Deep Web Users Deanonimization System

S.M. Avdoshin <savdoshin@hse.ru>
A.V. Lazarenko <avlazarenko @edu.hse.ru>
School of Software Engineering,
National Research University Higher School of Economics,
20, Myasnitskaya st., Moscow, 101000 Russia

Abstract. Privacy enhancing technologies (PETs) are ubiquitous nowadays. They are beneficial for a wide range of users: for businesses, journalists, bloggers, etc. However, PETs are not always used for legal activity. There a lot of anonymous networks and technologies which grants anonymous access to digital resources. The most popular anonymous networks nowadays is Tor. Tor is a valuable tool for hackers, drug and gun dealers. The present paper is focused on Tor users' deanonimization using out-of-the box technologies and a basic machine learning algorithm. The aim of the work is to show that it is possible to deanonimize a small fraction of users without having a lot of resources and state-of-the-art machine learning techniques. The first stage of the research was the investigation of contemporary anonymous networks. The second stage was the investigation of deanonimization techniques: traffic analysis, timing attacks, attacks with autonomous systems. For our system, we used website fingerprinting attack, because it requires the smallest number of resources needed for successful implementation of the attack. Finally, there was an experiment held with 5 persons in one room with one corrupted entry Tor relay. We achieved a quite good accuracy (70%) for classifying the webpage, which the user visits, using the set of resources provided by global cybersecurity company. The deanonimization is a very important task from the point of view of national security.

Keywords: Tor; deanonimization; website fingerprinting; traffic analysis; anonymous network; deep web.

DOI: 10.15514/ISPRAS-2016-28(3)-2

For citation: Avdoshin S.M., Lazarenko A.V. Deep Web Users Deanonimization System. *Trudy ISP RAN/Proc. ISP RAS*, vol. 28, issue 3, 2016, pp. 21-34. DOI: 10.15514/ISPRAS-2016-28(3)-2

1. Introduction

Internet privacy is considered as an integral part of freedom of speech. A lot of people are concerned about their anonymity in public and therefore, there is a growing need for privacy enhancing technologies.

The Deep Web is a layer of the Internet, which can not be accessed by traditional search engines, so the content in this layer is not indexed. The typical website in the deep web is static, with potentially no links to outer resources. For that reason, it is very hard to measure the real size of the deep web.

In the modern world, there are a lot of networks and technologies, which grant access to deep web resources, for example, Tor, I2P, Freenet, etc. Each of these instruments hides users' traffic from adversaries, thus making the deanonimization a hard thing to do. A detailed overview of such technologies can be accessed in paper [1].

Nowadays, the largest and most widely used system is Tor [2]. Our research focuses on Tor users' deanonimization, because of its popularity and prevalence.

2. Tor background

Tor is the largest active anonymous network in the world. There are more than two million users per month, and the number of relays is close to 7000 [3]. Tor is a distributed overlay network consisting of volunteer servers. Every user in the world can provide Tor with computational resources needed for traffic retranslation over the network.

Despite being a great privacy enhancing technology for law-abiding citizens, Tor is an essential tool in criminal society. Terrorists, drug and arm dealers in line with other offenders use Tor for their criminal activities. Thus, the solution of the deanonimization problem is very important for government special services [4]. For example, Russian Ministry of Internal Affairs (MIA) has recently announced a bidding for Tor deanonimization system [5].

The next key component of Tor is Hidden Services (HS). Tor HS provides users with anonymous servers to host their websites or any other applications. HS are accessed via special pseudo-domains «.onion», where Deep Web is located. From the user`s point of view, accessing a particular hidden service is as easy as visiting a normal website.

In order to establish a connection with Tor network, the user must have pre-installed software (Tor client). The easiest way is to install TorBrowser, which is a customized version of Mozilla Firefox with built-in Tor software. To initiate the connection, a Tor client obtains a list of Tor nodes from a directory server. Then, the client builds a circuit of encrypted connections through relays in the network. The circuit is extended hop by hop, and each relay on the path knows only which relay gives data and which relay it is giving data to. There is no particular relay in the circuit (see Fig. 1), which knows the complete users path through the network.

A Layered encryption is used along the path. The most interesting relays for a potential attacker are entry and exit relays. Every piece of information in the network is transferred in Tor cells that have equal size. An Entry relay (also called the guard) knows the IP address of the user, and Exit relay knows the destination

resource. Traffic interception in the middle would not give any advantage to the attacker because everything is encrypted and secure.

3. Deanonimization techniques

There is a wide range of deanonimization methods (attacks). Some of them are passive: an adversary only observes traffic, without any trials to modify it somehow. Contrariwise, some of them are active: an attacker modifies traffic causing delays, insert patterns, etc. Earlier, we proposed classification of attacks, where the main principle is the amount of resources needed by an attacker to perform the deanonimization (see table 1).

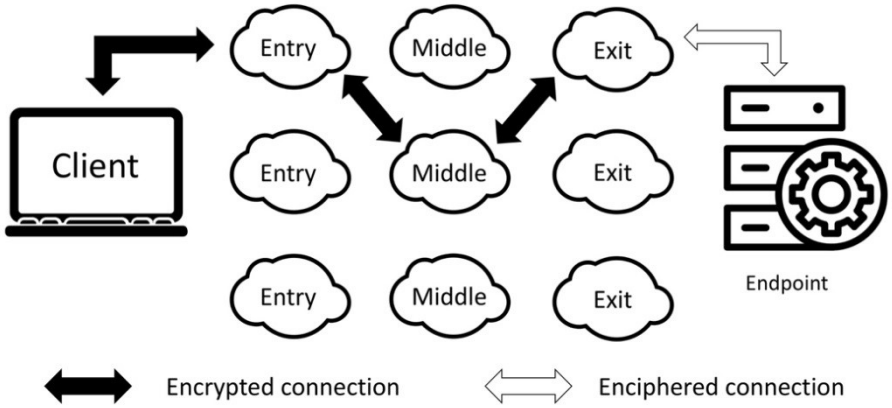

Fig. 1. Tor circuit example

Table 1. Attacks classification

#	Resources	Attacks
1	Corrupted entry guard	• Website fingerprinting attack
2	Corrupted entry and exit nodes	• Traffic analysis • Timing attack • Circuit fingerprinting attack • Tagging attack
3	Corrupted exit node	• Sniffing of intercepted traffic
4	Corrupted entry and exit nodes, external server	• Browser based timing attack with JavaScript injection • Browser based traffic analysis attack with JavaScript injection
5	Autonomous system	• BGP hijacking • BGP interception • RAPTOR attack
6	Big number of various corrupted nodes	• Packet spinning attack • CellFlood DoS attack • Other DoS and DDoS attacks

More information about attacks mentioned in Table 1 can be found in paper [6]. We are focused on the resource-effective attack (WF), which only requires an attacker to control an entry relay of the user. The relay, which is fully controlled by an attacker is called a *corrupted* relay.

4. Website fingerprinting attack

4.1 Website Fingerprinting Attack Overview

A website fingerprinting attack (WF) is an attack designed for a local passive eavesdropper to determine the client's endpoint using features from packet sequences. Generally speaking, WF breaks privacy, which is achieved by the proxy, VPN or Tor. This is an application of various machine learning techniques in the field of privacy.

The first appearance of the WF was discussed in paper [7]. This attack has been widely discussed in the researchers` community because it has proven its effectiveness against various privacy enhancing technologies, such as Tor, SSL and VPN.

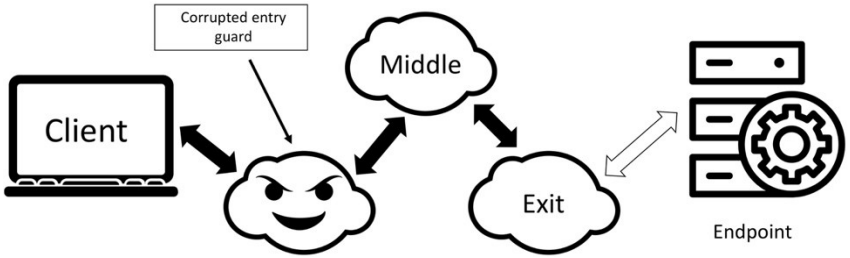

Fig. 2. Configuration of Tor circuit suitable for the WF attack

To perform a WF, an eavesdropper has to simulate users` behavior in the network, using the same conditions as the victim. In case of Tor, an attacker must have a corrupted entry relay (see Fig.2) that will be used for collecting data. The Attacker visits each site from the list and stores all packet sequences related to the request. Afterwards, he uses the traffic for training a classifier in a supervised way. The machine learning problem could be stated as a binary classification problem or multilabel classification problem. In the first case, classifier is trained to answer the question: «If the user visits a site from our list?». The second option is about guessing a particular website that the user visits.

4.2 The Oracle Problem

Since WF works with packet sequences, determining sequences related to the webpage is quite a difficult task. This issue is known as the Oracle problem. Researchers make two major assumptions, which simplify WF a lot: 1) an attacker has such an oracle at his disposal, 2) the victim loads pages one-by-one in a single

tab. The Oracle helps to find precise subsequence of packets from overall captured traffic. Any excess packet sequence sent to classifier can significantly reduce its' accuracy. That is why, splitting the whole sequence is crucially important. Another reason is the user`s web-browsing behavior. The majority of people uses multi-tab browsing instead of loading a page in a single tab, working with it and loading another one. This behavior makes WF difficult in real life.

An Oracle problem for packet sequences has not been solved yet, but Wang proposed a solution for Tor, which can work with a single tab [8]. He considered three-step process of determining correct split in case of single tab browsing between two pages. Wang used Tor cells instead of packets. The first step is making a time based split. The Attacker splits sequences if the time gap between two adjacent cells is greater than some constant, then the sequence is splitted there into two subsequences. If the time gap is too small, classification-based splitting is typically used. Wang used machine learning techniques that decide where to split and whether to split or not. After splitting, the result is ready for further classification. This method achieves quite good accuracy. However, the proposed solution doesn't work with multi-tab browsing and raw packet sequences, narrowing the range of real implementations. Study [9] proposed a time-based way to split traffic traces when the user utilizes 2 open tabs. They classify the first page with 75.9% and second with 40.5% of accuracy.

4.3 Real World Scenario

Overall, the applicability of WF in the real world scenario is still questionable. Users may visit hundreds of thousands of webpages every day. So, can the attacker successfully apply WF in reality? Panchenko et al. [10] checked the attack with a really huge dataset, and their approach outperformed the previous state of the art attack proposed by Wang. To conclude, WF attacks are still a serious threat to anonymous communication systems.

The aim of the current work is to show that an attacker can build a deanonimization system, applying learning libraries for most popular programming languages, which will be able to deanonimize a group of users trying to access the deep web content.

5. Deanonimization system scheme

For the sake of simplicity, we will use as much preconfigured software as possible. In order to deal with deanonimization problem, our system must have two modules. The first module is used for mining Tor data, which will be used for collecting traffic traces. The second is aimed at applying machine learning techniques.

5.1 Data Mining

The data mining module is using various software, which can be easily installed on Mac OS or any Linux distributive. Since the packet traces can be collected on the relay side, or on the client side (the difference is only in the source/destination pair),

we can use data mining module on local machine or on the remote server. We will use local machine for data mining (see Fig. 3). Simple data transformation can be applied for packet traces, to look exactly like those collected on the relay.

The following software must be installed on the machine:

- Tor – free software for enabling anonymous communication,
- Torsocks – free software that allows using any kind of application via the Tor network,
- Wget – a program, which retrieves content from the web server and supports downloading via http, https, ftp,
- Tshark – a free and open packet analyzer; it is used for network troubleshooting, analysis, etc.,
- Mozilla Firefox or Tor Browser – an open web-browser (in case of Mozilla Firefox, it is needed to configure it for using Tor manually).

Nevertheless, any program can be replaced by the specific library. The simplest solution is to use the proposed software. We must have full control over Tor circuits construal to use our own relay. For this purpose, we will use *Stem* Python library, which is freely accessible on the web. *Stem* is a Python controller library for Tor.

We use *Stem* to create Tor circuits through our corrupted entry guard. Without this action, the accuracy of the classifier might become worse, because of different Tor versions on the relays and other reasons. Another option is to modify Tor configuration for using specified entry guards. It is very important to use the same entry guard, which will be used in production.

Tshark is used as the main packet capturing tool. We also use Tshark for extracting TLS records from data. Tshark can be substituted with any library, which supports capturing of TCP packets.

After that, the attacker has to automate the data gathering process. There are two ways to do it, namely, using wget via torsocks, or Mozilla Firefox. In case of wget, an attacker just launches page downloading from the command line, but the use of Mozilla Firefox requires more work. The automation of Mozilla can be done in two ways. The first option is to launch it from the command line and wait while the page is uploading; another one is to use Selenium Webdriver to automate the process.

5.2 Feature Extraction

We can extract features of the traffic at three different levels (see Fig.4) – they are Tor cells, TLS and TCP. At the application level, Tor retranslates data in the fixed size packets called cells. All cells have equal length of 512 bytes and travel throughout the network in TLS records. It is noteworthy that several cells can be packed in a single TLS record. The last level is transport level: TLS record is then fragmented into several TCP packets. TCP packets size is limited by the MTU. Furthermore, several TLS records can be packed into a single TCP packet. However, it is questionable, which level is the most informative from the website

fingerprinting attack perspective. The majority of researchers assume that the most informative level is the cells level.

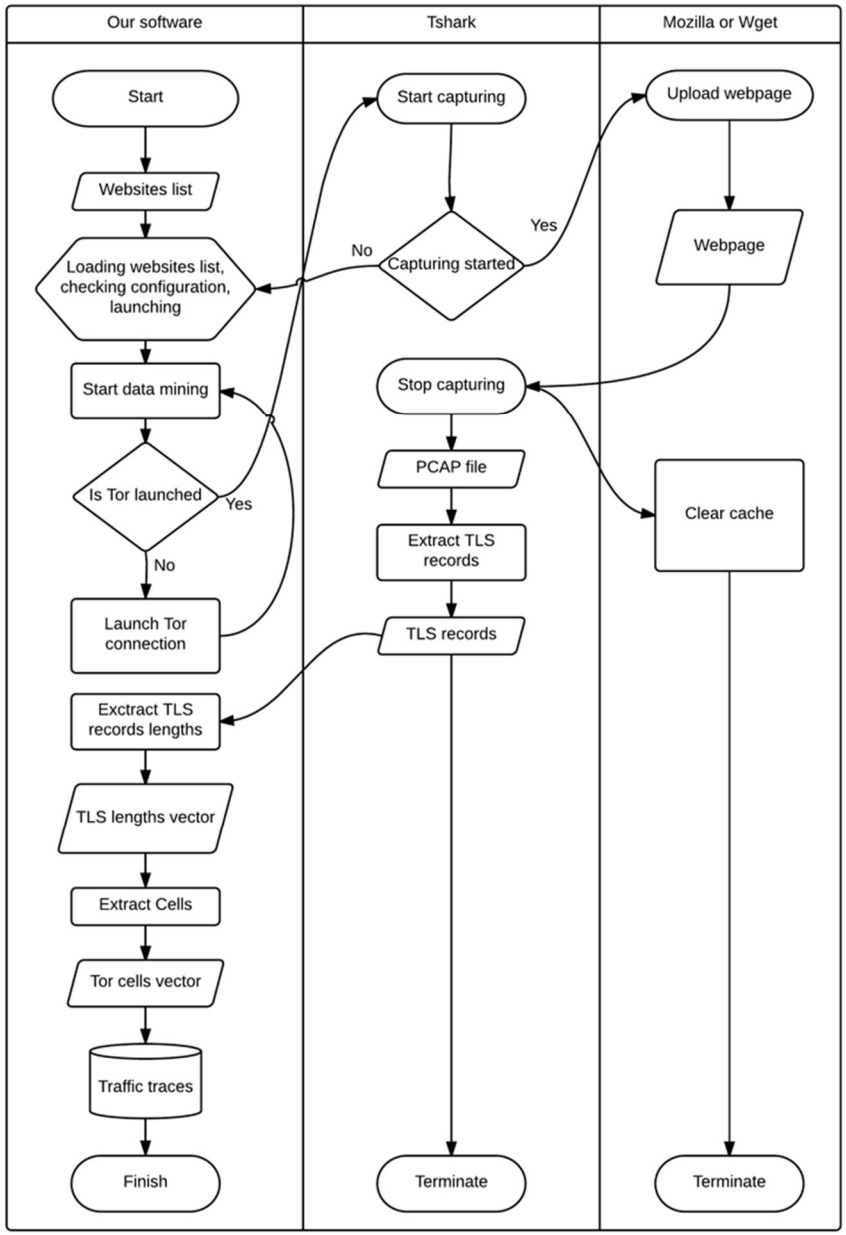

Fig. 3. Data mining process

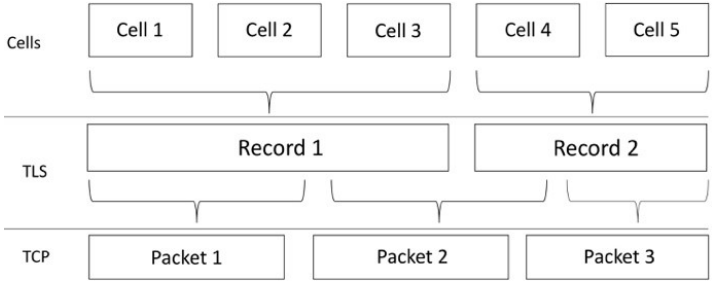

Fig. 4. Information extraction levels

Firstly, the cell traces extraction should be performed in the following way: an attacker must extract TLS records from TCP packets – it could be achieved with the tshark software.

Here the file_name should be substituted by the .pcap file with TCP packets, whereas ouput_file is the desired ouput file with textual representation of TLS records. Hence, a simple regular expression can than be used for length extraction. Once the number is an extended extraction, an attacker should then multiply it by -1 if it is outgoing.

The resulting array of TLS records lengths should then be transformed into Tor cells. An attacker should divide each number by 512 and append to the cells vector as many -1's or 1's as the number of integers found in the result of division. For example, if the length of TLS record is equal to 2048, the resulted cells vector would be [1,1,1,1].

After completion of cell traces extraction, we will have the representation of data in the form of [-1,1,1,1,-1,…]. Such arrays are then used as features, subsequently, the actual webpages are used as labels. However, such arrays have different lengths. Hence, as we are trying to simplify the process, we will append zeros to the end of input vectors because the majority of machine learning algorithms requires the input vectors to have equal lengths. By means of such operation, we will equalize the length of cell vectors.

5.3 Machine Learning Module

For machine learning purposes, we will use *sklearn* Python library, which is the most popular Python library for machine learning. The trained model will be used for classification of new traffic samples.

This module works in a straightforward way. An attacker must train the model using collected cells and then use it as a ready model.

6. Experimental setup

We have implemented such a scheme using Java programming language and Python (Fig. 5). The aim of our experiment is to show that we can deanonimize a small

fraction of users in the real world even if we don't use cutting-edge deanonimization techniques.

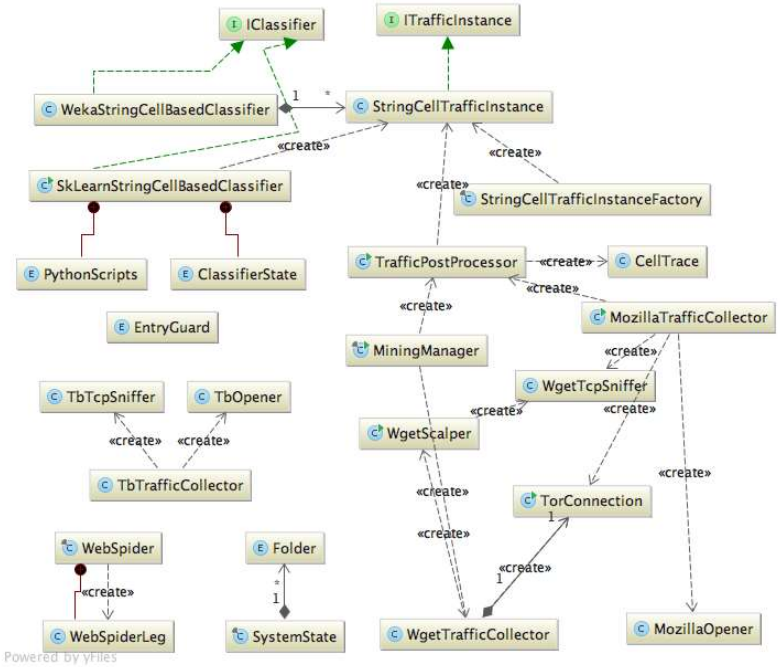

Fig. 5. UML class diagram of traffic collection module

6.1 Experimental Environment

Consider the following situation: the group of terrorists is trying to gain access to illegal content from a small room in the dormitory. The list of resources was provided by the Group-IB cybersecurity company. In our experiment there were three users playing the role of terrorists. Each of them visited the resources from the list according to the following rules: only single tab browsing is used, and the time spent to read the webpage is, at least, 5 seconds. According to the research [10], situation described looks pretty realistic. Such rules allow us to simplify the process of splitting packet sequences and extracting traces.

6.2 Data Gathering

Before trying to deanonimize users, we made a preparation step and collected 80 traffic instances from our list of resources. Such a low number of traffic instances is sufficient, because bigger datasets are not affecting accuracy of classifier on the

same number of websites. We have studied 7 resources related to drugs, weapons and extremism issues.

Our users repeated the process of reading and uploading a webpage 5 times for each webpage from the list. After that, we downloaded collected packet sequences and made the data preprocessing step. We used time-based splitting as was proposed by Wang [11]. After this step, our data became ready for classification.

6.3 Machine Learning Model

Support vector machines (SVM) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. An SVM model represents examples as points in space mapped so that the examples of separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted as belonging to a category based on which side of the gap they fall to.

We used the NuSVC machine learning algorithm with default hyperparameters from the *sklearn* library. NuSVC is Nu-Support vector classification based on the support vector machines. This algorithm uses a parameter to control the number of support vectors, where the parameter is an upper bound of the fraction of training errors and lower bound of the fraction of support vectors.

7. Evaluation

7.1 Evalution metrics

- True positives (tp) - equal with hit.
- False positives (fp) - type I error, equal with correct rejection.
- False negatives (fn) - type II error.
- Precision - the ratio tp / (tp + fp); there is an intuitive ability of the classifier to avoid labelling a negative sample with the positive label.
- Recall - the ratio tp / (tp + fn); there is an intuitive ability of the classifier to find all the positive samples (the best is 1, the worst is 0).
- F1-score - a weighted average of the precision and recall (its best value is 1, the worst is 0) = 2 * (precision * recall) / (precision + recall).
- Score – the subset accuracy returned in a multilabel classification. If the entire set of predicted labels for a sample strictly matches the true set of labels, then the subset accuracy is 1.0, otherwise it is 0.0.

7.2 Experimental results

We have performed the classifier evaluation using a built-in *sklearn* function. For ethical reasons documented in Tor ethical research [12], we've anonymized the websites used in the experiment.

Our simple model has achieved results presented in table 2.

Table 2. Classifier evaluation

Website	Precision	Recall	F1-score
Site_1	1.00	1.00	1.00
Site_2	0.80	0.80	0.80
Site_3	0.80	0.80	0.80
Site_4	0.50	0.40	0.44
Site_5	1.00	1.00	1.00
Site_6	0.38	0.60	0.46
Site_7	0.67	0.40	0.50
Avg/total	0.73	0.71	0.72

Overall, the total score of the classifier = 0.714

These results are not outstanding in comparison with the state-of-the-art techniques, but they show that we can deanonimize users with the help of a relatively simple program and achieve sufficient accuracy.

8. Conclusion

It was shown that the attacker without cutting-edge machine learning techniques can apply website fingerprinting. If the attacker has enough experience and technical competence, he will be able to build such a system and use it for the purpose of deanonimization. Moreover, the proposed solution will work better if the attacker sniffs Wi-Fi or other local network, because it is very easy for him to find Tor related traffic and collect traces. In this case, the deanonimization is targeted and easily implemented.

9. Future work

In our future work, we are going to solve the Oracle problem using the recurrent neural networks and test them in the field of website fingerprinting attacks. Next, we are going to build a cloud application using state-of-the art techniques and results based on Recurrent Neural Networks research.

The main purpose of solving the Oracle problem is to have a pretty accurate splitting algorithm, which will allow to use WF attacks even with the multi-tab browsing.

References

[1]. S.M. Avdoshin, A.V. Lazarenko. [Technology of anonymous networks]. Informacionnye tehnologii [Information Technologies], vol. 22, №4, pp. 284-291, 2016 (in Russian).

[2]. R. Dingledine, N. Mathewson, P. Syverson. “Tor: The Second-Generation Onion Router”. In Proceedings of the 13th USENIX Security Symposium, August 2004 (online publication). Available at: http://www.onion-router.net/Publications/tor-design.pdf, accessed 12.07.2016.

[3]. Relays and bridges in the network (online publication). Tor METRICS [Official website]. Available at: https://metrics.torproject.org/networksize.html, accessed 12.07.2016.

[4]. The NSA’s Been Trying to Hack into Tor’s Anonymous Internet For Years (online publication). Gizmodo [Official website]. Available at: http://gizmodo.com/the-nsas-been-trying-to-hack-into-tors-anonymous-inte-1441153819, accessed 12.07.2016.

[5]. Zakupka No0373100088714000008 (online publication). Gosudarstvennie zakupky [State Procurements] [Official website]. Available at: http://zakupki.gov.ru/epz/order/notice/zkk44/view/common-info.html?regNumber=0373100088714000008, accessed 12.07.2016 (in Russian).

[6]. S.M. Avdoshin, A.V. Lazarenko, [Tor Users Deanonimization Methods]. Informacionnye tehnologii [Information Technologies], vol. 22, №5, pp. 362-372, 2016 (in Russian).

[7]. X. Cai, X.C. Zhang, B. Joshy, R. Johnson. Touching from a Distance: Website Fingerprinting Attacks and Defenses (online publication). Available at: http://www3.cs.stonybrook.edu/~xcai/fp.pdf, accessed 12.07.2016.

[8]. T. Wang, Website Fingerprinting: Attacks and Defenses, PhD Thesis (online publication), 2015. Available at: https://uwspace.uwaterloo.ca/bitstream/handle/10012/10123/Wang_Tao.pdf?sequence=3, accessed 12.07.2016.

[9]. X.Gu, M.Yang, J.Luo. A Novel Website Fingerprinting Attack Against Multi-Tab Browsing Behavior (online publication). In Computer Supported Cooperative Work in Design (CSWD), 2015. Available at: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=7230964&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D7230964, accessed: 12.07.2016.

[10]. A. Panchenko, F. Lanze, A. Zinnden, M. Henze, J. Pannekamp, K. Wehrle, T. Engel. Website Fingerprinting at Internet Scale (online publication). Available at: https://www.comsys.rwth-aachen.de/fileadmin/papers/2016/2016-panchenko-ndss-fingerprinting.pdf, accessed 12.07.2016.

[11]. J.Nielsen. How Long Do Users Stay on Web Pages (online publication), Available at: https://www.nngroup.com/articles/how-long-do-users-stay-on-web-pages/, accessed 12.07.2016.

[12]. Ethical Tor Research: Guidelines (online publication). Available at: https://blog.torproject.org/blog/ethical-tor-research-guidelines, accessed 12.07.2016.

Система деанонимизации пользователей теневого интернета

С.М. Авдошин <savdoshin@hse.ru>
А.В. Лазаренко <avlazarenko @edu.hse.ru>
Департамент программной инженерии,
Национальный исследовательский университет “Высшая школа экономики”,
101000, Россия, г. Москва, ул. Мясницкая, д. 20.

Аннотация. Технологии обеспечения пользовательской приватности являются неотъемлемой частью жизни современных людей. Они востребованны широким пользовательским сегментом. Однако такие инструменты зачастую используются для мошеннической и нелегальной деятельности. В современном мире есть много сетей и технологий, которые предоставляют анонимный доступ к ресурсам сети. Наиболее

распространенной и широко используемой анонимной сетью является Tor. При этом именно Tor является основным инструментом многочисленных хакеров, торговцев наркотиками и оружием. Настоящая статья фокусируется на деанонимизации пользователей Tor с применением доступных в интернете технологий и базового алгоритма машинного обучения. Цель работы – показать, что деанонимизация небольшого количества пользователей возможна без использования большого количества вычислительных ресурсов. В начале работы представлен обзор различных анонимных сетей. Затем - различные методы деанонимизации: анализ траффика, тайминг атаки, атаки на уровне автономных систем. Построена классификация атак по ресурсам, необходимым атакующим для успешного применения. Для реализации была выбрана website fingerprinting атака. Эта атака требует наименьшего количества ресурсов для ее использования и внедрения в сеть Tor с целью успешной деанонимизации пользователей. Описан эксперимент использования website fingerprinting атаки. Список отслеживаемых в эксперименте ресурсов был получен от компании, специализирующейся в области информационной безопасности. Эксперимент проводился в одной комнате при участи 5 человек и одного входного узла. Была достигнута точность классификации просматриваемых страниц равная 70% процентам. Задача деанонимизации крайне важна для национальной безопасности, что подчеркивает актуальность проведенного исследования.

Ключевые слова: Tor; деанонимизация; website fingerprinting; анализ трафика; анонимная сеть; теневой интернет.

DOI: 10.15514/ISPRAS-2016-28(3)-2

Для цитирования: Авдошин С.М., Лазаренко А.В.. Система деанонимизации пользователей теневого интернета. Труды ИСП РАН, том 28, вып. 3, 2016 г. стр. 21-34 (на английском). DOI: 10.15514/ISPRAS-2016-28(3)-2

Список литературы

[1]. Авдошин С.М., Лазаренко А.В. Технология анонимных сетей // Информационные технологии. 2016. Т. 22, № 4, стр. 284-291.

[2]. R. Dingledine, N. Mathewson, P. Syverson. Tor: The Second-Generation Onion Router, in Proceedings of the 13th USENIX Security Symposium, August 2004. URL: http://www.onion-router.net/Publications/tor-design.pdf, 12.07.2016.

[3]. Relays and bridges in the network (online). Tor METRICS [Official website], Доступно по ссылке: https://metrics.torproject.org/networksize.html, 12.07.2016.

[4]. The NSA's Been Trying to Hack into Tor's Anonymous Internet For Years (online), Gizmodo [Official website], Доступно по ссылке: http://gizmodo.com/the-nsas-been-trying-to-hack-into-tors-anonymous-inte-1441153819, 12.07.2016.

[5]. Закупка № 0373100088714000008 (online). Государственные закупки [Официальный сайт]. Доступно по ссылке: http://zakupki.gov.ru/epz/order/notice/zkk44/view/common-info.html?regNumber=0373100088714000008, 12.07.2016.

[6]. Авдошин С.М., Лазаренко А.В. Методы деанонимизации пользователей TOR // Информационные технологии. 2016. Т. 22, № 5, стр. 362-372.

[7]. X. Cai, X.C. Zhang, B. Joshy, R. Johnson. Touching from a Distance: Website Fingerprinting Attacks and Defenses (online). Доступно по ссылке: http://www3.cs.stonybrook.edu/~xcai/fp.pdf, 12.07.2016.

[8]. T. Wang, "Website Fingerprinting: Attacks and Defenses", PhD Thesis, 2015 (online). Доступно по ссылке: https://uwspace.uwaterloo.ca/bitstream/handle/10012/10123/Wang_Tao.pdf?sequence=3 12.07.2016.

[9]. X.Gu, M.Yang, J.Luo. A Novel Website Fingerprinting Attack Against Multi-Tab Browsing Behavior, in Computer Supported Cooperative Work in Design (CSWD), 2015 (online). Доступно по ссылке: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=7230964&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D7230964, accessed: 12.07.2016.

[10]. A. Panchenko, F. Lanze, A. Zinnden, M. Henze, J. Pannekamp, K. Wehrle, T. Engel. , Website Fingerprinting at Internet Scale (online). Доступно по ссылке: https://www.comsys.rwth-aachen.de/fileadmin/papers/2016/2016-panchenko-ndss-fingerprinting.pdf, accessed: 12.07.2016.

[11]. J.Nielsen. How Long Do Users Stay on Web Pages (online). Доступно по ссылке: https://www.nngroup.com/articles/how-long-do-users-stay-on-web-pages/, accessed: 12.07.2016.

[12]. Ethical Tor Research: Guidelines (online). Доступно по ссылке: https://blog.torproject.org/blog/ethical-tor-research-guidelines, accessed: 12.07.2016.