

Crear un dominio en la deep web de forma fácil y sencilla

Descargar TOR

- <https://www.torproject.org/>

Crear un servidor web en el puerto 80 desde PowerShell (también puedes utilizar un servidor web como Apache o IIS)

```
#####
#####
#Servidor web con PowerShell
#Ejecutar PowerShell como administrador
#####
#####
$routes = @{
    "/" = { return ' asdfasfasServidor web funcionando' }
}

#IP localhost
$url = 'http://127.0.0.1:80/'
$listener = New-Object System.Net.HttpListener
$listener.Prefixes.Add($url)
$listener.Start()

Write-Host "Funcionando $url..."

while ($listener.IsListening)
{
    $context = $listener.GetContext()
    $requestUrl = $context.Request.Url
    $con
    $response = $context.Response

    Write-Host ''
    Write-Host "Petición: $requestUrl"

    $localPath = $requestUrl.LocalPath
    $route = $routes.Get_Item($requestUrl.LocalPath)
```

```

if ($route -eq $null)
{
    $response.StatusCode = 404
}
else
{
    $content = & $route
    $buffer = [System.Text.Encoding]::UTF8.GetBytes($content)
    $response.ContentLength64 = $buffer.Length
    $response.OutputStream.Write($buffer, 0, $buffer.Length)
}
$response.Close()

$responseStatus = $response.StatusCode
Write-Host "Respuesta: $ResponseStatus"
}

```

The screenshot shows the Windows PowerShell ISE interface with two scripts loaded. The left script, titled 'Sin título1.ps1', is a web server listener on port 127.0.0.1:80. It uses the System.Net.HttpListener class to listen for requests. If a route is found, it returns the content of the route; otherwise, it returns a 404 status code. The right script, titled 'Sin título1.ps1 X', is a web server listener on port 127.0.0.1:80. It uses the System.Net.HttpListener class to listen for requests. It has a route table with one route: '/ui' pointing to 'c.onion/'. If a route is found, it returns the content of the route; otherwise, it returns a 404 status code. The status bar at the bottom indicates 'Lin.45 Col.2' and '110 %'.

```

Sin título1.ps1
1 $url = 'http://127.0.0.1:80/'
2 $listener = New-Object System.Net.HttpListener
3 $listener.Prefixes.Add($url)
4 $listener.Start()
5
6 write-Host "Funcionando $url..."
7
8 while ($listener.IsListening)
9 {
10     $context = $listener.GetContext()
11     $requestUrl = $context.Request.Url
12     $con = $context.Response
13
14     write-Host ''
15     write-Host "Petición: $requestUrl"
16
17     $localPath = $requestUrl.LocalPath
18     $route = $routes.Get_Item($requestUrl.LocalPath)
19
20     if ($route -eq $null)
21     {
22         $response.StatusCode = 404
23     }
24     else
25     {
26         $content = & $route
27         $buffer = [System.Text.Encoding]::UTF8.GetBytes($content)
28         $response.ContentLength64 = $buffer.Length
29         $response.OutputStream.Write($buffer, 0, $buffer.Length)
30     }
31
32     $response.Close()
33
34     $ResponseStatus = $response.StatusCode
35     write-Host "Respuesta: $ResponseStatus"
36 }
37
38 Funcionando http://127.0.0.1:80/...
39
40 Petición: http://ui
41 Respuesta: 200
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

Sin título1.ps1 X
1 #####
2 #Servidor web con PowerShell
3 #####
4 $routes = @{}
5 $routes['/'] = { return '<html><meta charset="UTF-8"> <meta http-equiv="refresh" content="0; u
6 }
7
8 #IP localhost
9 $url = 'http://127.0.0.1:80/'
10 $listener = New-Object System.Net.HttpListener
11 $listener.Prefixes.Add($url)
12 $listener.Start()
13
14 write-Host "Funcionando $url..."
15
16 while ($listener.IsListening)
17 {
18     $context = $listener.GetContext()
19     $requestUrl = $context.Request.Url
20     $con = $context.Response
21     $response = $context.Response
22
23     write-Host ''
24     write-Host "Petición: $requestUrl"
25
26     $localPath = $requestUrl.LocalPath
27     $route = $routes.Get_Item($requestUrl.LocalPath)
28
29     if ($route -eq $null)
30     {
31         $response.StatusCode = 404
32     }
33     else
34     {
35         $content = & $route
36         $buffer = [System.Text.Encoding]::UTF8.GetBytes($content)
37         $response.ContentLength64 = $buffer.Length
38         $response.OutputStream.Write($buffer, 0, $buffer.Length)
39     }
40 }

```

Modificar el fichero torrc que se encuentra en la ruta
 \TOR\Tor Browser\Browser\TorBrowser\Data\Tor y añadir



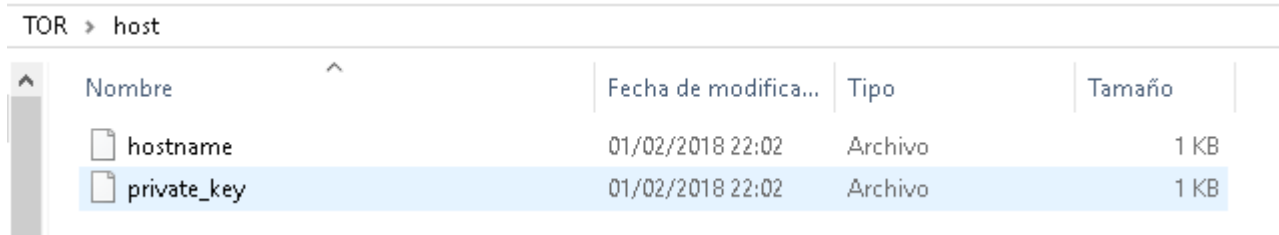
Añadir al fichero las siguientes líneas (el directorio en
 donde se crearan los ficheros necesarios para crear el dominio

y la dirección IP y el puerto donde se ejecuta nuestro servidor web):

HiddenServiceDir C:\Users\juan\Desktop\TOR\host

HiddenServicePort 80 127.0.0.1:80

Ejecutar TOR y verificar que aparecen dos ficheros en la carpeta HiddenServiceDir



The screenshot shows a Windows File Explorer window with the address bar set to 'TOR > host'. The main area displays a table of files. The table has four columns: 'Nombre', 'Fecha de modifica...', 'Tipo', and 'Tamaño'. There are two files listed: 'hostname' and 'private_key', both of which are 1 KB in size and were last modified on 01/02/2018 at 22:02. The 'private_key' file is currently selected, highlighted in blue.

Nombre	Fecha de modifica...	Tipo	Tamaño
hostname	01/02/2018 22:02	Archivo	1 KB
private_key	01/02/2018 22:02	Archivo	1 KB

Abrir el fichero hostname y dentro del fichero se encuentra el dominio creado .onion