

Forum:[Hack Forums](#)

Thread: [Tor Browser](#)

Shoot - January 13, 2015, 6:44 pm

how did they make their own ".onion" web extensions when they created the tor browser?

does anyone know? im posting in this section because if i knew i believe it could be done as a vb.net project

Sakura-chan - January 13, 2015, 6:48 pm

Custom .DLL's I believe

If not, I have not clue what i'm talking about

-Jax

Nikey646 - January 13, 2015, 6:51 pm

All network traffic when using a Tor Network goes through Tor-based servers. They then use a local DNS in those servers to convert .onion into IP Addresses like any normal DNS service, otherwise they will either lookup other domains, or re-direct you to your usual DNS.

This is my guess, and the most logical conclusion.

Shoot - January 13, 2015, 6:58 pm

(Jan 13, 2015 07:51 PM)Nikey646 Wrote: [\[CITE\]](#) All network traffic when using a Tor Network goes through Tor-based servers. They then use a local DNS in those servers to convert .onion into IP Addresses like any normal DNS service, otherwise they will either lookup other domains, or re-direct you to your usual DNS.

This is my guess, and the most logical conclusion.

seems to clear it up a bit because you cant just make your own custom web extensions (i google'd it) the only ones you can have are premade.

i thought it was more like they did:

have a VPS and on this VPS (in the public_html directory) they have files called "example.onion" then using the web browser to compile the html using this code:

```
Code Dim request As System.Net.HttpWebRequest =  
System.Net.HttpWebRequest.Create("http://192.168.1.1/example.onion")  
Dim response As System.Net.HttpWebResponse = request.GetResponse()
```

```
Dim sr As System.IO.StreamReader = New  
System.IO.StreamReader(response.GetResponseStream())
```

```
Dim sourcecode As String = sr.ReadToEnd()
```

```
txtSrc.Text = sourcecode
```

```
WebBrowser1.DocumentText = txtSrc.Text
```

that way it grabs the source of the custom file and the document text compiles the source but i could be wrong.

i did try it with a free host and it did work like that.

Nikey646 - January 13, 2015, 7:56 pm

(Jan 13, 2015 07:58 PM)Shoot Wrote: [\[CITE\]](#) seems to clear it up a bit because you cant just make your own custom web extensions (i google'd it) the only ones you can have are premade.

i thought it was more like they did:

have a VPS and on this VPS (in the public_html directory) they have files called "example.onion" then using the web browser to compile the html using this code:

```
Code Dim request As System.Net.HttpWebRequest =  
System.Net.HttpWebRequest.Create("http://192.168.1.1/example.onion")  
Dim response As System.Net.HttpWebResponse = request.GetResponse()
```

```
Dim sr As System.IO.StreamReader = New  
System.IO.StreamReader(response.GetResponseStream())
```

```
Dim sourcecode As String = sr.ReadToEnd()
```

```
txtSrc.Text = sourcecode
```

```
WebBrowser1.DocumentText = txtSrc.Text
```

that way it grabs the source of the custom file and the document text compiles the source but i could be wrong.

i did try it with a free host and it did work like that.

Exceeepppttt, that .Onion is the TLD, not a file extension. You can configure the way File Extensions are handled in Apache via header control, and skip the entire internal processing.

KingDeath360 - January 14, 2015, 3:06 am

I remember doing a project on this last year.

I was using fiddler core to intercept url's that match example.test, lol.test...etc...

Ill try and look out the source code for you

EDIT: <https://pastebin.com/f53enqtA>

????????? - January 15, 2015, 10:17 pm

the .onion must be registered at a DNS server of somesort, which looks up the URL when all other URL resolving fails (computer, router). then it forwards you to an onion load balancer which takes you to the right IP chain. (my assumption)

iamskid - January 16, 2015, 10:57 am

Stop assuming things and look in the tor source code.

•Evolved• - January 16, 2015, 11:24 am

It wouldn't be too hard to do, it just wouldn't be as advanced or secure as TOR, implying that itself is anyways. You would just need to make your own DNS server, your own fake TLD, and a browser that would send the request for that TLD to the custom DNS server that runs the TLD lookup. I might make an example project of this. HMU in PM if you are interested in watching me livestream the process.

•Evolved• - January 17, 2015, 5:04 pm

Posted a thread with a small project demonstrating what I'm talking about.

<https://hackforums.net/showthread.php?tid=4641627>

