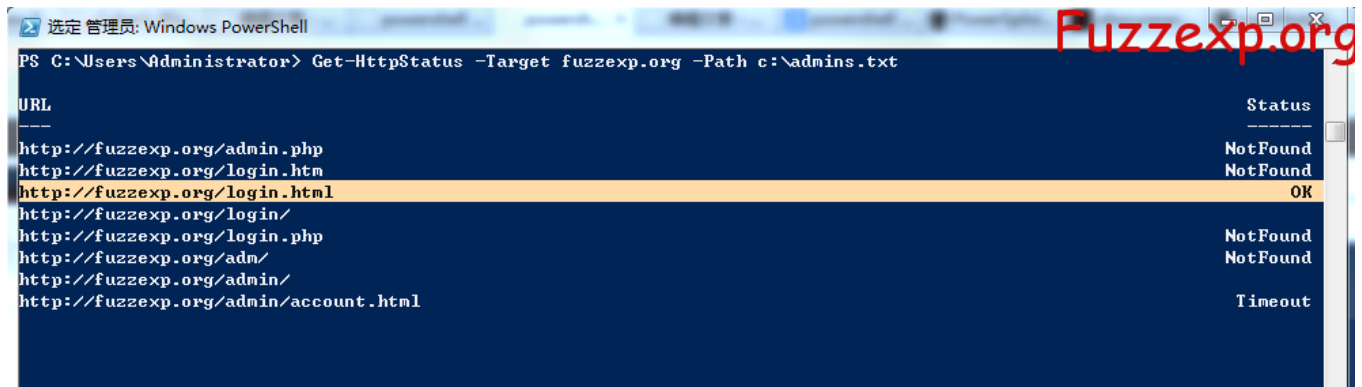


powershell :Get-HttpStatus Function

2012-08-18 08:58:31 By admin

DT小妹妹说今年黑冒出了很多powershell的东西，一搜索，还真多，看到了 Get-HttpStatus Function这东西不错 还真好玩



URL	Status
http://fuzzexp.org/admin.php	NotFound
http://fuzzexp.org/login.htm	NotFound
http://fuzzexp.org/login.html	OK
http://fuzzexp.org/login/	NotFound
http://fuzzexp.org/login.php	NotFound
http://fuzzexp.org/adm/	NotFound
http://fuzzexp.org/admin/	NotFound
http://fuzzexp.org/admin/account.html	Timeout

当然他不会那么无聊，前几天的[Set: Windows7 Bypassing using Powershell](#) 证明他的HACK方面真心强大，感谢DT小妹妹，愿你早日破处 msdn大叔的[Bash vs PowerShell](#) 值得围观一下，附带 [Presentation: PowerShell for Pen Testers](#) Metasploit的POST EXPLOIT模块 `post/windows/manage/powershell/exec_powershell`能很方便的免杀. 对于WIN7，取得会话用SET的powershell BYPASS功能，运行命令用exec_powershell是一个小技巧哦

代码：

```
function Get-HttpStatus {  
<#  
.SYNOPSIS  
PowerSploit Module - Get-HttpStatus
```

Returns the HTTP Status Codes and full URL for specified paths.

Author: Chris Campbell (@obscuresec)

License: BSD 3-Clause

.DESCRIPTION

A script to check for the existence of a path or file on a webserver.

.PARAMETER Target

Specifies the remote web host either by IP or hostname.

.PARAMETER Path

Specifies the remost host.

.PARAMETER Port

Specifies the port to connect to.

.PARAMETER UseSSL

Use an SSL connection.

.EXAMPLE

```
PS > Get-HttpStatus -Target www.example.com -Path c:\dictionary.txt | Select-Object {where StatusCode -eq 20*}
```

.EXAMPLE

```
PS > Get-HttpStatus -Target www.example.com -Path c:\dictionary.txt -UseSSL
```

.NOTES

HTTP Codes: 100 - Informational * 200 - Success * 300 - Redirection * 400 - Client Error * 500 - Server Error

Status Codes: <http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>

.LINK

<http://obscuresecurity.blogspot.com>

#>

```
[CmdletBinding()] Param(
    [Parameter(Mandatory = $True)] [String] $Target,
    [Parameter()] [String] [ValidateNotNullOrEmpty()] $Path = '.\Dictionaries\admin.txt',
    [Parameter()] [Int] $Port,
    [Parameter()] [Switch] $UseSSL
)
```

```
if (Test-Path $Path) {
```

```
    if ($UseSSL -and $Port -eq 0) {
        # Default to 443 if SSL is specified but no port is specified
        $Port = 443
    } elseif ($Port -eq 0) {
        # Default to port 80 if no port is specified
        $Port = 80
    }
}
```

```
$TcpConnection = New-Object System.Net.Sockets.TcpClient
Write-Verbose "Path Test Succeeded - Testing Connectivity"
```

```
try {
    # Validate that the host is listening before scanning
    $TcpConnection.Connect($Target, $Port)
} catch {
    Write-Error "Connection Test Failed - Check Target"
    $Tcpconnection.Close()
    Return
}
```

```
    $Tcpconnection.Close()
} else {
    Write-Error "Path Test Failed - Check Dictionary Path"
    Return
}
```

```
if ($UseSSL) {
    $SSL = 's'
    # Ignore invalid SSL certificates
    [System.Net.ServicePointManager]::ServerCertificateValidationCallback = { $True }
} else {
    $SSL = ''
}

if (($Port -eq 80) -or ($Port -eq 443)) {
    $PortNum = ''
} else {
    $PortNum = ":$Port"
}

# Check Http status for each entry in the doctionary file
foreach ($Item in Get-Content $Path) {

    $WebTarget = "http$($SSL)://$($Target)$($PortNum)/$($Item)"
    $URI = New-Object Uri($WebTarget)

    try {
        $WebRequest = [System.Net.WebRequest]::Create($URI)
        $WebResponse = $WebRequest.GetResponse()
        $WebStatus = $WebResponse.StatusCode
        $ResultObject += $ScanObject
        $WebResponse.Close()
    } catch {
        $WebStatus = $Error[0].Exception.InnerException.Response.StatusCode

        if ($WebStatus -eq $null) {
            # Not every exception returns a StatusCode.
            # If that is the case, return the Status.
            $WebStatus = $Error[0].Exception.InnerException.Status
        }
    }

    $Result = @{ Status = $WebStatus;
        URL = $WebTarget}

    $ScanObject = New-Object -TypeName PSObject -Property $Result

    Write-Output $ScanObject
}
}
```

更多的牛B功能围观这里 <https://github.com/mattifestation/PowerSploit/>

版权声明：

本站遵循 [署名-非商业性使用-相同方式共享 2.5](#) 共享协议.

转载请注明转自[Dis9 Team](#)并标明URL.

本文链接 <http://fuzzexp.org/?p=4954>