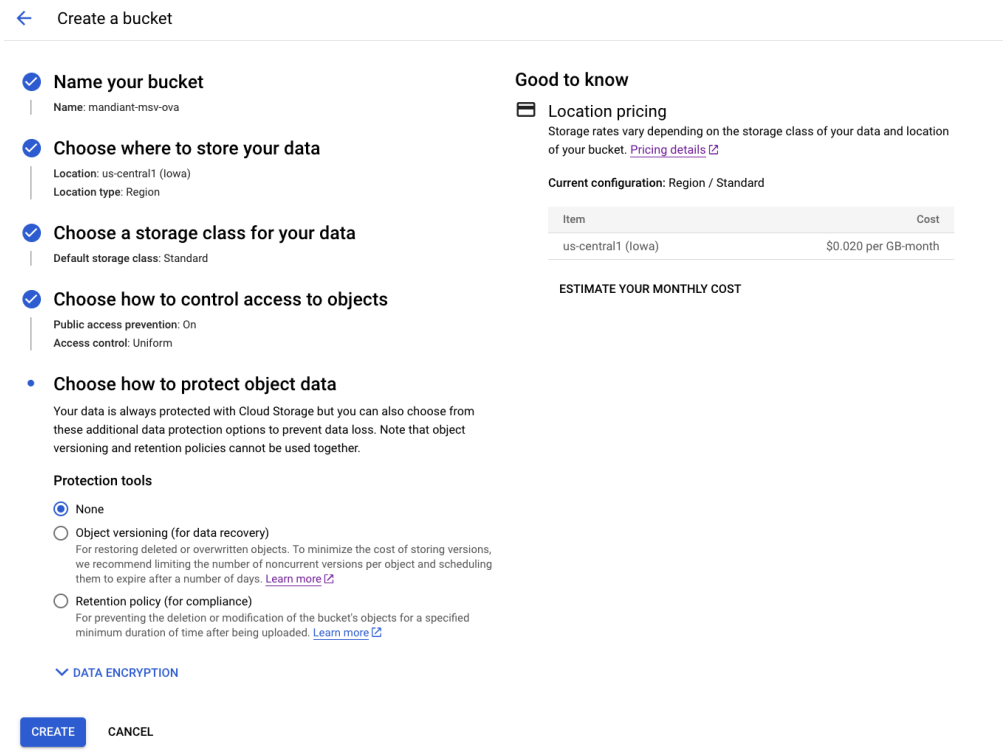
Prerequisites

- Three approved OS virtual machines (VMs) along with five static IPs
- A Mandiant account and an MSV license file.
- This document assumes that the Google Cloud organization is created, along with projects with at least two Virtual Private Cloud (VPC) networks created and with the API service enabled.
- Download the appropriate OVA and tar files. The setup consists of two different files: **Actor (https://docs.mandiant.com/home/msv-actor-installers)** and **Director (https://docs.mandiant.com/home/msv-director-installers)**.
    - **Director deployment:** The Director is deployed using the OVA file (`.ova`) through direct import. This method is fully supported for the Director's single NIC requirements.

      Actor deployment: The Actor is deployed by installing the Linux software (`.tar.gz`) on a base Rocky Linux VM.
        - **Network Actor vs. Endpoint Actor NIC Requirements:**
            - **Network Actor:** Requires two separate Network Interface Cards (NICs) – one for Management and one for Test traffic. These must be on different VPC networks.
            - **Endpoint Actor:** Requires only one NIC for management.
        - **Why not OVA for Actor:** MSV Network Actors instances have a mandatory requirement for two separate NICs. While Google Cloud generally supports importing VMs from OVA files, there are limitations in adding or reconfiguring network interfaces across different VPCs after an instance has been provisioned from an OVA.
        - **Recommended approach:** To ensure the Actor's NIC requirements are met, the supported and recommended method is to first provision a standard Rocky Linux VM in Google Cloud, configuring the required network interfaces in their respective VPCs during the instance creation process (two for Network Actors, one for Endpoint Actors). The MSV Actor software is then installed on this properly networked VM. This approach provides the necessary network configuration flexibility.
        - **Container technology:** Please note that MSV Actors utilize Podman for containerization. Docker Community Edition (docker-ce) is not supported and may cause conflicts if installed on the same VM.

Upload Director and Actor files to Google Cloud bucket

1. Once the file is downloaded, open Google Cloud.
2. Browse to Google Cloud Storage to create the bucket for the OVA file.



3. Once the bucket is created, click **Upload files** and then select tar and OVA files (Actor and Director). The files are several GB in size, so give time for the operation to complete.

Provision Director VM with OVA

1. Create a VM using the OVA file for the Director, available at **Director Downloads (https://docs.mandiant.com/home/msv-director-installers)**. This is the file you uploaded to the Google Cloud bucket. Execute the following gcloud terminal command to create the VM:

```
gcloud compute instances import mandiant-msv-director-gcp
--source-uri=gs://mandiant-msv-ova/director_[RELEASE_NUMBER].ova --zone=us-west1-a
```

   Where *RELEASE_NUMBER* is the Director release that you downloaded. For example:

```
gcloud compute instances import mandiant-msv-director-gcp
--source-uri=gs://mandiant-msv-ova/director_4.14.0.0-84.ova --zone=us-west1-a
```

You may receive errors when initially importing the OVA File. If you encounter the following errors, attempt the import command again.

```
ERROR: (gcloud.compute.instances.import) HTTPError 404: The resource 'projects/example/zones/us-west1-
a/instances/mandiant-msv-director-gcp' was not found. [import-ovf]: 2023-08-29T16:10:59Z error while
opening archive gs://mandiant-msv-files/director_4.14.0.0-84.ova: googleapi: got HTTP response
code 403 with body:
<!--?xml version='1.0' encoding='UTF-8'?--><code>AccessDenied</code>Access denied.
<details> does not have storage.objects.get access to the
Google Cloud Storage object. Permission 'storage.objects.get' denied on resource
(or it may not exist).</details>
```

The process should take approximately 10 minutes to complete. Upon completion, a `Cleaning up.` status appears, followed by the required cleanup activity steps.

2. Validate the VM was created from the file by connecting to the VM using the default SSH option.
3. After signing in, run the following command to verify that Mandiant web services are running:

```
ps aux
```

You should see a reference to the verodin process.

Provision Rocky Linux VM for Actor

1. Click **Create Instance** to create a Linux VM within Google Compute Engine. Ensure the file is changed to either RHEL or Rocky Linux.
2. Select the appropriate compute power required.
    - **Network Configuration:**
        - **Network Actors:** Typically require two different NICs to be associated with the VM. Ensure **Advanced Networking** is expanded and two different interfaces are added to the VM (management and test). Both Interfaces need to reside on different networks/VPCs.
        - **Endpoint Actors:** Require only one NIC for management.
    - The process should take less than two minutes to spin up a new VM.
3. Configure an Access Control List (ACL)/security group to allow network traffic on one of the interfaces from 443 for management (the Director IP address).

Set up the Linux machine

1. For the instance you created, click **SSH** to sign in using the Google Cloud console. A new browser opens with the terminal.

Container Runtime: Ensure that Docker Community Edition (docker-ce) is **not** installed on this VM. MSV Actors use Podman, and the presence of docker-ce can lead to conflicts.

2. Create a temporary folder for the uploaded files:

```
sudo mkdir temp
```

3. Copy the content from the Google Cloud Storage bucket (`tar.gz`) to the VM instance by running the `gsutil` command:

   ```
   sudo gsutil cp gs://BUCKET_NAME/FILENAME VM_DIRECTORY
   ```

   Where:

   - *FILENAME* is the Actor `tar.gz` file.
   - *VM_DIRECTORY* is the location of the VM instance.

   Example:

   ```
   sudo gsutil cp gs://mandiant-msv-ova/actor_4.14.0.0-84.tar.gz temp
   ```

   - The command saves the file to `home/temp` directly within the VM.
   - This should take less than one minute to copy over.

4. Browse to the directory and run the following command to ensure the copied file is there:

   ```
   ls
   ```

5. Unzip the Actor install file from the zip file to the local directory:

   ```
   sudo tar -xvf actor_4.14.0.0-84.tar.gz
   ```

   Output similar to the following appears:

```
actor_VERSION/.verodin-actor-install/array.cpython-38-x86_64-linux-gnu.so
actor_VERSION/.verodin-actor-install/_queue.cpython-38-x86_64-linux-gnu.so
actor_VERSION/.verodin-actor-install/_asyncio.cpython-38-x86_64-linux-gnu.so
actor_VERSION/.verodin-actor-install/_bz2.cpython-38-x86_64-linux-gnu.so
actor_VERSION/.verodin-actor-install/_csv.cpython-38-x86_64-linux-gnu.so
actor_VERSION/.verodin-actor-install/resource.cpython-38-x86_64-linux-gnu.so
actor_VERSION/.verodin-actor-install/readline.cpython-38-x86_64-linux-gnu.so
actor_VERSION/.verodin-actor-install/_opcode.cpython-38-x86_64-linux-gnu.so
actor_VERSION/.verodin-actor-install/_pickle.cpython-38-x86_64-linux-gnu.so
actor_VERSION/.verodin-actor-install/_heapq.cpython-38-x86_64-linux-gnu.so
actor_VERSION/.verodin-actor-install/_multibytecodec.cpython-38-x86_64-linux-gnu.so
actor_VERSION/.verodin-actor-install/_codecs_jp.cpython-38-x86_64-linux-gnu.so
actor_VERSION/.verodin-actor-install/_codecs_kr.cpython-38-x86_64-linux-gnu.so
actor_VERSION/.verodin-actor-install/_codecs_iso2022.cpython-38-x86_64-linux-gnu.so
actor_VERSION/.verodin-actor-install/_codecs_tw.cpython-38-x86_64-linux-gnu.so
actor_VERSION/.verodin-actor-install/_codecs_cn.cpython-38-x86_64-linux-gnu.so
actor_VERSION/.verodin-actor-install/_codecs_hk.cpython-38-x86_64-linux-gnu.so
actor_VERSION/.verodin-actor-install/grp.cpython-38-x86_64-linux-gnu.so
actor_VERSION/.verodin-actor-install/netifaces.cpython-38-x86_64-linux-gnu.so
actor_VERSION/.verodin-actor-install/_json.cpython-38-x86_64-linux-gnu.so
actor_VERSION/.verodin-actor-install/_datetime.cpython-38-x86_64-linux-gnu.so
actor_VERSION/.verodin-actor-install/libz.so.1
actor_VERSION/.verodin-actor-install/libkrb5support.so.0
actor_VERSION/.verodin-actor-install/libselinux.so.1
actor_VERSION/.verodin-actor-install/libk5crypto.so.3
actor_VERSION/.verodin-actor-install/libpcre.so.1
actor_VERSION/.verodin-actor-install/libkrb5.so.3
actor_VERSION/.verodin-actor-install/libcrypto.so.10
actor_VERSION/.verodin-actor-install/libssl.so.10
actor_VERSION/.verodin-actor-install/libgssapi_krb5.so.2
actor_VERSION/.verodin-actor-install/libkeyutils.so.1
actor_VERSION/.verodin-actor-install/libcom_err.so.2
actor_VERSION/.verodin-actor-install/libffi.so.6
actor_VERSION/.verodin-actor-install/libbz2.so.1
actor_VERSION/.verodin-actor-install/libreadline.so.6
actor_VERSION/.verodin-actor-install/libtinfo.so.5
actor_VERSION/.verodin-actor-install/base_library.zip
actor_VERSION/.verodin-actor-install/lib/
actor_VERSION/.verodin-actor-install/lib/python3.8/
actor_VERSION/.verodin-actor-install/lib/python3.8/config-3.8-x86_64-linux-gnu/
actor_VERSION/.verodin-actor-install/lib/python3.8/config-3.8-x86_64-linux-gnu/Makefile
actor_VERSION/.verodin-actor-install/include/
actor_VERSION/.verodin-actor-install/include/python3.8/
actor_VERSION/.verodin-actor-install/include/python3.8/pyconfig.h
actor_VERSION/.launcher
actor_VERSION/.verodin-actor-install
actor_VERSION/README
actor_VERSION/example-centos.ini
actor_VERSION/example-ubuntu.ini
USERNAME@mandiant-msv:~/home/temp$
```

Where:

- *VERSION* is the MSV software version.
- *USERNAME* is the account that you used to connect to SSH.

6. Create a service account for running MSV services, because `sudo` access is needed.

   This assumes that the wheel group is added to the sudoers file. In the event that wheel group is not added to the sudoers file, run the following:

   ```
   sudo useradd -G wheel USERNAME
   ```

   Where *USERNAME* is the service account you want added to the sudoers file.
   Example:

   ```
   sudo useradd -G wheel svc_mandiant
   ```

For further guidance, see **Configure routing for an additional network interface (https://cloud.google.com/vpc/docs/configure-routing-additional-interface)**.

2. Install the Director

The Director installer is provided as a gzipped tar archive file: `director_4.x.y.z.tar.gz`. Regardless of which installation method you use, this installer does the following:

- Copies the executables for certain programs into the `/usr/local/bin` directory[1]
- Sets up and migrate the database for the Director and runs system migrations

The installer does not includes a repository for which all dependencies are installed. It is up to you to set up a local mirror and install all dependencies.

If there are issues during installation, specific messages are provided so you can quickly resolve the issue and continue.

The Director tar file consists of the following items:

- `verodin-director-install` : the executable installer
- `files` : a folder containing files used by the installer
- `dependencies` : a folder containing the dependency packages
- `example.ini` : a sample ini file that can be used to automate the installation
- `README` : a file providing an overview of the install process

Installation

If users and user groups for the Redis and Pgbouncer services are not created during the installation, you must manually create them. Redis and Pgbouncer are dependencies of the Director. For information about how to create these users and user groups, see **Redis and pgbouncer users and user groups not created after an upgrade (https://docs.mandiant.com/home/msv-redis-and-pgbouncer-users-and-user-groups-not-created-after-an-upgrade)**.

1. **Download the installer (https://docs.mandiant.com/home/msv-director-installers)** and then copy it to the system where you want to install it.

   ```
   $ scp FILE_NAME user@IP_ADDRESS:
   ```

   Replace the following:
   - **FILE_NAME**: the name of the Director install file
   - **IP_ADDRESS**: the IP address of the system where you are installing the Director

2. Use SSH to open a command line on the system where you want to install the Director.

   > Use the account that you created in **Configure the Linux Environment to support installation of the Security Validation Actor (https://docs.mandiant.com/home/msv-actor-configure-the-environment)**.

3. Extract the Director `tar.gz` file.

   ```
   $ tar -xvf director_VERSION.tar.gz
   ```

   Replace **VERSION** with the version number of the Director that is part of the install file.

4. Launch the installer using flags. There are two versions of each flag. The Syntax uses the first flag format and Example uses the second. The Full list of Flags table shows both formats, if the flag is required, and if there are any expected values for that flag.

   Syntax:

   ```
   $ cd director_VERSION
   $ sudo ./verodin-director-install --user USERNAME --group GROUP_NAME --interface INTERFACE --repository REP
   OSITORY --dbpassword PASSWORD --checkpoint CHECKPOINT_RESPONSE --no-dot-files NO_DOT_FILES_RESPONS
   E
   $ vrestart
   ```

   Replace the following:

   - **NAME**: The username for the Director to use.
     The Security Validation software needs to run as a named system user in order to have the appropriate permissions and file access. This user should already exist on your system.
   - **GROUP_NAME**: The group the user will be a part of.
     The Security Validation software needs to run as a named system group in order to have the appropriate permissions and file access. This group should already exist on your system.
   - **INTERFACE**: The network interface you want the Director to use.
     The Director listens for connections on a named network interface. These can be seen with the `ifconfig(8)` or `ip(8)` command.

     > You can include up to three interfaces.

   - **REPOSITORY**: Enter either `yum` or `verodin`.
     - **yum**: Getting the dependencies online or through a customer-provided repository

       > Using yum is the preferred method, because it is more in tune with your security policy. For more information, see **Handling Software Dependencies (https://docs.mandiant.com/home/msv-handling-software-dependencies)**.

     - **verodin**: Using the files that are included with the installer.

       > The verodin repository is only valid for CentOS systems.

   - **PASSWORD**: The password of the Director's database.

     > During initial installation, you *must* provide a database password. You are not prompted to specify this password during system updates.

- **CHECKPOINT_RESPONSE**: Enter either `True` or `False`.
  The Director requires specific rpm libraries to run the Checkpoint integration.
- **NO_DOT_FILES_RESPONSE**: Enter either `True` or `False`
  The Director requires an rvm installation at `/usr/local` which can conflict with other ruby applications that rely on their own rvm installation. This flag prevents the creation of dotfiles that will be ingested into user profiles.

  If you are running other Ruby applications on the host where you are installing the Director, set `no_dot_files` to True.

Example:

```
$ cd director_4.10.2.0
$ sudo ./verodin-director-install -u nodeone -g nodeone -i ens160 -r yum -d dbpass -p false --no-dot-files false
$ vrestart
```

To see a full list of the flags, you can type the following command:

```
$ sudo ./verodin-director-install --help
```

5. The installer checks your input and verifies preliminary conditions are satisfied (installing as root, username exists, system requirements are met, and so on). If no issues are found, installation completes. If issues are found, the installer provides messages clearly identifying the issue.

Full list of Flags

Flag format 1	Flag format 2	Required?	Expected values, if any
`--user`	`-u`	✔	
`--interface`	`-i`	✔	
`--repository`	`-r`	✔	`yum` or `verodin`
`--dbpassword`	`-d`	✔	
`--checkpoint`	`-p`	✔	`true` or `false`
`--no-dot-files`	na		`true` or `false`
`--help`	`-h`		

Director Database password info

- If your password uses special characters, you must escape those characters
- After changing the password, you must run `vrestart`

If you are uncertain if a character needs to be escaped, run this command first, adding the escapes where you think they are needed. If it comes out as you expect, you've escaped it correctly.

```
echo This\!sMynew\$P@ssw0rd\!
```

The results of that command would be:

```
This!sMynew$P@ssw0rd!
```

While this method is helpful for syntax, entering plain-text passwords into a terminal history is a security risk. Make sure to clear the terminal history or use a more secure method for sensitive credentials.

[1] The executables include `rar` and `unrar`, `tcpflow`, `Tcprelay`, `tcpprep`, `tcprewrite`, `tcpreplay-edit`, `tcpcapinfo`, and `tcpliveplay`.

3. Install and configure Actor

Follow these steps to install MSV on Google Cloud:

1. **Install the Actor**
2. **Configure the Director**
3. **Register the Actor**

Install the Actor

1. Browse to the extracted install directory and run the following:

   ```
   ls
   ```

   The verodin installer should be visible.

2. Run the installer:

   ```
   sudo ./verodin-actor-install
   ```

   A series of interactive options appear, asking you select the appropriate options, such as local account used (for example, `svc_mandiant`), install location, and so on.

   For Network Actors, one interface is used for management, the other interface is used for testing. Endpoint Actors only use the one interface for management.

3. Select the appropriate options for the current installation. As the installation progresses, status updates appear. You see `OK` next to items as they are completed successfully and an update when the installation process is finished.

   When completed, the network on the VM restarts and the configuration of the Actor VM is complete.

Configure the Director

1. From a web browser, browse to the Director IP address and sign in. See **Validation Director Credentials (https://docs.mandiant.com/home/msv-dir-creds)** for the default username and password.

2. When prompted, create a new password.
3. Accept the agreement and click **Submit**.
4. Upload and apply the license file.

   This file can be uploaded directly from the client that is connected to `https://DIRECTOR_IP`

Register the Actor

This process is the same for both Network and Endpoint Actors.

1. From the Director, go to **Environment > Actors** to register an Actor on the Director.
2. Click **Add Network Actors** or **Add Endpoint Actors** depending on the type you are registering. The Actor then appears as **Unregistered** in the **Pending Actors** table.
3. Under **Actions**, click **more** and then click **Connect**.
4. Enter the IP address of the management interface for the Actor.