VM Monitoring on Azure

paloaltonetworks.com/documentation

techDOCS

Contact Information

Corporate Headquarters:
Palo Alto Networks
3000 Tannery Way
Santa Clara, CA 95054
www.paloaltonetworks.com/company/contact-support

About the Documentation

- For the most recent version of this guide or for access to related documentation, visit the Technical Documentation portal www.paloaltonetworks.com/documentation.
- To search for a specific topic, go to our search page www.paloaltonetworks.com/documentation/document-search.html.
- Have feedback or questions for us? Leave a comment on any page in the portal, or write to us at documentation@paloaltonetworks.com.

Copyright

Last Revised

August 14, 2018

VM Monitoring on Azure

VM Monitoring of Microsoft® Azure® resources enables you to dynamically update security policy rules to consistently enforce Security policy across all assets deployed within your Azure subscription. The VM Monitoring solution on Azure uses a VM Monitoring script that runs on a virtual machine within the Azure public cloud. This script collects the IP address-to-tag mapping for all your Azure assets and uses the API to push the VM information to your Palo Alto Networks® firewall(s).

About VM Monitoring on Azure

As you deploy or terminate virtual machines in the Azure public cloud, you can use the Azure plugin on Panorama to or the VM Monitoring script for Azure to consistently enforce security policy rules on these workloads. If you are using Panorama 8.1.3 or later to manage your firewalls, use the Azure plugin on Panoramato use Panorama as the anchor that connects to your Azure subscriptions, retrieves VM information on the Azure workloads and register them to the firewalls that you configure for notification. If you do not have Panorama or are using an earlier version of Panorama, use the VM Monitoring script to monitor your Azure workloads.

The VM Monitoring script runs on a virtual machine in the Azure public cloud environment. The operating system of the virtual machine that the script runs on, must be Red Hat Enterprise Linux (RHEL) 7.4 with Python version 2.7.5. The script collects the IP address to tag mappings for all your Azure assets and uses the Azure and PAN-OS APIs to register the VM information—IP address to tag mapping—on the firewalls you specify. You can specify one or more virtual systems on the firewall to which you want to register the VM information. The script can register the information on firewall running running PAN-OS 8.0 or later.

The solution, which is posted on GitHub, is released under the community support policy of Palo Alto Networks. The GitHub repository includes two files:

- Parameters file—The parameters file is named **parameters.json**. This file allows you to specify details on your Azure subscription, how to authenticate to it, which Azure resources to monitor, and to which firewalls you want to publish the IP address to tag mapping information that the script collects.
- VM Monitoring script—The VM Monitoring script uses Python and is named **run.py**. This script collects the IP address-to-tag mapping information for the Azure deployment that you want to monitor and pushes the information to the specified firewalls using the PAN-OS API. The script registers new IP address to tag mapping on the firewalls, and unregisters IP addresses and tags that are deprovisioned in your Azure deployment from the firewall. To prevent overwriting the VM information, make sure that a virtual system receives IP address and tag information from one instance of the script only.

  You must use the management interface on the firewall to communicate with the virtual machine (RHEL instance) that runs the script.

  The script generates 2 sets of log files. The audit log includes all messages, including the API calls and the responses. The error log includes error messages only. The log files require about 30 GB on the hard disk of the virtual machine. The log file is rotated at 1 GB, and a maximum of 30 logs files are stored on disk. If you want persistent log storage, make sure to export or archive the log files to an external location.

You can deploy one or more instances of the virtual machine (RHEL instance) to run the VM Monitoring script that monitors your Azure subscription. Because the script is designed to execute as a cron task, the script executes only when it detects that the process isn't already running. Therefore, a new cron task does not execute when one is running, and you cannot have multiple instances of the VM Monitoring script run on a single virtual machine (RHEL instance).

Gather the Resources Required for VM Monitoring on Azure

The following table lists the resources needed to deploy this VM Monitoring solution for Microsoft® Azure®.

What you need	Description
□ System Requirements for the virtual machine. *Only one instance of the VM Monitoring script can run on a virtual machine instance.*	The VM Monitoring solution on Azure requires a system with: • **Operating System**—Red Hat Enterprise Linux (RHEL) 7.4 • **Python Version**—2.7.5 • **Disk Size**—60GB minimum
□ Set up the Active Directory application and a Service Principal to enable API access for the VM Monitoring script.	Because the VM Monitoring script uses the Azure API to collect the attributes for your Azure deployment, you need to set up an Active Directory application and a Service Principal to assign permissions. When you follow the instructions in the preceding link, you must assign an IAM role with a minimum privilege of **reader** when prompted to **Assign application to role**. The workflow will provide you with various keys and IDs that are required to generate an Azure Bearer Token used in the header of the API call. Ensure that you collect the following information, which you must enter as input in the parameters.json file: • **Application ID** • **Authentication Key**—Make sure to jot down this secret key. You cannot view this key again. • **Directory ID** • **Subscription ID**

□ Collect the details required for the parameters.json file that the script invokes to monitor your Azure deployment.
`{"parameters":{"clientId":{"value":"e12a3fb1-`
`cef2-0000-abf8-7a9cee0dd55f"},`
`"clientSecret":{"value":"jEWXJcNswGWv9VmpJCR80S`
`2GQl/eDQq3W6Yu7yjN2/c="},`
`"tenantId":{"value":"77a9116e-`
`edcc-44b6-84c4-4f19fdda335b"},`
	You must have the following information to fill out the parameters.json file:
	• **Client ID**—The Application ID that you copied earlier.
	• **Secret Key**—The authentication key you copied earlier when you set up the Active Directory application. To to log in as the application, the key

What you need	Description

`"subscriptionId":{"value":"0123402e-4559-4b1a-`
`b645-92fa1234f4b8"},`
`"targetIps":{"value":"172.30.161.201,`
`172.30.161.202"},`
`"resourceGroupName":{"value":"vmscript5RG",`
`"vnetName":{"value": "vpn5vnet5"},`
`  "targetApiKeys":{"value":"LUFRPT14MW5xOEo`
`1R09KVlBZNnpnemh0VHRBOWl6TGM9bXcwM`
`JHUGVhRlNiY0dCR0srNERUQT09,00000000`
`000ZNnpnemh0VHRBOWl6TGM9bXcwM3JHUG`
`hRlNiY0dCR0sra"},`
`"targetVsys":{"value": "vsys1,vsys3"}}}`
	value with the Application ID are required.
	• **Tenant ID**—The Directory ID you copied earlier.
	• **Azure Subscription ID**—The Azure subscription you want to monitor.
	• **Target IPs**—A comma separated list of IP addresses of the next-gen firewalls to which you want to register the IP address-to-tag mapping. You can then configure the firewalls that receive the VM information to enforce policy.
	If the firewalls are in an HA configuration, include the IP address for both HA peers. The script will register tags to the active peer only.
	• **Vsys**—The virtual system that you want to set as the destination for registering the IP address-to-tag mapping that the script retrieves.
	• **Resource Group Name**—(Optional, but recommended if you have overlapping IP addresses across your Resource groups and VNets within your subscription) Enter (only) one resource group name that you want to monitor.
	• **VNet Name**—(Optional, but recommended if you have overlapping IP addresses in your resource group) Enter the name of a single VNet that you want to monitor.
	• **API Keys**—Comma separated list of the API keys for the administrative user account on each firewall.
	For all comma separated values—Target IPs, API keys, and Vsys—you should not have space between the comma and the value.

Set Up VM Monitoring on Azure

This workflow guides you through deployment of the RHEL virtual machine and configuration of the VM monitoring script to run as a cron task on this RHEL instance so that the script can collect the virtual machine attributes within your Azure subscription. You can then use this information to proactively enforce policy using your Palo Alto Networks firewalls.

There is no default interval or frequency at which the script will execute, so you must configure the script to run at a specific interval at which the script collects the IP address-to-tag mapping and publishes the information to a target virtual system on your next-gen firewalls. The script registers new IP addresses and associated tags on the firewall, and unregisters IP addresses and tags for assets that were deleted or terminated within your Azure environment.

STEP 1 | Make sure that you first Gather the Resources Required for VM Monitoring on Azure.

STEP 2 | Deploy a Red Hat Enterprise Linux 7.4 OS with at least 60GB hard disk space on the Azure public cloud.

The virtual machine must have network connectivity to the management interface of the firewalls to which you are registering the IP address-to-tag information.

STEP 3 | Use an SSH client to log in to the virtual machine and verify the python version with the command **python -V**.

Authenticate to the RHEL virtual machine using the option —password or SSH key— you selected when deploying the instance.

STEP 4 | Copy the files from the GitHub repository to the virtual machine.

The VM Monitoring solution includes two files— parameters.json and run.py.

```
git clone https://github.com/PaloAltoNetworks/azure-vm-monitoring
```

STEP 5 | Edit the parameters.json file and specify the resources you want to monitor within your Azure subscription.

```
vi parameters.json
```

STEP 6 | Set up the cron task to run the VM Monitoring script at a specified frequency.

The minimum frequency you can set is one minute. The amount of time the script takes to retrieve the IP address-to-tag information in your environment and register it on the firewall varies based on the number of virtual machines in your deployment.

1. To set up the cron task, enter the following command:

   ```
   sudo crontab -e
   ```

   This will open up an editor where you can enter the interval and specify the absolute path for the directory in which to save the log files. For example:

   ```
   */5 * * * * /usr/bin/python/home/vmMonitoring/run.py -f
   /home/vmMonitoring/parameters.json -l /vmagentlogs
   ```

2. Verify that the cron task is set up properly with the command **sudo crontab -l**

To execute the VM Monitoring script on demand, use the command ***python run.py -f parameters.json -l <log-directory>****, where log directory is the absolute path where you want to save the log files.*

STEP 7 | Open the audit log file to confirm that the script was executed successfully and to view the IP address-to-tag mapping that it retrieved.

vi ***<log-directory>*****/audit.log**

```
</entry><count>7</count></result></response>
2018-03-20 17:24:31.822 +0000 VM Monitoring log INFO: : Get Tags: retrieved
 7 tags
2018-03-20 17:24:31.822 +0000 VM Monitoring log INFO: : Get Tags: Retrieved
 total of 7 tags
2018-03-20 17:24:32.167 +0000 VM Monitoring log INFO: : Get Tags: <response
 status="success"><result>Session target vsys changed to none</result></
response>
2018-03-20 17:24:32.168 +0000 VM Monitoring log INFO: : current:
 ['10.155.1.1', '10.155.1.2', '10.155.1.3', '10.155.2.1', '10.155.2.2',
 '10.155.3.3', '10.155.3.4']
2018-03-20 17:24:32.168 +0000 VM Monitoring log INFO: : new: ['10.155.1.1',
 '10.155.1.2', '10.155.1.6', '10.155.2.1', '10.155.2.2', '10.155.3.5',
 '10.155.3.6']
2018-03-20 17:24:32.168 +0000 VM Monitoring log INFO: : Script completed
 normally.
```

STEP 8 | Log in to the CLI on the firewall and verify that you can view the IP address and tags that the script published.

You can quickly confirm that the registered VM count on the firewall matches the audit log:

On a hardware-based firewall, you must specify the target virtual system on which you are registering the VM information using the command `admin@PA5000>`**`set system setting target-vsys vsys1`**.

```
 Session target vsys changed to vsys1
```

`admin@PA5000vsys1>`**`show object registered-ip all`**

```
registered IP                          Tags
10.155.2.5                          #"azure-tag.vm-name.vrpn5server"
                                    "azure-tag.resource-
group.vrpn5RG                                         "azure-
tag.subnet.vrpn5Untrust"                                           "azure-
tag.vnet.vrpn5vnet0"
                                    "azure-tag.region.eastus2"
                                    "azure-tag.vm-size.Standard_D2s_v3
                                    "azure-tag.os-type.Linux"
                                    "azure-tag.os-publisher.Canonical"
                                    "azure-tag.os-offer.UbuntuServer"
                                    "azure-tag.os-sku.16.04-LTS"
```

STEP 9 | Set up Dynamic Address Groups and use them inSecurity policy.

Attributes Monitored on Azure

The VM Monitoring script allows you to gather the following set of metadata elements or attributes on the virtual machines in your Microsoft® Azure® deployment.

Attributes Monitored	Example
VM Name	azure-tag.vm-name.web_server1
VM Size	azure-tag.vm-size.standard_ds2_v2
OS Type	azure-tag.os-type.Linux
OS Publisher	azure-tag.os-publisher.Canonical
OS Offer	azure-tag.os-offer.UbuntuServer
OS SKU	azure-tag.os-sku.14.04.5-LTS
Subnet	azure-tag.subnet.webtier
VNet	azure-tag.vnet.untrustnet
Azure Region	azure-tag.region.east-us
Resource Group Name	azure-tag.resource-group.myResourceGroup
User Defined Tags	azure-tag.mytag.value