page was renamed from Preventing Tor DNS Leaks

FIXME: Also, after someone merges the non-redundant info into Socks and DNS, it probably won't have a reason to exist anymore.

Many Tor users don't realize that when they use Tor, there is a risk that although connections across Tor network are torified and encrypted, DNS requests made within the torified application will not be routed via the Tor network. Consequently the DNS requests will be made via your own machine. This is a security risk because anyone watching your computer or your DNS server can tell what names you are looking up, and guess that those are the domains you are visiting via Tor.

How do I know if my applications are leaking DNS?

In your Tor logs or Vidalia panel there will be warning messages that say this: "Your application (using socks4/5 on port %d) is giving Tor only an IP address. Applications that do DNS resolves themselves may leak information. Consider using Socks4A (e.g. via privoxy or socat) instead. For more information, please see Socks and DNS. "

In order to be able to view warning messages in realtime, open your Vidalia client, click 'Select Filter' and make sure that (Warn) Messages are ticked.

On Unix-like systems, where `/var/log/tor/log` is Tor's log file, use `tail` to view Tor messages in realtime:

```
tail -f /var/log/tor/log
```

Sometimes these messages may be false alarms. To find out, you should run a packet sniffer on your network interface. In Wireshark (ex Ethereal), use the filter expression "`dns`". For tcpdump, where `eth0` is the name of your network interface, use this command:

```
tcpdump -pni eth0 'port domain'
```

Tor gives you these warnings when your application passes it a raw IP address instead of a hostname. IP addresses look like "18.244.0.188." Hostnames look like "moria.seul.org." As general rule of thumb is, if Tor gives you those warnings, then either:

- you have a DNS leaking problem.
- you gave your application a raw IP address to begin with, and your application passed that address on to Tor.

What are some methods I can use to rectify this problem?

If your application does DNS prefetching, turn it off. In Firefox, type `about:config` in the location bar and change **network.dns.disablePrefetch** to **true**. For Konqueror or Akregator, add the line

```
DNSPrefetch=false
```

under the `[HTML Settings]` header in `$HOME/.kde/share/config/kdeglobals` (if the header `[HTML Settings]` doesn't already exist, you can add it to the end of the file).

Torsocks is probably the best way to ensure that all of an application's network traffic passes through Tor. Torsocks can prevent DNS leakage due to things like DNS prefetching or binary plugins that generate their own network traffic, whereas HTTP or SOCKS proxies cannot prevent these kinds of leaks. (On the other hand, proxies like Privoxy can be configured to send only certain traffic through Tor, whereas Torsocks forces all traffic through Tor.)

If your application supports HTTP proxies, you might consider using Privoxy. Privoxy listens on localhost port 8118, and the versions shipped with Tor come preconfigured to forward traffic via Tor using SOCKS 4a. This is good, because using SOCKS 4a causes DNS requests to be made remotely, and therefore does not leak DNS. So if you set it up properly, Privoxy *itself* does not leak DNS at all. Note however, as mentioned above, that poorly behaved applications may still leak DNS even if configured to send traffic through Privoxy. If you wish to use a web-browser-like application with Privoxy, you may also want to run it with Torsocks if you want to ensure that any stray traffic gets sent through Tor.

If your application supports SOCKS4a, then configure it to use a SOCKS4a proxy at localhost, port 9050. However, it is important to note that this does not always work. Sometimes applications are written with poor SOCKS4a functionality, even if they list it as an option. This results in DNS leaks.

If your application supports SOCKS5, it may use either hostnames or IP addresses.

Another option that works is running a DNS server locally that is designed to support Tor. `tor-dns-proxy.py` from [http://www.monkey.org/~dugsong/dsocks/|dsocks] is such an implementation for GNU/Linux (doc/TransocksifyingTor), BSD ([TheOnionRouter/PreventingDnsLeaksInTor]), Mac OS X, and probably other operating systems as well. `tor-dns-proxy.py` may work on Windows, but it hasn't been tested there yet. Somebody reading this page ought to test it and add instructions to the wiki if they have success.

On Windows you can use TorDNS. Be aware, though, that "TorDNS" hasn't gotten any security overview from the Tor maintainers, since it's written in Powerbasic of all things.

Alternatively, if you are looking for another solution that has both many positives and many negatives, you can download the Anonym.OS LiveCD. This is a modified OpenBSD distribution, and because it is a LiveCD, you run it straight from the CD-ROM drive. It sets up its own local DNS server that routes requests via Tor. So you can be sufficiently sure that you're being protected using this LiveCD.

Windows users can run JanusVM to transparently proxy their HTTP, TCP and DNS traffic using a PPTP VPN connection. If the default router is also resolving DNS queries on your system you made need to alter name server addresses to avoid leaking DNS information due to a windows routing table deficiency. JanusVM documentation explains this issue in more detail.

By !DrChemicalX April 23rd 2006

You may want to setup a local DNS server which rejects onion addresses, in case you disable accidentally torbutton or simply want to avoid DNS leaks to hidden services to reach your ISP DNS server.

The changes to bind9 are:

Add to /etc/bind/named.conf.local the following text:

```
zone "onion" {
        type master;
        file "onion";
};
```

Create a new file /etc/bind/onion with the following content:

```
$TTL 3D
@       IN      SOA     ns.linux.bogus. hostmaster.linux.bogus. (
                        199802151       ; serie, fecha de hoy + serie de ho
                        8H              ; refresco, segundos
                        2H              ; reintento, segundos
                        4W              ; expira, segundos
                        1D )            ; mínimo, segundos
;
                NS      ns              ; Dirección Inet del servidor de no
;

```

(semicolon means start of comment) Restart bind9.

By Ruben Garcia October 11, 2008

Google Chrome

If you are using Google Chrome disable the DNS pre-fetching.