Learning more about the GFW's active probing system

Posted September 14th, 2015 by phw

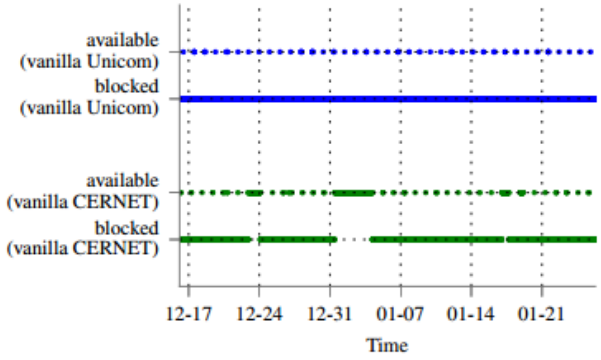
This blog post is also available in Chinese (https://zh.greatfire.org/blog/2015/9%E6%9C%88/gfw%E4%B8%BB%E5%8A%A8%E6%8E%A2%E6%B5%8B%E7%B3%BB%E7%BB%9F%E7%A0%94%E7%A9%B6%E6%8A%A5%E5%91%8A), translated by our friends from GreatFire.org (https://zh.greatfire.org).

Roya (https://cs.princeton.edu/~rensafi/), David, Nick (https://cs.princeton.edu/~feamster/), nweaver (http://www1.icsi.berkeley.edu/~nweaver/), Vern (http://www.icir.org/vern/), and I just finished a research project in which we revisited the Great Firewall of China's (GFW) active probing system. This system was brought to life several years ago to reactively probe and block circumvention proxies, including Tor. You might remember an earlier blog post (https://blog.torproject.org/blog/knock-knock-knockin-bridges-doors) that gave us some first insight into how the active probing system works. Several questions, however, remained. For example, we were left wondering what the system's physical infrastructure looked like. Is the GFW using dedicated machines behind their thousands of probing IP addresses? Does the GFW even "own" all these IP addresses? Rumour had it that the GFW was hijacking IP addresses for a short period of time, but there was no conclusive proof. As a result, we teamed up and set out to answer these, and other questions.

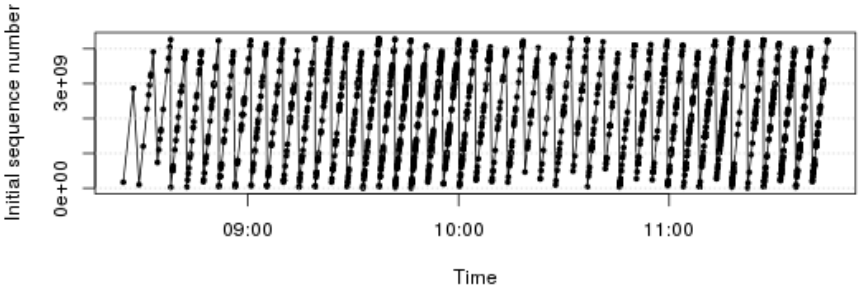
Because this was a network measurement project, we started by compiling datasets. We created three datasets, comprising *hours* (a Sybil-like experiment to attract many probes), *months* (an experiment to measure reachability for clients in China), and even *years* (log files of a long-established server) worth of active probing data. Together, these datasets allow us to look at the GFW's active probing system from different angles, illuminating aspects we wouldn't be able to observe with just a single dataset. We are able to share two of our datasets (https://nymity.ch/active-probing/#code), so you are very welcome to reproduce our work, or do your own analysis.

We now want to give you an overview of our most interesting findings.

- Generally, once a bridge is detected and blocked by the GFW, it remains blocked. But does this mean that the bridge is *entirely* unreachable? We measured the blocking effectiveness by continuously making a set of virtual private systems in China connect to a set of bridges under our control. We found that every 25 hours, for a short period of time, our Tor clients in China *were able to connect* to our bridges. This is illustrated in the diagram shown below. Every point represents one connection attempt, meaning that our client in China was trying to connect to our bridge outside of China. Note the curious periodic availability pattern for both Unicom and CERNET (the two ISPs in China we measured from). Sometimes, network security equipment goes into "fail open" mode while it updates its rule set, but it is not clear if this is happening here.



- We were able to find patterns in the TCP headers of active probes that suggest that all these thousands of IP addresses are, in fact, controlled by *a single source*. Check out the initial sequence number (ISN) pattern in the diagram below. It shows the value of ISNs (y-axis) over time (x-axis). Every point in the graph represents the SYN segment of one active probing connection. If all probing connections would have come from independent computers, we would have expected a random distribution of points. That's because ISNs are typically chosen randomly to protect against off-path attackers. Instead, we see a clear linear pattern across IP addresses. We believe that active probes derive their ISN from the current time.



- We discovered that Tor is not the only victim of active probing attacks; the GFW is targeting other circumvention systems, namely SoftEther (https://www.softether.org/) and GoAgent. This highlights the *modular nature* of the active probing system. It appears to be easy for GFW engineers to add new probing modules to react to emerging, proxy-based circumvention tools.
- The GFW is able to (partially) speak the vanilla Tor protocol, obfs2, and obfs3 to probe bridges. Interestingly, node-Tor (https://github.com/Ayms/node-Tor) —a JavaScript implementation of the Tor protocol—is immune to active probing because it implements the Tor protocol differently, which seems to confuse active probes. We were also able to resist active probes by modifying a bridge of ours to ignore old `VERSIONS` Tor cells. This is unlikely to be a sustainable circumvention technique, though.
- Back in 2012 (https://blog.torproject.org/blog/knock-knock-knockin-bridges-doors), the system worked in 15-minute-queues. These days, it seems to be able to scan bridges in *real-time*. On average, it takes only half a second after a bridge connection for an active probe to show up.
- Using a number of traceroute experiments, we could show that the GFW's sensor is stateful and seems unable to reassemble TCP streams.

Luckily, we now have several pluggable transports (https://www.torproject.org/docs/pluggable-transports.html.en) that can defend against active probing. ScrambleSuit (http://www.cs.kau.se/philwint/scramblesuit/) and its successor, obfs4 (https://github.com/Yawning/obfs4/blob/master/doc/obfs4-spec.txt), defend against probing attacks by relying on a shared secret that is distributed out of band. Meek (https://trac.torproject.org/projects/tor/wiki/doc/meek) tunnels traffic over cloud infrastructure, which does not prevent active probing, but greatly increases collateral damage when blocked. While we keep developing and maintaining

circumvention tools, we need to focus more on usability. A powerful and carefully-engineered circumvention tool is of little use if folks find it too hard to use. That's why projects like the UX sprint (https://blog.torproject.org/blog/ux-sprint-2015-wrapup) are so important.

Finally, you can find our research paper (https://nymity.ch/active-probing/imc2015.pdf) as well as our datasets and code (https://nymity.ch/active-probing/#code) on our project page (https://nymity.ch/active-probing/). And don't hesitate to get in touch with us (https://nymity.ch/active-probing/#contact) if you have any questions or feedback!

Comment viewing options

Threaded list - expanded | Date - oldest first | 300 comments per page

Save settings

Select your preferred way to display the comments and click "Save settings" to activate your changes.

On September 15th, 2015 Anonymous said:

What security problem to TBB I should care about if I get TBB running in the command line:

sh -c '"/home/uuss/tor-browser_en-US/Browser/start-tor-browser" --detach || ([! -x "/home/uuss/tor-browser_en-US/Browser/start-tor-browser"] && "$(dirname "$*")"/Browser/start-tor-browser --detach)' dummy %k

On September 17th, 2015 dcf said:

This is a good question for the Q&A site: https://tor.stackexchange.com/ (https://tor.stackexchange.com/).

On October 8th, 2015 Anonymous said:

So wait, you're wanting to run it as root? Don't do that. If the browser gets comprised, which is really not that difficult, it will be able to fuck with your system badly. A non exhaustive list of how root can take over the kernel:
- write to MSRs that allow arbitrary memory access
- load kernel modules
- exploit privileged subsystems like perf
- boot into a malicious kernel with kexec
- replace the kernel image in /boot
- issue iopems and iopls that allow arbitrary write access
- send code to the GPU, which has direct access to memory
- disable many access controls, and bypass permissions
- and much more

Do you REALLY want JavaScript or anything else you come across on the web to run in a context which can do that? Unless you are a masochist, I should think not.

Judging by your understanding of command line though, I should think you would already know that running things like web browsers as root is a bad idea...

On October 8th, 2015 Anonymous said:

Shit. Disregard my last comment... I thought you had run su -c, not sh -c...

On September 16th, 2015 Anonymous said:

Great post and research - this is what Tor needs the most! Pluggable transports are vital for those dissidents who need it the most and continually improving upon them is important!

On September 16th, 2015 Anonymous said:

This is great work, thanks for all of your efforts to understand and counteract this behavior! If I understand correctly, the active probes can detect obfs2 and obfs3, but not obfs4, correct? If the GFW detects Tor on an IP, does it block connections to all ports at that IP? If so, does that imply that bridge administrators running obfs4 and wishing to avoid being blocked should disable obfs2 and obfs3?

> On September 16th, 2015 phw said:
>
> *This is great work, thanks for all of your efforts to understand and counteract this behavior! If I understand correctly, the active probes can detect obfs2 and obfs3, but not obfs4, correct?*
>
> Yes, the GFW can detect obfs2 and obfs3. I don't think that the GFW is able to detect obfs4 yet.
>
> *If the GFW detects Tor on an IP, does it block connections to all ports at that IP? If so, does that imply that bridge administrators running obfs4 and wishing to avoid being blocked should disable obfs2 and obfs3?*
>
> Currently, the GFW only blocks the IP address and port, the service was running on. Still, running obfs4 on the same IP address as the easy-to-detect obfs2 might help a censor figure out that your obfs4 is, in fact, obfs4, even though it should currently work against the GFW.

On September 16th, 2015 Anonymous said:

Currently, the GFW only blocks the IP address and port, the service was running on. Still, running obfs4 on the same IP address as the easy-to-detect obfs2 might help a censor figure out that your obfs4 is, in fact, obfs4, even though it should currently work against the GFW.

Good to know, thanks!

On September 18th, 2015 Anonymous said:

Even if a bridge has only obfs4, doesn't the regular ORPort give it away?

> On September 18th, 2015 phw said:
>
> Unfortunately yes. See https://trac.torproject.org/projects/tor/ticket/7349 (https://trac.torproject.org/projects/tor/ticket/7349)

On September 16th, 2015 Anonymous said:

I depend on Tor to access the news (not from China), so this kind of research is much appreciated.

On September 16th, 2015 Anonymous said:

http://www.engadget.com/2014/11/18/tor-browser-vulnerability/

Tor is vulnerable like no other...

On September 17th, 2015 dcf said:

Be careful with the press's interpretation of that research. You should read the comment left by the author of the research:

> *https://blog.torproject.org/blog/traffic-correlation-using-netflows#comm... (https://blog.torproject.org/blog/traffic-correlation-using-netflows#comment-78918)*
> *I am here to myself clarify all misconceptions. Firslty, they have blow it a bit out of proportion by saying that "81% of Tor traffic", which is not true. It was only 81.4% of our experiments, and we have spoken about this upfront in our paper. Secondly, its only a case of experimental validation and the challenges involved in it that is the highlight of the paper. In my thesis I have also tried to address how to solve this particular attack, which might work for other attacks as well...*

You can find links to other analysis in the article attached to that comment.

On September 21st, 2015 Anonymous said:

Has any of the author's suggestions been discussed. Can these attacks be mitigated? What has the team come up with so far?

On September 17th, 2015 Anonymous said:

Just a question, when obfs4 is used what kind of traffic is seen by someone who is inspecting/probing traffic i.e. ISP or Global surveillance? Does it look like a clearnet webpage connection or something else?

On September 17th, 2015 dcf said:

obfs4 is one of the "look-like-nothing" transports. It doesn't try to look like normal web browsing or anything like that; instead it looks like random bytes.

We have a wiki page (https://trac.torproject.org/projects/tor/wiki/doc/AChildsGardenOfPluggableTransports) that visualizes some of the pluggable transports. obfs4 is not described there yet, but you can get the idea by looking at the other look-like-nothing transports: obfs2 (https://trac.torproject.org/projects/tor/wiki/doc/AChildsGardenOfPluggableTransports#obfs2), obfs3 (https://trac.torproject.org/projects/tor/wiki/doc/AChildsGardenOfPluggableTransports#obfs3), and ScrambleSuit (https://trac.torproject.org/projects/tor/wiki/doc/AChildsGardenOfPluggableTransports#ScrambleSuit). Compare those to FTE (https://trac.torproject.org/projects/tor/wiki/doc/AChildsGardenOfPluggableTransports#FTE), which *does* transform traffic to fool a firewall into thinking it is some other protocol.

On September 18th, 2015 Anonymous said:

If it is "random bytes" could an adversary not figure out that this not traffic that is seen by a regular user without Tor? Or would this be equivalent to seeing traffic of someone who has downloaded a binary file?

> On September 18th, 2015 phw said:
>
> *If it is "random bytes" could an adversary not figure out that this not traffic that is seen by a regular user without Tor?*
>
> In a way, yes. The obfs family and ScrambleSuit hide in the set of (not

entirely known) protocols whose payload bytes are uniformly distributed. As an adversary, you could go ahead and drop every TCP connection that looks like that. And maybe that would work fine. We are not sure, because we don't know exactly what the networks of our adversaries are like. We expect, or hope, however, that blocking such TCP connections would come with high collateral damage.

On September 17th, 2015 Anonymous said:
Most IP of Tor keeps getting banned! Cannot access anything anymore!

On September 17th, 2015 Anonymous said:
Does scramble suit change packet timings in a way that would stop global adversary timing attacks? You know for the rest of us that have to worry about 5 eyes.

On September 18th, 2015 phw said:
ScrambleSuit does alter its packet lengths and inter-arrival times, but only on the link between the Tor user and her bridge. End-to-end characteristics are not modified. Also, ScrambleSuit's shape shifting isn't much more than an experiment at this point. Low-latency anonymity networks that can protect against global adversaries are still an unsolved problem.

On September 18th, 2015 Anonymous said:
while global adversarywould be a tough nut, china, iran or syria are not nearly global adveraries. most of the relays, the developer team, the website, the servers that some tor services rely on are outside of these countries.
This is not like an unsolved math problem its just about coding man power. there are enough ideas how to improove pluggable transports but atm china seems to have more or better organized people. its a needle and haystack situation. we are the needle but we still we loose hide and seek.

On September 21st, 2015 yawning said:
None of China, Iran, or Syria look at packet timing as far as I am aware. Of all the adversaries China and Iran present the most challenge (China because the GFW people are decent, Iran because they will happily incur lots of collateral damage at times).

As far as I am aware (and according to the paper) there are circumvention methods that work in all three of the countries. That said, if someone has an idea for something that is:

a) Deployable (no closed-source components, passes design/code review)
b) Gets past a protocol whitelist.
c) Provides adequate performance on a 128 kbit link.
d) Is resilient to connections over 60s being throttled.

It should get further investigation.

FYI: Neither ScrambleSuit nor obfs4 will alter packet timing significantly in the default configuration because it absolutely kills performance, and no known censors look at it.

On September 21st, 2015 Anonymous said:
replay recorded http connections.

send the bridge a get request with the url encoded that has been recorded.
the bridge accesses that url and records it as well.
bridge and torclient play a webhost and browser game.
the payload gets hidden inside the content like transmitted images or html

On September 22nd, 2015 yawning said:

This sounds like Stegotorus. If someone were to clean up and audit the code, it may actually be deploy-able some day, though certain things are unsolved (primarily how to mange to build a large enough unique corpus on each bridge, and targeted attacks against the Stegonography used).

The performance of this sort of thing likely isn't going to be fantastic, but the link characteristics that call for this aren't great to begin with so that might be ok.

On September 26th, 2015 Anonymous said:

Hopefully Tor can assign someone to maintain and help develop this. When I first saw it, I was surprised development had stopped. Only reason I heard from developers was because not too many people were requesting it but how can people know what they want when 99% of users don't pay attention to the types of pluggable transports until they are actually implemented.

On September 18th, 2015 Anonymous said:

If a bridge was to change from a bridge to regular node, would Tor Browser know that it has changed and notify user (error message) or will users be still using it under the false impression that it is still a bridge?

On September 21st, 2015 Anonymous said:

@tordevs - can we get someone official to answer this?

On September 26th, 2015 Anonymous said:

It's seems tor developers themselves don't know the correct answer...

On September 19th, 2015 Anonymous said:

why not to bulid meek-cloud9 plugin and bulid a new browser which based on chrouim.

On September 20th, 2015 Anonymous said:

What about the LOSFWs of America, Europe, Australia, Afrika, ... blocking tor successful?

On September 20th, 2015 Anonymous said:

Excellent work guys. Many thanks to the translation team.

All efforts in determining these probes will assist in opening up avenues for perhaps

eventually, if not soon, to not have a need for secrecy and probing. This is where we as a collective (such as fantastic ones here at Tor) Roger, Mike, and many others can assist humanity in getting out of their cocoon.

Keep up the great work.

imu.

On September 21st, 2015 Anonymous said:
Will Tor Browser randomize your MAC address or machine ID to allow access to networks or servers than banned your MAC or machine ID, instead of the IP?

> On September 22nd, 2015 Anonymous said:
> How would a server know about your MAC anyway?

> On September 22nd, 2015 Anonymous said:
> Anyone can change their Mac address on their own. Tails does that automatically, though.

On September 21st, 2015 Anonymous said:
What do you think of Tallow?

https://reqrypt.org/tallow.html

On September 21st, 2015 Anonymous said:
The best actually is that I am reading this article from China mainland without any kind of proxy, a straight tcp connection to blog.torproject.org .

On September 22nd, 2015 Anonymous said:
Why does Tails in VM need host OS time to be correct?

On September 22nd, 2015 Anonymous said:
Current TBB Firefox version is 38.0, but the latest version is 40.0.3. What to do?

On September 22nd, 2015 Anonymous said:
RUSSIA'S PLAN TO CRACK TOR CRUMBLES - Bloomberg Business

http://www.bloomberg.com/news/articles/2015-09-22/russia-s-plan-to-crack-tor-crumbles

> On September 30th, 2015 Anonymous said:
> If they did crack Tor, do you think they would make that information public?

> > On October 6th, 2015 Anonymous said:
> > Nah. They would just sit back and read the mail. Literally.

On September 23rd, 2015 Anonymous said:

thanks a lot for your effort

On September 30th, 2015 Anonymous said:
Can there be an option for Tor relays to ignore old VERSION connections?

On October 1st, 2015 Anonymous said:
> Anyone can change their Mac address on their own. Tails does that automatically, though.

A new challenge for the Tails team: Tails users are potentially being set up for swatting by local police trying to locate stolen laptops and other devices:

http://www.techdirt.com/articles/20150912/
Cop Invents Device That Sniffs MAC Addresses To Locate Stolen Devices
Tim Cushing
16 Sep 2015

https://thehackernews.com/2015/09/track-stolen-devices.html
Techie Police Officer Builds a Sniffing Tool to Track Stolen Devices (based on War-Driving)
Swati Khandelwal
9 Sep 2015

From their perspective, this is analogous to using ALPR to search for stolen cars. From our perspective, we are trying to defend ourselves against unwarranted nastiness like this:

https://thehackernews.com/2014/01/spying-agencies-tracking-your-location_31.html
Spying agencies tracking your location by capturing MAC address of your devices
31 Jan 2014
Swati Khandelwal

The arms race between the 99.9% and the slavish servants of the 0.1% continues...

On October 3rd, 2015 Anonymous said:
Read mention of obfs5 awhile back in Tor Blog. Would anyone give us an idea of what features are being considered for obfs5?

> On October 7th, 2015 yawning said:
> No. When there's an actual spec and reference implementation, I'll publish it, but not till then.

On October 7th, 2015 Anonymous said:
Is scramblesuit the same thing as obfs4?

On October 10th, 2015 Anonymous said:
Obfs3 provides obfuscation. ScrambleSuit provides packet padding. Obfs4 provides both obfuscation and packet padding. Tor Metrics Portal shows 9000 users of obfs3 but only 2000 users of obfs4 when obfs4 seems to me to be the better Pluggable Transport. Is it because obfs3 continues to be the 'reccommended' Pluggable Transport when setting up Tor Browser?

On October 19th, 2015 Anonymous said:

The Russians using DPI. Who would have guessed?

http://www.telegraph.co.uk/news/worldnews/europe/russia/11934411/Russia-tried-to-cut-off-World-Wide-Web.html