

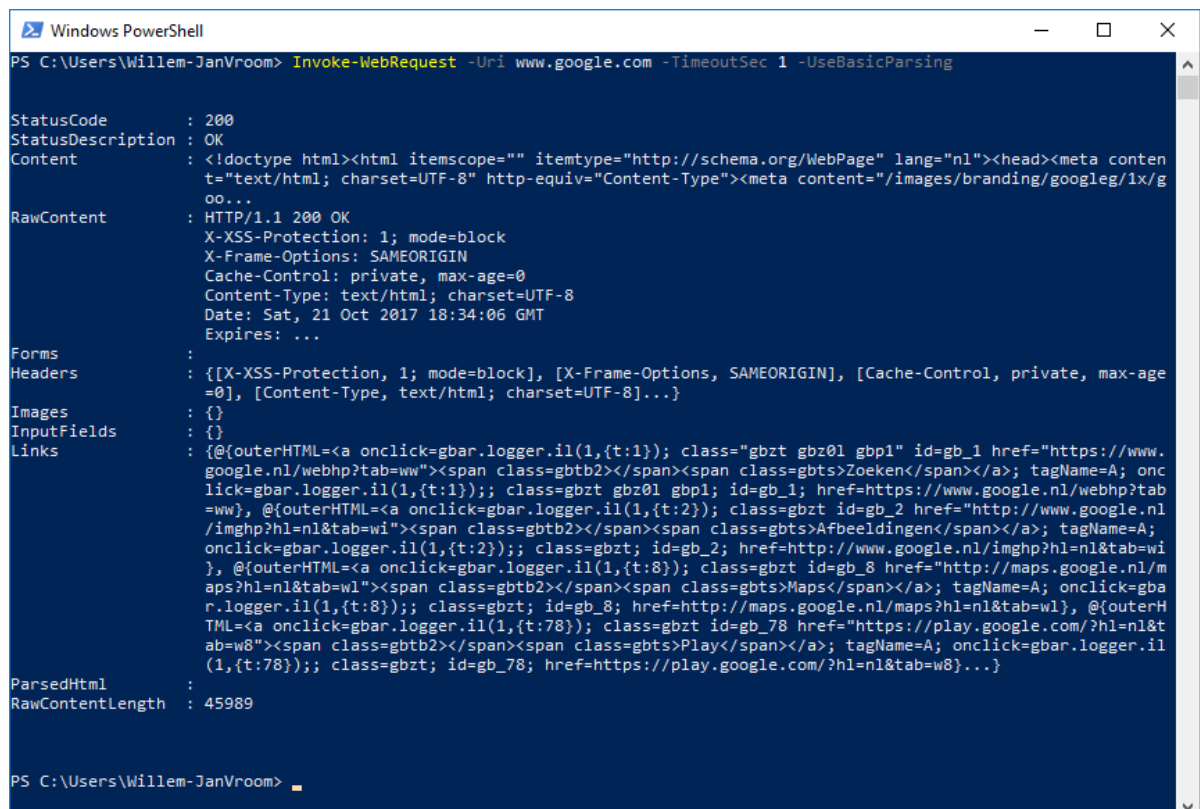
Check if there is a working internet connection when a firewall blocks ping requests

Sometimes you need to check if there is a working internet connection. You can use the `pingcommand` for that. But this command does not work if [ICMP echo request have disabled on the firewall](#). If this is the case we have to check the status of a website. If the status is 200 then there is a working internet connection.

In this example I will use both `vbscript` and `powershell` to check if there is a working internet connection.

The basics

1. The command `Invoke-WebRequest -Uri www.google.com -TimeoutSec 1 -UseBasicParsing` gives a lot of internet about the webpage. Pay attention to the statuscode.



```
Windows PowerShell
PS C:\Users\Willem-JanVroom> Invoke-WebRequest -Uri www.google.com -TimeoutSec 1 -UseBasicParsing

StatusCode      : 200
StatusDescription : OK
Content         : <!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="nl"><head><meta content="text/html; charset=UTF-8" http-equiv="Content-Type"><meta content="/images/branding/googleg/1x/goo...
RawContent      : HTTP/1.1 200 OK
                  X-XSS-Protection: 1; mode=block
                  X-Frame-Options: SAMEORIGIN
                  Cache-Control: private, max-age=0
                  Content-Type: text/html; charset=UTF-8
                  Date: Sat, 21 Oct 2017 18:34:06 GMT
                  Expires: ...
Forms           :
Headers         : {[X-XSS-Protection, 1; mode=block], [X-Frame-Options, SAMEORIGIN], [Cache-Control, private, max-age=0], [Content-Type, text/html; charset=UTF-8]...}
Images         : {}
InputFields    : {}
Links          : {@{outerHTML=<a onclick=gbar.logger.il(1,{t:1}); class="gbzt gbz01 gbpl" id=gb_1 href="https://www.google.nl/webhp?tab=ww"><span class=gbtb2></span><span class=gbts>Zoeken</span></a>; tagName=A; onclick=gbar.logger.il(1,{t:1}); class=gbzt gbz01 gbpl; id=gb_1; href=https://www.google.nl/webhp?tab=ww}, @{outerHTML=<a onclick=gbar.logger.il(1,{t:2}); class=gbzt id=gb_2 href="http://www.google.nl/imghp?hl=nl&tab=wi"><span class=gbtb2></span><span class=gbts>Afbeeldingen</span></a>; tagName=A; onclick=gbar.logger.il(1,{t:2}); class=gbzt; id=gb_2; href=http://www.google.nl/imghp?hl=nl&tab=wi}, @{outerHTML=<a onclick=gbar.logger.il(1,{t:8}); class=gbzt id=gb_8 href="http://maps.google.nl/maps?hl=nl&tab=w1"><span class=gbtb2></span><span class=gbts>Maps</span></a>; tagName=A; onclick=gbar.logger.il(1,{t:8}); class=gbzt; id=gb_8; href=http://maps.google.nl/maps?hl=nl&tab=w1}, @{outerHTML=<a onclick=gbar.logger.il(1,{t:78}); class=gbzt id=gb_78 href="https://play.google.com/?hl=nl&tab=w8"><span class=gbtb2></span><span class=gbts>Play</span></a>; tagName=A; onclick=gbar.logger.il(1,{t:78}); class=gbzt; id=gb_78; href=https://play.google.com/?hl=nl&tab=w8}...}

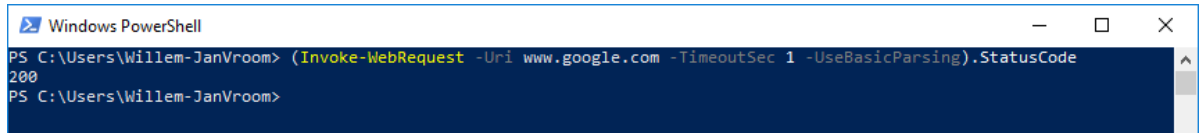
ParsedHtml     :
RawContentLength : 45989

PS C:\Users\Willem-JanVroom>
```

`Invoke-WebRequest -Uri www.google.com -TimeoutSec 1 -`

UseBasicParsing

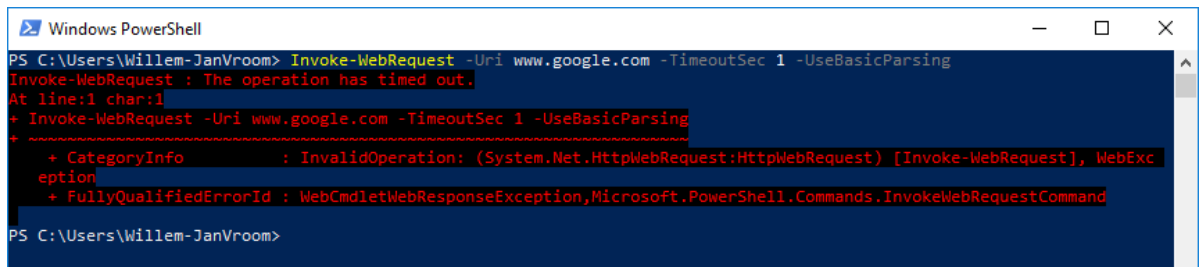
- The command `(Invoke-WebRequest -Uri www.google.com -TimeoutSec 1 -UseBasicParsing).StatusCode` returns only the statuscode.



```
Windows PowerShell
PS C:\Users\Willem-JanVroom> (Invoke-WebRequest -Uri www.google.com -TimeoutSec 1 -UseBasicParsing).StatusCode
200
PS C:\Users\Willem-JanVroom>
```

`(Invoke-WebRequest -Uri www.google.com -TimeoutSec 1 -UseBasicParsing).StatusCode`

- There is an error message if there is no working internet connection.

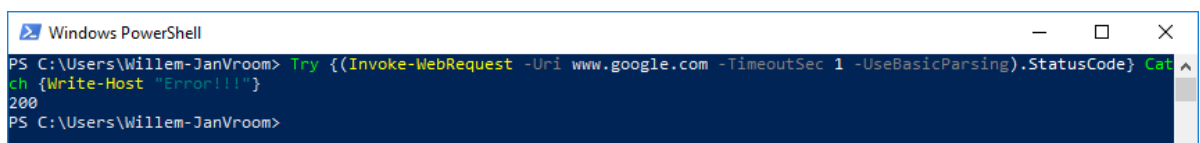


```
Windows PowerShell
PS C:\Users\Willem-JanVroom> Invoke-WebRequest -Uri www.google.com -TimeoutSec 1 -UseBasicParsing
Invoke-WebRequest : The operation has timed out.
At line:1 char:1
+ Invoke-WebRequest -Uri www.google.com -TimeoutSec 1 -UseBasicParsing
+ ~~~~~
+ CategoryInfo          : InvalidOperation: (System.Net.HttpWebRequest:HttpWebRequest) [Invoke-WebRequest], WebExc
option
+ FullyQualifiedErrorId : WebCmdletWebResponseException,Microsoft.PowerShell.Commands.InvokeWebRequestCommand
PS C:\Users\Willem-JanVroom>
```

`Invoke-WebRequest -Uri www.google.com -TimeoutSec 1 -UseBasicParsing`

With no working internet connection.

- If there is a working internet connection then the script returns 200.

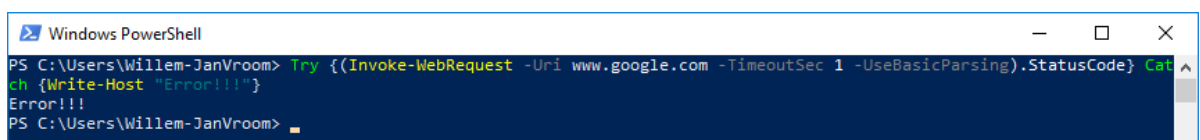


```
Windows PowerShell
PS C:\Users\Willem-JanVroom> Try {(Invoke-WebRequest -Uri www.google.com -TimeoutSec 1 -UseBasicParsing).StatusCode} Catch {Write-Host "Error!!!"}
200
PS C:\Users\Willem-JanVroom>
```

`Try {(Invoke-WebRequest -Uri www.google.com -TimeoutSec 1 -UseBasicParsing).StatusCode} Catch {Write-Host "Error!!!"}`

Working internet connection.

- In this case there is no working internet connection, so an error message is shown.



```
Windows PowerShell
PS C:\Users\Willem-JanVroom> Try {(Invoke-WebRequest -Uri www.google.com -TimeoutSec 1 -UseBasicParsing).StatusCode} Catch {Write-Host "Error!!!"}
Error!!!
PS C:\Users\Willem-JanVroom>
```

`Try {(Invoke-WebRequest -Uri www.google.com -TimeoutSec`

```
1 -UseBasicParsing).StatusCode} Catch {Write-Host
"Error!!!"}
No working internet connection.
```

Further testing with PowerShell

```
<#
```

```
.NOTES
```

```
=====
```

```
=====
```

```
Created with:      Windows PowerShell ISE
Created on:        21-October-2017
Created by:       Willem-Jan Vroom
Organization:
Filename:         CheckIfWebsiteIsReachable.ps1
```

```
=====
```

```
=====
```

```
.DESCRIPTION:
```

It checks if there is a working internet connection by reading the webpage.

This is very usefull in case the ping command does not work when ICMP echo requests are disabled on the firewall.

```
#>
```

```
param ($URL)
```

```
Function Get-StatusCodeFromWebsite
```

```
{
    param($Website)
    Try
    {
        (Invoke-WebRequest -Uri $Website -TimeoutSec 1 -
UseBasicParsing).StatusCode
    }
    Catch
    {
    }
}
```

```
$URL = "www.google.com"
$Result = Get-StatusCodeFromWebsite -Website $URL
if($Result -eq 200)
{
    Write-Host "The website $URL is reachable."
}
else
{
    Write-Host "The website $URL is not reachable."
}
```

```
PS C:\PowerShell> C:\PowerShell\CheckIfWebsiteIsReachable.ps1
The website www.google.com is reachable.

PS C:\PowerShell>
```

Result if there is a working internet connection.

```
PS C:\PowerShell> C:\PowerShell\CheckIfWebsiteIsReachable.ps1
The website www.google.com is not reachable.

PS C:\PowerShell> |
```

Result if there is no working internet connection.

The next step: use vbscript and powershell to detect if there is a working internet connection

As far as I know vbscript has not the option to detect the status from a website. So we use the powershell function with vbscript to detect if there is a working internet connection.

The powershell script has changed a little bit:

```
<#
.NOTES
```

```
=====
=====
```

```
Created with:    Windows PowerShell ISE
Created on:      21-October-2017
Created by:      Willem-Jan Vroom
```

Organization:

Filename: CheckIfWebsiteIsReachable_with_vbscript.ps1

=====
=====

.DESCRIPTION:

It checks if there is a working internet connection by reading the webpage.

This is very usefull in case the ping command does not work when ICMP echo requests are disabled on the firewall.

#>

Param (\$URL)

Function Get-StatusCodeFromWebsite

```
{
    param($Website)
    Try
    {
        (Invoke-WebRequest -Uri $Website -TimeoutSec 1 -
UseBasicParsing).StatusCode
    }
    Catch
    {
    }
}
```

\$Result = Get-StatusCodeFromWebsite -Website \$URL

if(\$Result -eq 200)

```
{
    Exit 0
}
else
{
    Exit 1
}
```

It exits with an exit code:

0 = success

1 = failure

And we use vbscript to check:

```

'
=====
=====
' Check if a there is a working network connection. Do not use
ping, as the ping command cannot
' bu used in case ICMP echo requests have been disabled on the
firewall.
' Created by Willem-Jan Vroom
' Version history:
'
' 0.0.1
' - Initial version
'
' 1.0.0
' - Final version
'
=====
=====

' -----
-----
' Declare the most variables.
' -----
-----

Option Explicit

    Dim objShell          : set objShell          =
WScript.CreateObject("WScript.Shell")
    Dim strWebsite       : strWebsite           = "http://www.vroom.cc"
        Dim strCurrentDir : strCurrentDir       =
Left(Wscript.ScriptFullname, InstrRev(Wscript.ScriptFullname,
"\"))
    Dim strCommand       : strCommand           =
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -
ExecutionPolicy Bypass -WindowStyle Hidden -File " & chr(34) &
strCurrentDir & "CheckIfWebsiteIsReachable_with_vbscript.ps1"
& chr(34) & " -URL "& strWebsite
    Dim strText          : strText              = ""
    Dim valResult        : valResult            = 0

```

```

' -----
-----
' Check the website that is mentioned in the string
strWebsite.
' Use the exitcode from the powershell script:
' 0 = Success
' 1 = Failure
' -----
-----

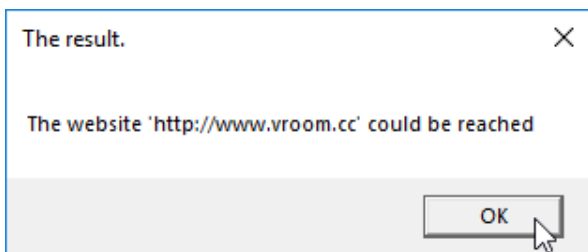
valResult = objShell.Run(strCommand, 6, True)
if valResult = 0 Then
    strText = "The website '" & strWebSite & "' could be
reached"
    else
        strText = "The website '" & strWebSite & "' could not be
reached"
    end if

' -----
-----
' Display a message box with the result.
' -----
-----

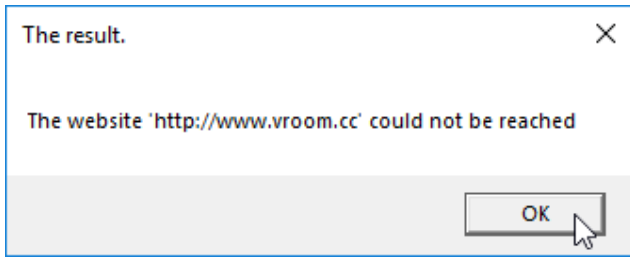
msgbox strText,0,"The result."

```

In this example we try the webpage <http://www.vroom.cc/>.



Result if there is a working internet connection.



Result if there is no working internet connection.

You can [download the scripts in a ZIP file](#).