# Input bindings are passed in via param block.

param($Timer)

# Get the current universal time in the default string format.

$currentUTCtime = (Get-Date).ToUniversalTime()

# The 'IsPastDue' property is 'true' when the current function invocation is later than scheduled.

if ($Timer.IsPastDue) {

Write-Host "PowerShell timer is running late!"

}

# Write an information log with the current time.

Write-Host "PowerShell timer trigger function ran! TIME: $currentUTCtime"

$FunctionName = 'TimerTrigger'

$ModuleName = 'MSOnline'

$ModuleVersion = '1.1.183.57'

$username = $Env:user

$pw = $Env:password

#import PS module

$PSModulePath = "D:\home\site\wwwroot\$FunctionName\bin\$ModuleName\$ModuleVersion\$ModuleName.psd1"

$res = "D:\home\site\wwwroot\$FunctionName\bin"

Import-module $PSModulePath

# Build Credentials

$keypath = "D:\home\site\wwwroot\$FunctionName\bin\keys\PassEncryptKey.key"

$secpassword = $pw | ConvertTo-SecureString -Key (Get-Content $keypath)

$credential = New-Object System.Management.Automation.PSCredential ($username, $secpassword)

Connect-ExchangeOnline -Credential $credential

#Modify the values for the following variables to configure the audit log search.

$outputFile = (Get-Date).tostring("dd-MM-yyyy-hh-mm-ss")

[DateTime]$start = [DateTime]::UtcNow.AddDays(-2)

[DateTime]$end = [DateTime]::UtcNow

$record = "AzureActiveDirectory"

$resultSize = 100

$intervalMinutes = 360

#Start script

[DateTime]$currentStart = $start

[DateTime]$currentEnd = $start

Write-Host "Retrieving audit records for the date range between $($start) and $($end), RecordType=$record, ResultsSize=$resultSize"

$totalCount = 0

while ($true) {

$currentEnd = $currentStart.AddMinutes($intervalMinutes)

if ($currentEnd -gt $end) {

$currentEnd = $end

}

if ($currentStart -eq $currentEnd) {

break

}

$sessionID = [Guid]::NewGuid().ToString() + "\_" + "ExtractLogs" + (Get-Date).ToString("yyyyMMddHHmmssfff")

Write-Host "Retrieving audit records for activities performed between $($currentStart) and $($currentEnd)"

$currentCount = 0

#Import-Module Search-UnifiedAuditLog

$sw = [Diagnostics.StopWatch]::StartNew()

do {

$results = Search-UnifiedAuditLog -StartDate $currentStart -EndDate $currentEnd -RecordType $record -SessionId $sessionID -SessionCommand ReturnLargeSet -ResultSize $resultSize

#$results | Select-Object -Property AuditData | Export-csv $outputFile -NoTypeInformation

if (($results | Measure-Object).Count -ne 0) {

$results | export-csv $outputFile -Append -NoTypeInformation

$outputFile | ConvertTo-Json

$currentTotal = $results[0].ResultCount

$totalCount += $results.Count

$currentCount += $results.Count

if ($currentTotal -eq $results[$results.Count - 1].ResultIndex) {

$message = "INFO: Successfully retrieved $($currentTotal) audit records for the current time range. Moving on!"

Write-Host "Successfully retrieved $($currentTotal) audit records for the current time range. Moving on to the next interval." -foregroundColor Yellow

""

break

}

}

}

while (($results | Measure-Object).Count -ne 0)

$currentStart = $currentEnd

}

Write-Host "Script complete! Finished retrieving audit records for the date range between $($start) and $($end). Total count: $totalCount" -foregroundColor Green

#Disconnect-ExchangeOnline -Confirm:$false

$Body = @{ Name = $outputFile }

Invoke-WebRequest -URI https://auditlogsapp.azurewebsites.net/api/HttpTrigger1?code=QewXWhMoKr0Um0Qs9K1wrPU9QSRVuG11vLRQAtslJQhu8T5BbfaqBA== -Body $Body