BLOCKING TOR - INSIGHT APROACH

Vladimir Stanojević[1]

[1]Medical and Business-Technology College of Applied Studies,
Hajduk Veljkova St., Šabac

Abstract:
Tor is a very powerful tool for avoiding restrictions of the Internet usage in local networks (companies or educational/public institutions). For example, students use it on everyday basis at university computer labs. In order to prevent such misuse from happening, most effective solutions widely known go as far as obtaining current list of Tor nodes and updating the local network security policy with thousands of IP addresses. In this paper, one more efficient and effective method is presented by analyzing Tor package source code as well as aiming at core infrastructure of Tor network. This method is much more reliable and consumes fewer resources while completing the task with success. It is used instead of not so reliable frequent Tor nodes sniffing, which is mainstream approach.

Keywords:
The Onion Routing, Tor, blocking Tor, Internet Censorship

1. INTRODUCTION

Tor, originally known as The Onion Routing [1], which was initially designed for covert communications of USA Naval Intelligence Agency, was released to public usage with original idea in mind to provide anonymity cover for individuals who cared about own Internet privacy.

Of course, misuse of Tor is widespread. One can use it to make anonymous hacker attacks to various targets online, or other criminal activity [2]. During years, Internet and its usage became everyday activity in offices all around the world. With power comes responsibility, but not everyone obeys this concept [3]. It is customary that employees browse through various fun content on the Internet, social networks, online video games, etc. So equally often, Internet access in companies and educational institutions is restricted more or less for that reason.

As a counter - counter measure, employees or other users of official network resources use Tor to circumvent actual Internet restriction policy.

However, not only administrative employees and students use Tor. World - class terrorists, criminals, Internet lurkers (pedophiles etc.), corrupt government officials, industrial spies, the list is endless. The significance is so high that even NSA engages dedicated resources in monitoring Tor activity [4]. This is publicly announced by Edward Snowden and is published in Guardian [5].

Correspondence:
Vladimir Stanojević

e-mail:
vlstanojevic@gmail.com

DOI: 10.15308/Sinteza-2017-25-29

This requires new, more advanced techniques for blocking such culprits. Reasons for this are various, and so are the techniques and resources used. For example, Government agencies have vast resources for that [6].

The Great Firewall of China is one of the most notorious Internet censors which blocks Tor with high success rate [7].

As one of the most powerful and largest countries in the world, China uses enormous computer resources to block Tor, with high success rate. However, for individual companies and educational institutions such resources are unthinkable. Therefore, approach to block Tor in such real-life scenarios is different. Tor itself is projected not to be able to defend against this scale attacker [8].

Most widely spread method for blocking Tor, due to limited resources, which are available to some company or public institution, is by IP filtering [9]. This method is tested in real-life, and proved to be borderline effective enough as experimental results show.

In this paper, one different approach is presented. It is based on analyzing the Tor source code and disabling the usage of Tor by denying access not to every Tor node, but to the Tor consensus servers, thus denying the Tor client even the possibility to know any available Tor node in order to connect to in the first place.

2. STANDARD METHOD FOR BLOKCING TOR METHOD DESCRIPTION

Standard, default method of blocking Tor (excluding in-depth packet analysis [10], which requires enormous resources available only on government level agencies as previously mentioned) is to obtain ever-current list of Tor nodes and to set up corresponding rules at central gateway in order to deny access to that pool of IP addresses, thus blocking Tor clients that run from inside local network. This method can be illustrated with algorithm shown on Figure 1.

The list of currently active Tor nodes can be acquired in various manners. One is through official Tor API or web service/application called Atlas [10]. The other one is to obtain “ready to use list” from third party sources [11]. As long as this list is suitable for human user [11], it consists of many other data, which obfuscates in certain degree usage of such list as “ban-list” of all IP addresses. For such purpose, a “plain” list of IP addresses of Tor nodes [12] is much more suitable.

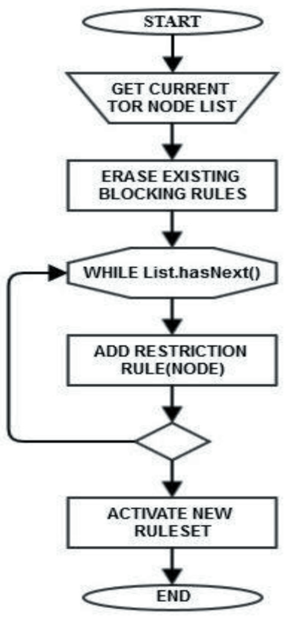

START
GET CURRENT TOR NODE LIST
ERASE EXISTING BLOCKING RULES
WHILE List.hasNext()
ADD RESTRICTION RULE(NODE)
ACTIVATE NEW RULESET
END

Fig. 1. Algorithm of standard Tor blockage method

3. REFERENCE METHOD EFFECTIVENESS, EXPERIMENT AND RESULT

This method was put to the test of effectiveness in Medical and Business-Technology College of Applied Studies, in three computer labs. Labs consisted of total 68 student computers. Several hundreds of students (over 200 students of IT dept.) used them during the experiment. The experimental period was 60 days.

Students had been using Tor actively before the experiment took place in order to circumvent current restrictions of the Internet usage. Mostly, they had been using it for social networking or online video games during classes. After implementing the blockage of Tor by means of previously described method, the following results are obtained and given in Table 1.

Description	Unsuccessful Tor connections/ total number of tries made ratio	
	Objective measure	Subjective impression
List refreshed every 2 hours	89%	100%
List refreshed every 1 hour	89.2%	100%
Number of IP addresses in blacklist	6083	

Table 1. Experimental results

Experiment showed rather high percentage of successful Tor blocking. In reality, subjective impression by students who tried to use Tor was that it was blocked 100%, because it took on average more than 7 minutes for Tor clients to connect successfully to Tor node which was not "blacklisted" in given time. By that time, students lost patience and closed it, supposing that a connection would never be established.

However, this method with success rate less than 90% is not satisfactory, and needs to be improved to achieve 100% success rate if possible.

4. REFFERENCE METHOD ISSUES AND MEANS OF IMPROVEMENT

There are two issues to this method that must be noted. Firstly, it can cause delays in Internet traffic routing if used on SOHO grade network router due to long list of iterative conditions against which every network package towards the Internet must be checked. Secondly, it "filters" network traffic only against currently "known" IP addresses that are Tor nodes. First limitation can be upgraded by using more powerful resource, but second issue is much more difficult if not impossible to overcome. The whole concept behind presented method is that all current tor nodes are known and blocked. However, it cannot be done in reality due to the fact that structure and size of the Tor network vary every second. Many Tor clients come on- and off-line every second. This results in a list, which is accurate, current and acquired in a moment. Next moment, many of listed nodes can go offline and as many, more can become online, but not known by previously acquired list.

So, besides making irrationally frequent updates of "blacklist", which will raise accuracy of current available Tor nodes list, at the same time creating even more and more delay and router resources exhausting, it will not completely resolve the issue.

It makes obvious that this method cannot be improved to be very efficient no matter the costs. Solution to this issue requires completely different approach.

5. INSIGHT APPROACH CONCEPT

Concept behind this new improved method is based on deeper understanding of Tor infrastructure, and using its own design against it. When reading about Tor specifications [13], [14], [15], one can focus attention on "directory authorities". Each Tor client "reports" itself to server acting as directory authority. Afterwards it refreshes its status every 18 hours.

In official Tor documentation [12], this report is called server descriptor. Every hour these directory servers referred as "authorities" vote on particular Tor node. After achieving "consensus" on that node, they enlist a specific node as regular Tor relay, so other Tor traffic can go through or even out (if node is allowed to be an exit, Tor relay by its immediate user). Such enlisted Tor relays are those available to fetch and to be used as list of currently active Tor nodes (relays), and blacklist can be populated from it as in previously described method.

However, being "live" and very fluid, the list is never accurate as we concluded in experiment above.

Taking a step further, instead of blocking access to any known Tor node in a moment of time, more efficient would be to deny Tor client neither to obtain the list in the first time nor be able to update it afterwards.

This can be achieved by obtaining the list of above mentioned directory authorities' servers IP addresses and filling the blacklist with only this final and one-figure number of IP addresses. Of course, this can be changed, but not as near as often as Tor relays list.

The list of directory authorities together with their IP addresses is "hardcoded" in Tor clients. Therefore, this list can only be changed with new version release, which is not even on daily basis, and is not obliged to be changed with new version release, quite the contrary.

6. IMPLEMENTATION

Implementation of this method is straightforward. First step is to obtain source code package of Tor client current version. Since it is available from static hyperlink [17], it is easy to incorporate it in a program code that will accomplish the task. On that hyperlink [17] numerous versions of Tor packages are located. Very simple program code is needed to parse through the available links to find download link for current version of source code package. In case when no newer version of Tor has been published, no further action is needed.

The second step is to compare previously known version with the available one, and if unchanged, no further action is needed. When doing this, checking on daily basis is more than enough to achieve reasonably high success rate. In comparison to the previous method, the check could be performed in the same way and the whole firewall rewrite would be done every second. If there were a new version of Tor available, third step would be just to download the package, unpack it, and then parse through config.c file, in order to obtain list of current directory authorities and their IP addresses.

Exact part of that file containing data of interest is displayed on listing 1. IP addresses of directory authorities are marked in bold-italic with the purpose of emphasizing them.

```
"128.31.0.39:9131 9695 DFC3 5FFE B861
329B 9F1A B04C 4639 7020 CE31",
"86.59.21.38:80 847B 1F85 0344 D787 6491
A548 92F9 0493 4E4E B85D",
"194.109.206.212:80 7EA6 EAD6 FD83 083C
538F 4403 8BBF A077 587D D755",
"Tonga orport=443 bridge no-v2
82.94.251.203:80 "
    "4A0C CD2D DC79 9508 3D73 F5D6
6710 0C8A 5831 F16D",
    "turtles orport=9090 no-v2 "
"76.73.17.194:9030 F397 038A DC51 3361
35E7 B80B D99C A384 4360 292B",
    "gabelmoo orport=443 no-v2 "
"212.112.245.170:80 F204 4413 DAC2 E02E
3D6B CF47 35A1 9BCA 1DE9 7281",
    "dannenberg orport=443 no-v2 "
"193.23.244.244:80 7BE6 83E6 5D48 1413
21C5 ED92 F075 C553 64AC 7123",
    "urras orport=80 no-v2
"208.83.223.34:443 0AD3 FA88 4D18 F89E
EA2D 89C0 1937 9E0E 7FD9 4417",
    "maatuska orport=80 no-v2
"171.25.193.9:443 BD6A 8292 55CB 08E6
6FBE 7D37 4836 3586 E46B 3810",
    "Faravahar orport=443 no-v2 "
"154.35.32.5:80 CF6D 0AAF B385 BE71 B8E1
11FC 5CFF 4B47 9237 33BC"
```

Listing 1. Part of config.c (Tor source package file) containing directory authorities' data

```
BufferedReader br = null;
int cnt=0;
try {
br = new BufferedReader(new
FileReader("config.c"));
Pattern pattern =
Pattern.compile(IPADDRESS_PATTERN);
String line = br.readLine();
while (line != null) {
 if (!semafor)if
(line.contains("authorities[] = {"))
semafor=true;
 else{
Matcher matcher=Pattern.matcher(line);
if (matcher.find()){
 imenik[cnt]=matcher.group();
 cnt++;
}
if (line.contains("};")) return imenik;
 line = br.readLine();
}
```

Listing 2. – Fragment of code parsing config.c for IP addresses of directory authorities

7. INSIDE APPROACH METHOD EXPERIMENTAL RESULTS

During the same test period (60 days) in the same facility (68 computers available to multiple students, being the same as in referenced method), by analyzing the network activity log files, results are given in Table 2. This experiment was made in real-life environment, due to available resources both technical and human. However, it can be replicated in simulated environment too [17].

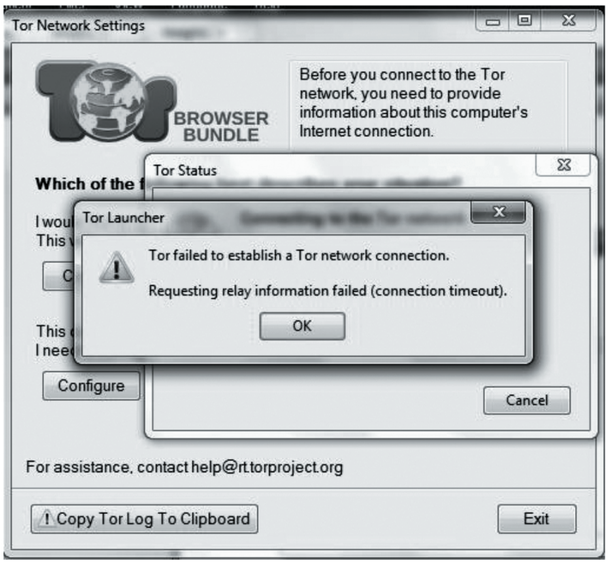
Description	Unsuccessful Tor connections/ total number of tries made ratio	
	Objective measure	Subjective impression
Tor source checked for new release every 24 hours	100%	100%
Number of IP addresses in blacklist	10	

Table 2. Results of experiment of new method implementation

As shown, using this approach to block Tor, enormous savings in CPU time and thus in network responses time is evident. On the contrary, this is not achieved by compromising with security. By reducing CPU consumption for more than 6800 times and avoiding consumption of bandwidth for acquiring fresh list of nodes every hour, absolute effectiveness is achieved.

8. CONCLUSION

Using an improved method presented in this paper, with frequency of potential updates on daily basis, by blacklisting only eight directory authorities servers, Tor client could not be used in a controlled environment. The same was used in previous experiment, in the same amount of time and computers/users. Picture 1. displays one screenshot of Tor client showing error message.


Tor Network Settings

Before you connect to the Tor network, you need to provide information about this computer's Internet connection.

Tor Status

Tor Launcher

Tor failed to establish a Tor network connection.

Requesting relay information failed (connection timeout).

For assistance, contact help@rt.torproject.org

Picture 1. Tor unable to connect

Most indicative is the error message in Log itself, saying:

[NOTICE] I learned some more directory information, but not enough to build a circuit: We need more microdescriptors: we have 0/4728, and can only build 0% of likely paths. (We have 0% of guards bw, 0% of midpoint bw, and 0% of exit bw.)

So experiment shows that by implementing such a method Tor is completely blocked on long-term basis, long enough to "keep gates shut tight". At the same time, resources and CPU time used to accomplish this result are symbolic.

REFERENCES

[1] M. G. Reed; P. F. Syverson; D. M. Goldschlag (1998); Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications*, 16(4):(482–494).

[2] Diana S. Dolliver (2015): "Evaluating drug trafficking on the Tor Network: Silk Road 2" *International Journal of Drug Policy*, The Department of Criminal Justice, The University of Alabama

[3] McCoy D.; Bauer K.; Grunwald D.; Kohno T. ; Sicker D. (2008). Shining light in dark places: Understanding the Tor network. In *Privacy Enhancing Technologies* (pp. 63-76). Springer Berlin Heidelberg.

[4] Bruce Schneier (2013): How the NSA Attacks Tor/Firefox users with QUANTUM and FOXACID, Schneier on security blog, published October 7th 2013.

[5] Glen Grenwald (2013): XKeyscore: NSA tool collects nearly everything a user does on the internet – The Guardian Jul 31st 2013 -http://www.theguardian.com/world/2013/jul/31/nsa-top-secret-program-online-data

[6] Seth Rosenblatt (2014): NSA likely targets anybody who's 'Tor-curious', CNet Security Jul 3rd 2014-http://www.cnet.com/news/nsa-likely-targets-anybody-whos-tor-curious/

[7] Xu, X.; Mao, Z; M., & Halderman; J. A. (2011); Internet censorship in China: Where does the filtering occur?. In *Passive and Active Measurement*/Springer Berlin Heidelberg (133-142)

[8] R. Dingledine; N. Mathewson (2006):Design of a blocking-resistant anonymity system,Tor Project technical report

[9] Chapman; D. B.; (1992); Network (In) Security Through IP Packet Filtering. In *USENIX Summer* (2-3)

[10] https://atlas.torproject.org/#about

[11] https://www.dan.me.uk/tornodes

[12] https://www.dan.me.uk/torlist/

[13] Guerraoui R.; Schiper A. (1996) Consensus service: a modular approach for building agreement protocols in distributed systems. In *Fault Tolerant Computing*, , *Proceedings of Annual Symposium on* (pp. 168-177). IEEE.

[14] Acquisti A.; Dingledine R.; Syverson, P. (2003). On the economics of anonymity. In *Financial Cryptography* (pp. 84-102). Springer Berlin Heidelberg.

[15] https://gitweb.torproject.org/torspec.git/blob/HEAD:/dir-spec.txt

[16] https://gitweb.torproject.org/torspec.git/blob/HEAD:/dir-spec.txt#l1485

[17] https://www.torproject.org/download/download.html.en

[18] Jansen, R., & Hooper, N. (2011). *Shadow: Running Tor in a box for accurate and efficient experimentation* (No. TR-11-020). MINNESOTA UNIV MINNEAPOLIS DEPT OF COMPUTER SCIENCE AND ENGINEERING.