Mandiant

DIRECTOR INSTALLATION

To support our customers' various environments, we provide the following ways to install the Director:

- Appliance
  - Local hardware
  - AMI for AWS
  - VHD for Azure
  - VHD for Hyper-V
- Software
  - Install wizard
  - CLI with Flags
  - CLI with an ini file

Local hardware

1. Download the installer from the Mandiant Documentation Portal (**https://docs.mandiant.com**).
2. Import the virtual machine into the existing virtual infrastructure and boot it. This step launches the Director install wizard.
3. After boot completes, a login prompt is displayed. Enter the default username and password noted in **Validation Director Credentials (https://docs.mandiant.com/home/msv-dir-creds)**.
4. After authentication, set up the Network Configuration.

   Remember to use a static IP address.

   ```
   sudo vsetnet
   ```

5. Create a unique password for the Director database using the `vsetdb` command:

   ```
   sudo vsetdb --password NEW_PASSWORD
   ```

   Replace *NEW_PASSWORD* with the password you want to use for your Security Validation database.

   You can also run `vsetdb` using this format:

   ```
   sudo vsetdb -p NEW_PASSWORD
   ```

   If your password uses special characters, you must escape those characters. If you are uncertain if a character needs to be escaped, open another shell and run this command, 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!
   ```

6. Confirm the IP settings have been changed.

```
ifconfig
```

AMI

When you install the Director using an AMI, you go through two steps:

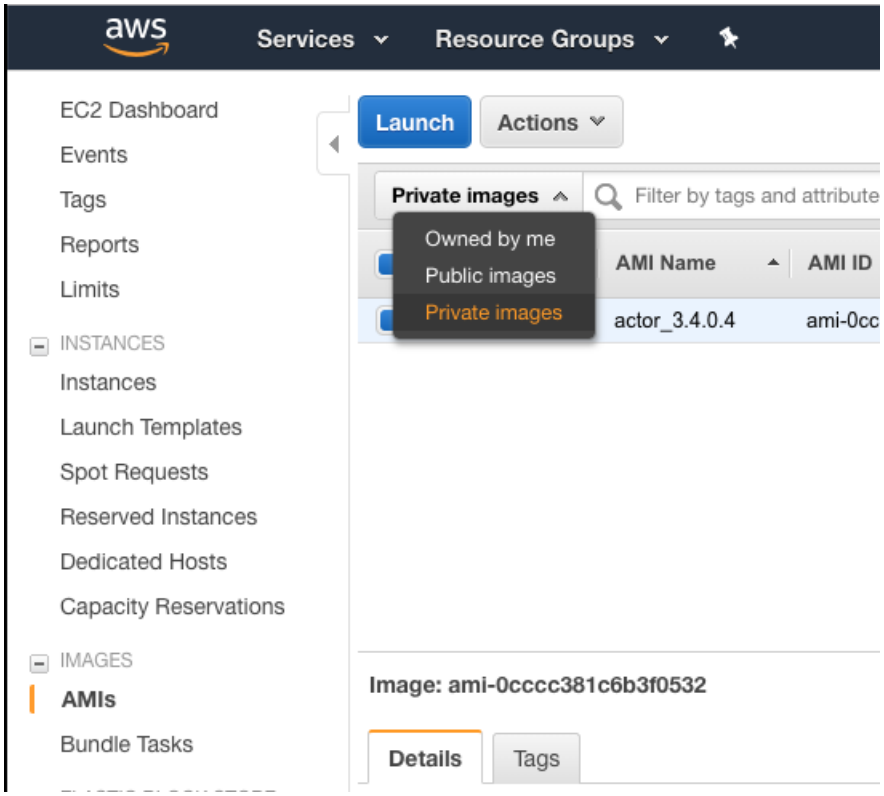
1. Receive the AMI
2. Configure AWS and Install the Director

Receive the AMI

To make the installation as easy as possible, Mandiant sends you the AMI directly in AWS. To notify us that you need the AMI, contact **support (https://www.mandiant.com/support)** (https://mandiant.com/support) and let them know you need the AMI. Include the following information with your request:

- AWS Account number
- Desired Region
- Which component you need (Director or Actor)

The support team provides access to the AMI directly in AWS. Once access is granted, the AMI becomes available in your AWS account console in **AMI > Private Images**.



Private AMI Images in AWS

Configure AWS and Install the Director

1. Launch the AMI.

Don't use **Auto-assign Public IP**. Clear this setting if it's selected.

2. Create and/or associate an Elastic IP.
3. After authentication, set up the Network Configuration.

```
sudo vsetnet
```

`vsetnet` only sets the interface in the Validation Platform config file. `vsetnet` does not configure the OS networking because that step is handled by `cloud-init`.

4. Confirm that the IP settings have been changed.

```
ifconfig
```

5. Note the IPv4 Public IP that was assigned to the Director.
6. Create a unique password for the Director database using the `vsetdb` command:

```
sudo vsetdb --password NEW_PASSWORD
```

Replace *NEW_PASSWORD* with the password you want to use for your Security Validation database.
You can also run `vsetdb` using this format:

```
sudo vsetdb -p NEW_PASSWORD
```

If your password uses special characters, you must escape those characters. If you are uncertain if a character needs to be escaped, open another shell and run this command, 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!
```

7. Restart the Director.

```
vrestart
```

Azure

Installation of the Director requires the following steps:

1. Convert the VHD from Dynamic to Static disks
2. Upload and Deploy the VHD to Azure

Convert the VHD from Dynamic to a Static disk

The Security Validation team provides VHD files for the installation of the Director and the Actor. To upload this to Azure, you must first convert it to Dynamic to Static disks.

Convert your VHD from Dynamic to Static Disks

1. Download the VHD image from the Mandiant Documentation Portal.
2. Extract the archive. It should extract a VHD file.
3. Run one of the following commands, depending on if you are installing a Director or Actor (*VERSION* corresponds to the version of the file that you downloaded):

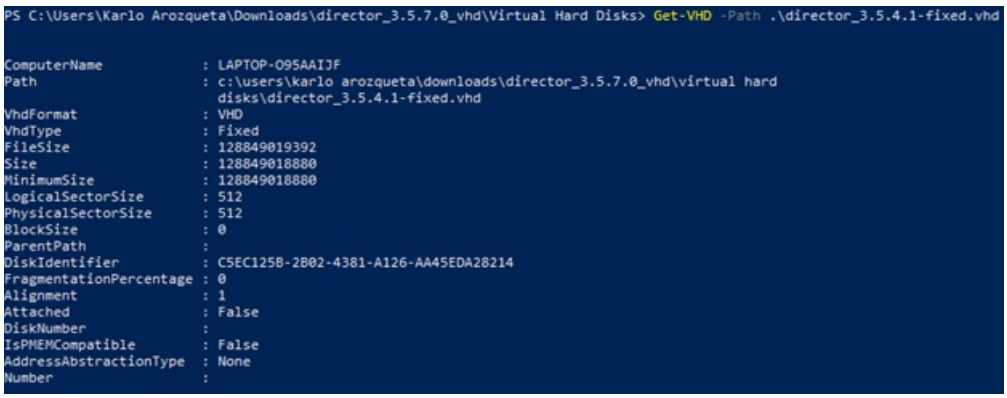
```
Convert-VHD -Path .\director_VERSION.vhd -DestinationPath .\director_VERSION-fixed.vhd -VHDType Fixed
```

```
Convert-VHD -Path .\actor_VERSION.vhd -DestinationPath .\actor_VERSION-fixed.vhd -VHDType Fixed
```

4. You can verify the conversion was successful by running one of the following commands:

```
Get-VHD -Path .\director_VERSION-fixed.vhd
```

```
Get-VHD -Path .\actor_VERSION-fixed.vhd
```



Verify successful conversion

If the conversion was successful, you can upload the VHD to Azure.

Troubleshooting Conversion issues

Some errors you might see when you try the conversion include:

```
Convert-VHD : You do not have the required permission to complete this task. Contact the administrator of the
authorization policy
for the computer
```

- This error means you are running PowerShell under the User context. Re-launch as an administrator.

```
Convert-VHD : The term 'Convert-VHD' is not recognized as the name of a cmdlet, function, script file, or operab
le program.
Check the spelling of the name, or if a path was included, verify that the path is correct and try again.
At line:1 char:1
```

- This error indicates that certain Windows features must be enabled for the command to process correctly. Proceed to the next steps to install the required services.

If you receive either of these errors, you need specific Hyper-V PowerShell tools to manage the Hyper-V. These can be installed by using PowerShell or through the GUI.

Add the Hyper-V PowerShell using PowerShell

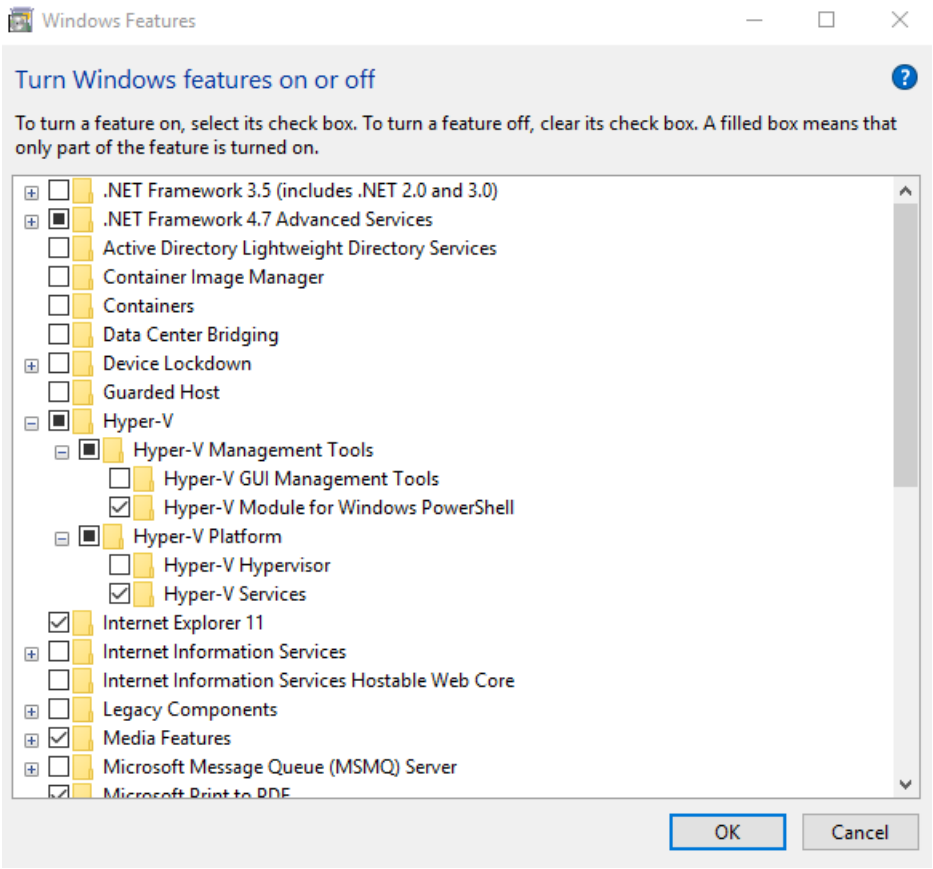
This procedure requires an internet connection.

While running PowerShell as an administrator, run the following command:

```
Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V -All
```

Add the Hyper-V PowerShell using the web interface

1. Open the Windows Control Panel.
   Select **Programs** (or **Programs and features**, based on your version of Windows).
   Select **Turn windows features on or off**.
2. Scroll down to Hyper-V and expand the options. Select the following Options:
   - Under **Hyper-V Management tools**, check the box for **Hyper-V Module for Windows PowerShell**.
   - Under **To add the Hyper-V PowerShell using PowerShell** , check the box for **Hyper-V Services**.



Hyper-V Windows Features

Sources

- **https://superuser.com/questions/1307441/powershell-resize-vhd-is-not-recognized-as-the-name-of-a-cmdlet**
- **https://social.technet.microsoft.com/Forums/windowsserver/en-US/cdd725d6-7f7b-4022-a19e-f7d242ba514b/convert-dynamic-to-fixed-size-vhd?forum=winserverhyperv**
- **https://www.altaro.com/hyper-v/gathering-vhd-info-get-vhd-powershell/**

Upload and Deploy the VHD to Azure

After you convert the Director or Actor VHD from Dynamic to Static disks, you can upload it to Azure. This process can be done using the GUI or from the command line. This process requires the following steps:

- Install the PowerShell cmd-let module
- Upload the VHD
- Convert the page blob to a managed disk

Install the PowerShell cmd-let module

1. Launch Windows PowerShell as an administrator.

   > If you can only launch as a user, you can append `-Scope CurrentUser` to the commands in step 2 to install as a user.

2. Use the following command to install the PowerShell cmd-let module:

   ```
   Install-Module -Name Az.Compute -Force
   ```

Upload the VHD from the GUI

Reference the following article: **https://aidanfinn.com/?p=20441**.

> If you prefer a user interface, install Azure Storage Explorer. This interface allows you to drag-and-drop uploads and downloads to Azure storage containers.

Upload the VHD using the Command line

1. Run the following command to log in in to your Azure account:

   ```
   Login-AzAccount
   ```

   This step launches a web browser and prompts you to log in to Azure.

2. In Azure, verify you have the following prerequisites set up:
   - A resource group created
   - A storage blob
   - A container in that blob to upload to

3. Upload the VHD by running the following command:

   ```
   Add-AzVhd -ResourceGroupName 'myResourceGroup' -Destination 'https://[myStorageAccount].blob.core.windows.net/[container]/[name.vhd]' -LocalFilePath '[path.vhd]'
   ```

   > Ensure the `--Blob-Type` is `PageBlob`.

   This step scans the static VHD and determines the free space (zeros) on the disk (expanded blank space). This step then uploads the VHD as a page storage blob. The upload size should be equivalent to the pre-converted disk size (dynamic to static), which as of March 2020, is approximately 11GB. Once uploaded, you must convert the page storage blob to a managed disk (next steps), which requires the full 160GB of the expanded disk.

4. If necessary, log back into the Azure instance using the following command:

   ```
   Login-AzAccount
   ```

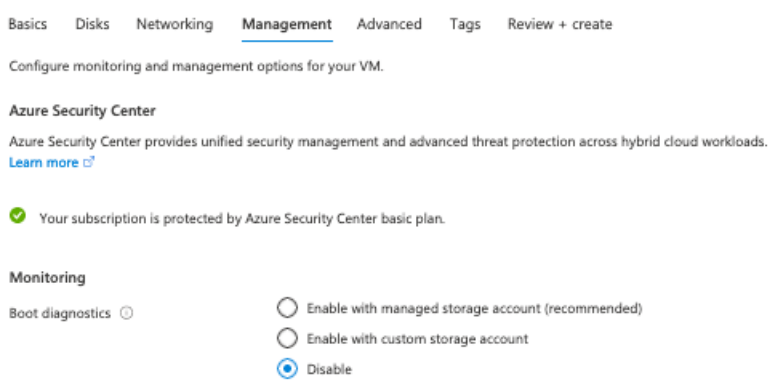
Convert the page storage blob to a managed disk

A sample of the script is provided as follows. You can also access the script from **https://docs.microsoft.com/en-us/azure/virtual-machines/scripts/virtual-machines-windows-powershell-sample-create-managed-disk-from-vhd**.

```
#Provide the subscription Id where Managed Disks will be created
$subscriptionId = 'yourSubscriptionId'

#Provide the name of your resource group where Managed Disks will be created.
$resourceGroupName ='yourResourceGroupName'

#Provide the name of the Managed Disk you are creating
$diskName = 'yourDiskName'

#Provide the size of the disks in GB. It should be greater than the VHD file size. (160GB)
$diskSize = '160'

#Provide the storage type for Managed Disk. Premium_LRS or Standard_LRS.
$storageType = 'Premium_LRS'

#Provide the Azure region (e.g. westus) where Managed Disk will be located.
#This location should be same as the storage account where VHD file is stored
#Get all the Azure location using command below:
#Get-AzLocation
$location = 'westus'

#Provide the URI of the VHD file (page blob) in a storage account. Please not that this is NOT the SAS URI of the stora
ge container where VHD file is stored.
#e.g. https://contosostorageaccount1.blob.core.windows.net/vhds/contosovhd123.vhd
#Note: VHD file can be deleted as soon as Managed Disk is created.
$sourceVHDURI = 'https://contosostorageaccount1.blob.core.windows.net/vhds/contosovhd123.vhd'

#Provide the resource Id of the storage account where VHD file is stored.
#e.g. /subscriptions/6472s1g8-h217-446b-b509-314e17e1efb0/resourceGroups/MDDemo/providers/Microsoft.Storage/
storageAccounts/contosostorageaccount
#This is an optional parameter if you are creating managed disk in the same subscription
$storageAccountId = '/subscriptions/yourSubscriptionId/resourceGroups/yourResourceGroupName/providers/Microsoft
.Storage/storageAccounts/yourStorageAccountName'

#Set the context to the subscription Id where Managed Disk will be created
Select-AzSubscription -SubscriptionId $SubscriptionId

$diskConfig = New-AzDiskConfig -AccountType $storageType -Location $location -CreateOption Import -StorageAccou
ntId $storageAccountId -SourceUri $sourceVHDURI -OsType Linux

New-AzDisk -Disk $diskConfig -ResourceGroupName $resourceGroupName -DiskName $diskName
```

Deploy the VHD

1. Deploy the VHD by building a new Virtual Machine with the newly staged **Managed Disk**.
   Ensure Boot monitoring is set to **Disable** on the virtual machine. Not setting this properly may result in Azure warning of an incomplete boot and a triggered automatic reboot of the VM.



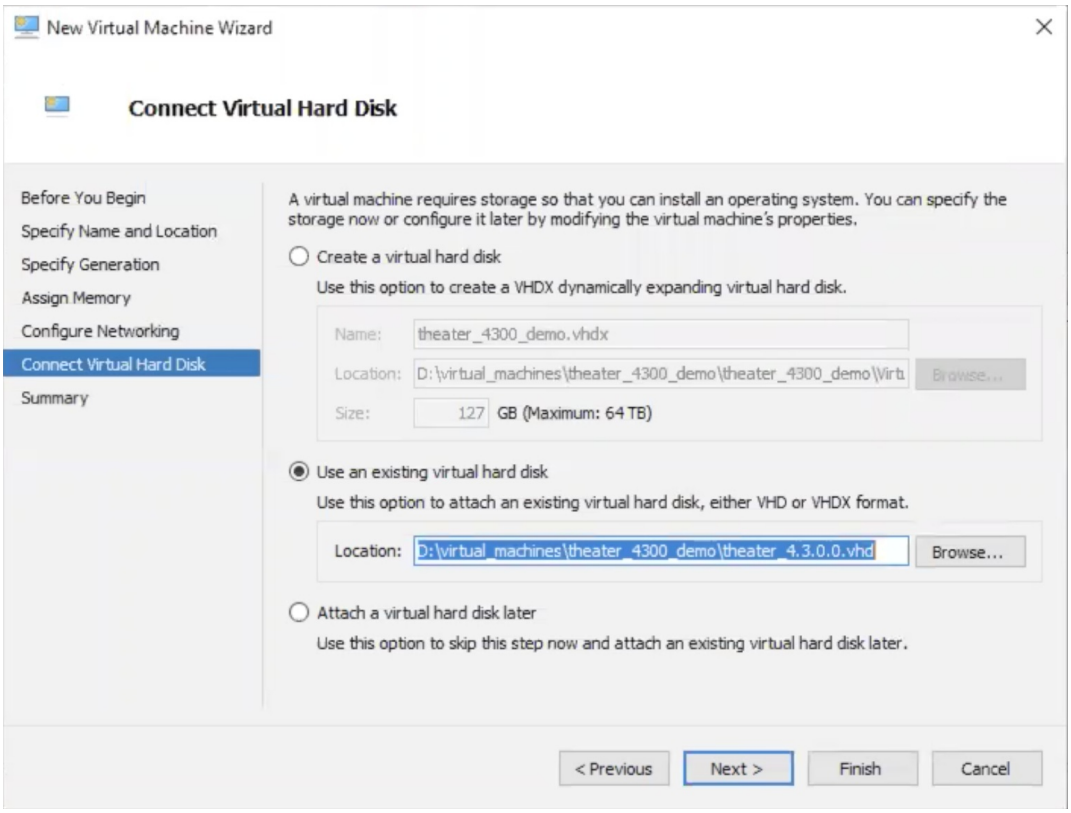
2. For an Actor that will have multiple interfaces, you must to shutdown the VM after deployment and add the optional TEST and MONITOR network interfaces.
    - These interfaces can be on the same subnet as the original MGMT interface.
    - As stated in the Actor operational documentation, the TEST and MONITOR interfaces should be a static IP and routable with a default gateway. A static public IP address is optional ONLY if testing is contained within customer private address space or a common outbound routed gateway is used to reach the internet.
    - After attaching the interfaces, reboot the VM and become familiar with which interfaces is the original MGMT interface and new TEST and MONITOR interfaces before proceeding to run `vsetnet`.
3. For Mandiant-provided VHD images, accessing the newly built VM is performed through an SSH connection that uses the default initial login on the IP address that was provided by the Azure installation.
    - Ensure the default account is changed soon after installation and is set to an appropriately complex combination.
    - Apply firewall rules in your Azure subscription that limit inbound connections to the MGMT interface of the newly deployed Actor. See documentation on required ports for Actor operations.

Hyper-V

1. Download the VHD image from the Mandiant Documentation Portal (**https://docs.mandiant.com**).
2. Extract the VHD and then copy it to your desired location. If you have a standard virtual machines folder, we suggest you use that.
3. Create the Virtual Machine in Hyper-V.
    a. Click **New** > **Virtual Machine**.
    b. Click **Next**.
    c. Enter a **Name** for your Actor virtual machine and (optional) select the **Location** where the virtual machine should be stored. Then click **Next**.
    d. Specify **Generation**. Generation 1 is recommended. Then click **Next**.
    e. Assign Memory. **16384 mb** is recommended. (For additional details, see **Director System Requirements (https://docs.mandiant.com/home/msv-director-system-requirements)**.) Then click **Next**.

    > Do not select Use Dynamic Memory for this virtual machine.

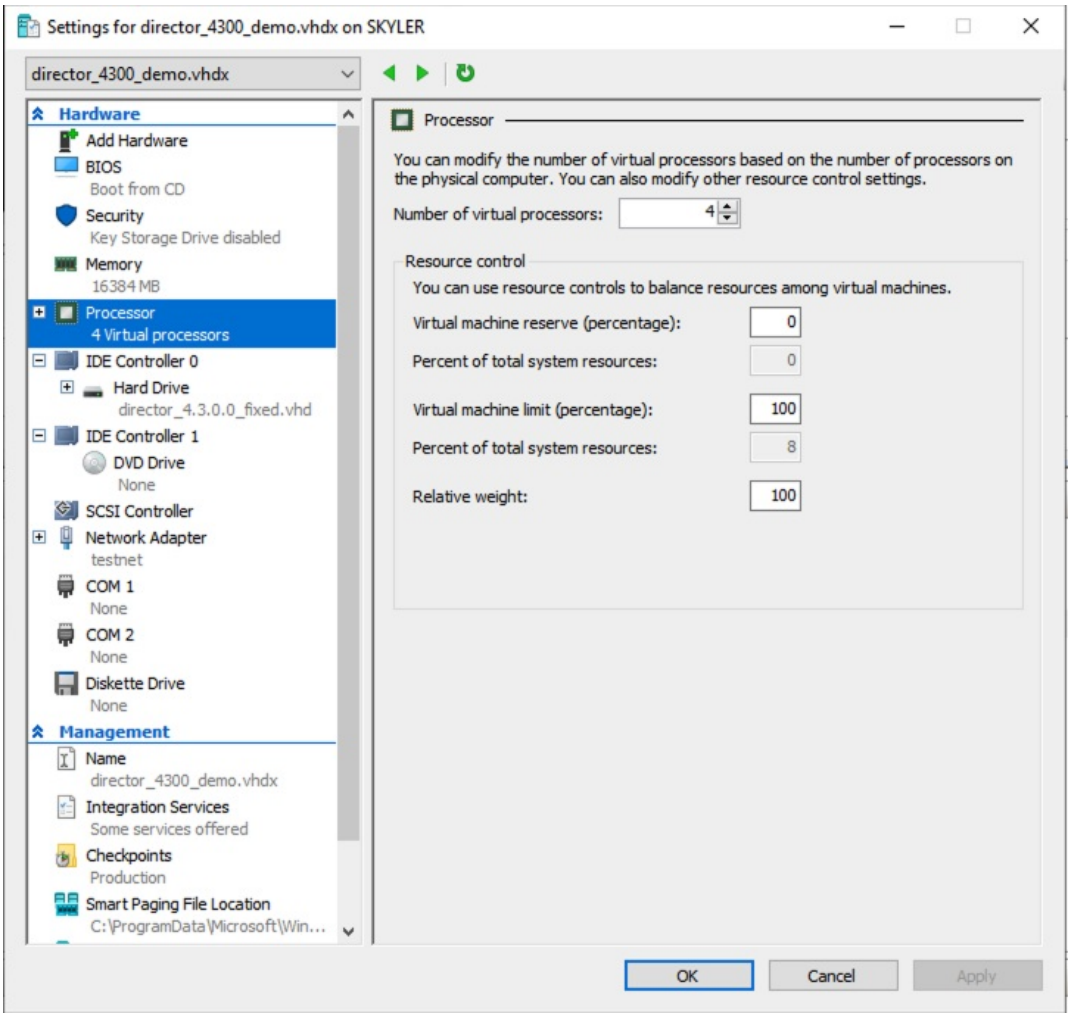
    f. Select your network **Connection,** then click **Next**.
    g. Choose **Use an existing virtual hard disk** , navigate to the disk's location, and then click **Next**.



Hyper-V example: Connecting a Virtual Hard Disk to a new Virtual Machine

h. Verify everything is configured as expected and then click **Finish**. The virtual machine is displayed and is selected in the Virtual Machines list.

4. Update the Virtual Machine's Processor info.
   a. Select your Director Virtual Machine and then click **Settings**.
   b. Click **Processor**, adjust Number of virtual processors to **4**, click **Apply**, and click **OK**.



Hyper-V: Adding processors to a virtual machine

5. Expose the Virtualization Extensions for your VM.
    a. Open a Windows PowerShell Admin window.
    b. Run the following command:

```
Set-VMProcessor <VMName> -ExposeVirtualizationExtensions $true
```

6. Start the Virtual Machine by selecting the VM in Hyper-V Manager and clicking **Connect**.
7. After authentication, set up the Network Configuration.
    a. Boot the installed image and open a console to the image through the virtual infrastructure.
    b. After the boot, a login prompt is displayed. Enter the default operating system username and password, and update if necessary.
    c. Setup the network configuration.

> Remember to use a static IP address.

```
sudo vsetnet
```

8. Confirm the IP settings have been changed.

```
ifconfig
```

9. Create a unique password for the Director database using the `vsetdb` command:

```
sudo vsetdb --password NEW_PASSWORD
```

Replace *NEW_PASSWORD* with the password you want to use for your Security Validation database.
You can also run `vsetdb` using this format:

```
sudo vsetdb -p NEW_PASSWORD
```

If your password uses special characters, you must escape those characters. If you are uncertain if a character needs to be escaped, open another shell and run this command, 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!
```

10. Restart the Director.

```
vrestart
```

Install Wizard

Overview

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. Change to the newly extracted Director directory, and run the installer.

   If there is a space in the path, the installation fails.

   ```
   $ cd director_VERSION
   $ sudo ./verodin-director-install
   ```

   The Wizard starts, walking you through the installation. This document includes much, but not all, of what you see during installation.

   ```
   *** Verodin Director Installer
   In order to start the installation, this program will ask you some questions about how your system is configured
   and how you want to configure the Verodin software installation.
   Some questions will have a [default value] which is shown in brackets.
   If you want to accept the default answer, just press Enter when prompted.
   You can cancel the installation process at any time with Ctrl-C.
   ```

   a. Answer the user question.

      Reminder: this is the user account you created (or decided to use with the Validation Platform) that has sudo privileges.

```
user
The Verodin 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.
Verodin Director and its services should run as which user [apache]:
```

b. Answer the group question.

```
group
The Verodin 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.
Verodin Director and its services should run as which group
default:
```

c. Answer the interface question.

```
interface
The Verodin Director will listen for connections on a named network interface. These can be seen with the
ifconfig(8) or ip(8) command.
Which network interface should Director communicate through:
```

d. RHEL7 only: Answer the repository question.

The offline repository option is only available if you are using an MSV release that is earlier than 4.14.4.0. This option is not available for 4.14.4.0 onward.

The preferred method is to get the files online using yum because it will be more in tune with your security policy. For more information, see **Handling Software Dependencies (https://docs.mandiant.com/home/msv-handling-software-dependencies)**.

```
repository
This installer depends on additional system packages (for example, the bzip compression library, or the a
pache webserver). If you have an internet connection and your security policy allows installation package
s from Red Hat repositories, reply "yum". You may also have configured this machine to install dependenc
ies from a local repository under your control, if so the answer is also "yum". If you have access to neither
the internet nor a private repository, you can reply "verodin" and the dependencies will be satisfied from
packages that we have included for unconnected systems.
Where should the installer obtain needed dependencies [yum]:
```

e. Answer the database password question.

Create a unique password for Director to database communication, which the installer will set. Retain a record of this password for troubleshooting purposes.

```
dbpassword
The Verodin Director requires a database password so it can configure and communicate with postgresql.
Password for the Director database:
```

If your password uses special characters, you must escape those characters. If you are uncertain if a character needs to be escaped, open another shell and run this command, 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!
```

f. Answer the Check Point Integration question.

```
checkpoint_integration
Verodin Director requires specific rpm libraries to run the checkpoint integration. .
Checkpoint Integration RPM Dependencies (True/False)[True]:
```

g. Specify if the Ruby virtual machine will create dotfiles.

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

```
no_dot_files
Verodin Director requires an rvm installation at /usr/local which can conflict with other ruby applications which rely on their own rvm installation. This flag prevents the creation of dotfiles that will be ingested into user profiles.
Do not create auto dotfiles for ruby virtual machine [False]:
```

5. The installer then checks your input and verify preliminary conditions are satisfied. Possible outcomes of this check include:
   - **Issue Found**: Interface not valid (example).

```
The environment check for interface has failed.
The network inferface interface does not exist on this system. You can list the available interfaces with the "ip link" command.
There is an unrecoverable error during installation. Your configuration and log files will be saved so that you can provide them to Verodin support. These files are human readable text and should not contain any security information.
The information gathered is saved in the file verodin-director-install.ini
Log information is contained in verodin-director-install.log.
```

   - **No Issue Found**: The installation continues, verifying requirements and packages are installed. The following is a sample of what is displayed:

```
checking is_root_user... ok
checking is_user nodeone... ok
checking interface_exists... ok
checking interface is up... ok
checking cpu_count_director... ok
checking memory_size... ok
checking that package yum is installed...
```

> The installation is a long-running process, announcing each dependency installation to provide status. Allow at least 10 minutes for installation to complete.

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

CLI with Flags

Overview

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!
```

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

CLI with an ini file

Overview

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. Create the configuration file, `my-director.ini` and open it.

```
$ cd director_VERSION
$ cp example.ini my-director.ini
$ vi my-director.ini
```

5. Update the `my-director.ini` configuration file you just created, editing the options as instructed by the comments in that file.

```
[options]
user = NAME
group = GROUP_NAME
interface = INTERFACE
repository = REPOSITORY
dbpassword = PASSWORD
checkpoint = CHECKPOINT_RESPONSE
no_dot_files = NO_DOT_FILE_RESPONSE
```

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.

6. Launch the installer with the configuration file:

```
$ sudo ./verodin-director-install --config-file my-director.ini
```

7. The installer then checks the user input and verifies that preliminary conditions are satisfied (installing as root, username exists, system requirements are met, and so on). If there are no issues, installation occurs; if there are issues, you are presented with user-friendly messages clearly identifying the issue.

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