如何在Proxy环境下正确使用PowerShell工具发送web http请求

PowerShell是一款非常实用的命令行界面和脚本编辑工具，同时我们也可以通过该工具来管理几乎所有azure上的资源。它可以被用来执行各种任务，其中包括脚本程序执行和命令提示行交互。

然而在一定的特殊条件下，比如一些客户是在有代理(Proxy)的环境下工作，而这些代理需要一些基本的验证，当这些客户使用PowerShell时可能会遇到一些问题导致无法正常使用PowerShell工具。这篇文档就是讨论这种情况下，我们应该做哪些配置的调整，从而使得我们可以成功的使用PowerShell工具。

**问题症状：**

当我们在代理(Proxy)的环境下使用PowerShell执行一些不涉及到http请求的命令时正常，比如get-azurezccount；然而涉及到http请求的命令会报错，比如：get-azureVM/get-osversions -subscriptionId \*\*\*\* -certificate (get-item cert:\CurrentUser\MY\\*\*\*\*\*\*)。

一些常见的报错信息如下：

PS C:\> Get-AzureVM -ServiceName "MySvc1"  
Get-AzureVM : An error occurred while sending the request.  
At line:1 char:1  
+ Get-AzureVM -ServiceName "MySvc1"  
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~  
    + CategoryInfo          : CloseError: (:) [Get-AzureVM], HttpRequestException  
    + FullyQualifiedErrorId : Microsoft.WindowsAzure.Commands.ServiceManagement.IaaS.GetAzureVMCommand

Get-OSVersions : The remote server returned an unexpected response: (407) Proxy Authenti cation Required. At line:1 char:15 + get-osversions <<<< -subscriptionId **\*** *-certificate (get-item cert:\CurrentUser\MY\**\*\*\*\*) + CategoryInfo : CloseError: (:) [Get-OSVersions], ProtocolException + FullyQualifiedErrorId : Microsoft.Samples.AzureManagementTools.PowerShell.HostedS ervices.GetOSVersionsCommand

**问题排查：**

从上面的报错信息来看，有些报错信息指向性比较模糊（get-azureVM），有一些则非常明确可以看到跟代理相关（get-osversions）。通常这种情况下，我们可以通过安装[fiddler](http://www.telerik.com/fiddler)工具来抓trace，进一步查看该问题。

抓取fiddler trace示例如下：

<HTML><HEAD>

<TITLE>Access Denied</TITLE>

</HEAD>

<BODY  bgcolor=green>

<FONT face="Helvetica">

<big><strong></strong></big><BR>

</FONT>

<blockquote>

<TABLE border=0 cellPadding=1 width="80%">

<TR><TD>

<FONT face="Helvetica">

<big>Access Denied (authentication\_failed)</big>

<BR>

<BR>

</FONT>

</TD></TR>

<TR><TD>

<FONT face="Helvetica">

Your credentials could not be authenticated: "Credentials are missing.". You will not be permitted access until your credentials can be verified.

</FONT>

</TD></TR>

<TR><TD>

<FONT face="Helvetica">

This is typically caused by an incorrect username and/or password, but could also be caused by network problems.

</FONT>

</TD></TR>

<TR><TD>

<FONT face="Helvetica" SIZE=2>

<BR>

<br>

 通过该trace可以发现是由于Proxy认证失败导致PowerShell无法发送http请求。这是由于PowerShell工具无法获取到Proxy的认证信息，从而连接失败。对于需要认证信息的代理，PowerShell是不支持的，需要我们做一些特殊的配置，让PowerShell能够成功获取到该认证信息。

**解决方法：**

通常的Http请求都是通过获取在浏览器IE中设置代理配置信息完成请求，但是如果您的代理环境是会提示您输入用户名密码的环境，这样简单的在IE中做配置是无法使得PowerShell成功发送http请求的。

但是，您可以自己实现一个简单的Proxy handler（包含代理的用户名和密码信息），并且将其导入到您的PowerShell应用程序对应路径下，这样就可以使得PowerShell成功发送http请求了。

具体方法如下：

1. 创建一个assembly，主要用来声明代理的认证信息。

例如： *SomeAssembly.dll*（可以通过Visual Studio创建project以及class，通过build该project获得该.dll文件）

Class示例：

namespace SomeNameSpace

{

    public class MyProxy : IWebProxy

    {

        public ICredentials Credentials

        {

            get { return new NetworkCredential("user", " password"); }

            //or get { return new NetworkCredential("user", " password"," domain"); }

            set { }

        }

        public Uri GetProxy(Uri destination)

        {

            return new Uri("<http://your.proxy:8080>");

       }

        public bool IsBypassed(Uri host)

        {

            return false;

        }

    }

}

注意：需要将对应的Proxy认证信息用真实环境信息代替。

1. 将该*.dll*文件放在PowerShell对应路径下：

32位PowerShell路径：C:\Windows\System32\WindowsPowerShell\v1.0.

64位PowerShell路径：C:\Windows\SysWOW64\WindowsPowerShell\v1.0

1. 创建一个powershell.exe.config文件，该文件用来声明Powershell不使用默认代理设置，而是使用刚才创建好的.dll中声明的代理。

powershell.exe.config示例：

<defaultProxy enabled="true" useDefaultCredentials="false">

  <module type = "SomeNameSpace.MyProxy, SomeAssembly" />

</defaultProxy>

注意：该config文件中高亮部分需要跟生成的.dll文件名称一致才能正确识别代理信息。

1. 将刚才创建好的powershell.exe.config放在PowerShell对应路径下，方法同2，这相当于向PowerShell导入一个代理信息。