ISSN: 2321-2152

IJMECE

International Journal of modern electronics and communication engineering

E-Mail
editor.ijmece@gmail.com
editor@ijmece.com

www.ijmece.com

Darknet Traffic Analysis: Examining How Modified Tor Traffic Affects the Classification of Onion Service Traffic

Karakkayala sravan sai kumar
Student
Department of CSE, VIT Vellore, Tamilnadu, India

ABSTRACT:

For the purposes of traffic shaping and monitoring, network traffic classification is crucial. Privacy-preserving technology have become increasingly important in the past 20 years as privacy concerns have grown. The Tor network is a well-liked method of achieving online anonymity because it offers its users privacy and enables anonymous services called Onion Services. The fact that this anonymity is commonly abused, particularly with Onion Services, prompts governments and authorities to de-anonymize them. Therefore, with a focus on three primary contributions, we attempt to classify the traffic of the Onion Service in this study. Initially, we attempt to separate Onion Services traffic from various other forms of Tor communication. Our methods have an accuracy of >99% in identifying Onion Service traffic. However, the Tor traffic may be altered in a number of ways to prevent its information loss. We assess the performance of our methods in the second contribution once the Tor traffic has undergone such changes. These circumstances reduce the ability to differentiate Onion Service traffic (in some situations, the accuracy lowers by more than 15%), according to our experimental data. Finally, we determine the most important feature configurations for our classification task and assess their influence.

INTRODUCTION

Tor is an incognito network that uses a number of intermediate nodes to route communications in order to conceal its users' identities. Additionally, Tor facilitates the delivery of anonymous services with the top-level domain name. Onion, which are also referred to as hidden services. Security professionals, network defences, and law enforcement organizations are better able to distinguish traffic sent over Tor from other secured and non-encrypted traffic because of Tor's capacity to function as a government circumvention tool. Classifying traffic sent via Tor from non-Tor traffic, different application kinds in Tor traffic, and additional anonymity network traffic like I2P and Web-mix traffic are a few examples of the tasks that have been attempted. In this study, however, we want to use traffic analysis to investigate if Onion Service traffic can be distinguished from regular Tor traffic. Three research questions serve as the cornerstone of our investigation. We attempt to respond to the query first. To browse an online resource on the Internet using Tor, a typical Tor circuit is made up of three Onion nodes. An Onion Account circuit is made up of six Tor nodes and is the sole method for accessing an Onion Service. We presume that while all communication in both of these circuits (the normal Tor and the Onion Service) is securely transmitted, we may use the metadata leak information (such as packet size, direction, and timestamps) to find distinctive patterns that make them different. Aside from hosting illicit websites, onion services have also been utilized as botnet control and command (C&C) servers in more recent times. Thus, from the standpoint of both governments and law enforcement, they wish to monitor, block, and control the traffic on the onion service. It may be beneficial for organizations to limit access to certain domains in order to safeguard their systems from possible threats and malicious actors, such as hackers. Thus, methods for detecting Onion Account traffic can be helpful for two primary reasons: 1. These methods can serve as a foundation for fingerprinting Onion Services. 2. They can help secure sensitive and private systems by limiting Onion Service traffic.The second thing we do is try to look into the same issue in multiple environments. We specifically try. It is possible to alter Tor's traffic patterns by using specific strategies. Examples of such methods include introducing padding, utilizing false bursts and complications, and dividing the stream. These methods1 were created with the goal of discouraging Tor traffic's information leaking. Answering RQ2 is crucial because it allows us to verify if the conclusions drawn from RQ1 will be valid as and when the Tor traffic is altered in this way.In the event that these changes are implemented in the future, it will be clear that they are ineffective in hiding Onion Service traffic if we can still identify it. The legitimacy of earlier efforts, such as a configuration with those alterations applied, is called into doubt if the changes do have a classifiable impact on the onion service. We contend that RQ2 is worth assessing as its results may lead to new directions for Tor traffic categorization research.

RELATED WORK

"Enhancing Tor’s performance using real-time traffic classification"

M. Al Sabah, K. Bauer and I. Goldberg, Oct. 2012.

Tor is a low latency network that protects anonymity and helps users stay private online. It is made up of routers that are run by volunteers from nearly the world and serve a millions of customers daily. Tor experiences performance problems as a result of traffic and an insufficient relay-to-client ratio, which may deter its broader use and offer less anonymity to all users in general. Our goal is to enhance Tor's performance by establishing distinct service classes for its traffic. Even though interactive web surfing accounts for a great deal of Tor traffic, we acknowledge that a comparatively tiny quantity of mass downloading unfairly uses up Tor's limited capacity. Tor currently provides these traffic classes with the same Quantity of Service (QoS), however these classes should not be granted the same QoS because they have distinct time and capacity limits. We present and assess Dafter, a machine-learning based method that is able to classify Tor's encrypted routes by client in real time and then provide different classes of service depending on the application. Our tests demonstrate that we can distinguish between circuits we created on the real Tor networks with an incredibly high accuracy of over 95%. With our straightforward methods, we demonstrate that our real-time categorization combined with QoS can significantly enhance the user interface of Tor consumers, as our interactive users report a 75% increase in interactivity and an 86% decrease in median download times.

"Tor traffic classification from raw packet header using convolutional neural network"

M. Kim and A. Anpalagan, Jul. 2018.

Since network traffic is increasing at an exponential rate, traffic analysis and categorization are essential for efficient allocation of resources and network administration. However, encrypted communication, like Tor, one of the most widely used encryption schemes, is making this task more challenging as security technologies advance. In this study, a method for classifying Tor traffic using a convolutional neural network structure with hexadecimal raw packet headers is proposed. When compared to competing machine learning methods, our method demonstrates notable accuracy. The UNB-CIC Tor system traffic information is used to publicly validate this approach. Experimental results indicate that our method is 99.3% accurate in classifying fractionized Tor/non-Tor traffic.

"Inferring application type information from Tor encrypted traffic"

G. He, M. Yang, J. Luo and X. Gu, Nov. 2014.

Tor is a well-known method for communicating anonymously and protecting users' privacy online. In order to conceal part of the users' private information, like the type of program that is running (Web, P2P, FTP, Others), it supports TCP services and compresses application data into encryption cells of similar size. Knowing the sorts of applications is dangerous as it may be used to lower the level of anonymity and make other attacks easier. Sadly, though, some application behaviours cannot be hidden by the Tor architecture as it is. As an illustration, P2P programs typically download and upload data at the same time, and Tor traffic likewise exhibits similar behaviour. This discovery prompts us to look at a novel attack against Tor called the traffic classification threat, which can identify different kinds of applications from Tor traffic. An attacker uses an effective machine learning technique to model various application types after carefully choosing certain flow parameters, such as burst volumes and directions, to depict the behaviour of the application. The target's Tor traffic may then be categorized using these well-established models, which can also be used to determine the application type. Our tests confirm the viability and efficacy of the traffic categorization attack, which we have deployed on Tor.

"Anonymity services for I2P Mononym: Classifying in the dark (web)"

A. Monteiro, D. Cuonzo, G. Aceto and A. Escape, May 2020.

With applications in security, administration, traffic engineering, and research and development, classified traffic (TC) is a crucial tool for a number of activities. Privacy-preserving techniques and tools that encrypt the conversation's content and, in the case of anonymity technologies, also conceal the communication's source, destination, and nature hinder or stop this process. Using five different machine learning classifiers, we present classification results in this paper using a public dataset published in 2017. The goal is to determine the extent to which the traffic of other anonymity tools can be distinguished from the traffic of the particular concealment tool (and the traffic it conceals).

First, the impacts of temporal-related

characteristics and feature significance on the network are examined, and flow-based TC is taken into consideration. Furthermore, the significance of finer-grained characteristics is established, including the (joint) spectrum of packet length (and inter-arrival durations). The analysis of anonymous networks' "early" TC is done successively. The results demonstrate that the anonymity networks under consideration (Tor, I2P, and Mononym) can be readily identified (with a specificity of 99.87% and 99.80% for flow-based while early-TC, respectively), as well as identifying the precise application that is causing the traffic (with a certainty of 73.99% and 66.76% for flow-based while early-TC, respectively).

"Toward an efficient website fingerprinting defence"

M. Juarez, M. Imani, M. Perry, C. Diaz and M. Wright, Sep. 2016.

Website fingerprinting attacks compare the observed traffic with prepared online traffic templates, allowing a passive eavesdropper to discover the user's otherwise anonymized web surfing history. The proposed defenses against these assaults are too expensive in terms of additional latency and bandwidth overhead to be deployed in real-world systems. Furthermore, these defenses have been attacked for requiring unsustainable assault circumstances in the assessment environment, even if they have been shown to be effective against attacks. We provide a new, lightweight defense in this work that is based on adaptive padding and offers a high enough degree of protection against website fingerprinting, especially under actual assessment settings. In a closed-world scenario, this defense introduces zero latency penalty and less than 60% bandwidth overhead while reducing the effectiveness of the modern assault from 91% to 20%. The assault accuracy is just 1% in an open environment and decreases more as the total number of locations increases.

METHODOLOGY

1. Upload Dataset: This module loads and displays the original TOR dataset for NO DEFENCE as well as the WTFPAD dataset.
2. Corelation Features Selection:The selected features significance graph is shown below. This module loads WTFPAD labels for classes and then defines code to choose 50 important features from the dataset using the Information Gain technique.

3. Preprocess & Split Dataset**:** This module separates 20% of the data for testing and 80% for training.

4. Original KNN:Train the KNN algorithm using the original dataset without defines.

5. Original Random Forest:Use the original no-defines dataset to train the Random Forest algorithm.

6. Original SVM: Use an original dataset without defines to train the SVM algorithm.

7. Defence KNN: Utilize defines datasets to train the KNN algorithm.

8. Defence Random Forest: Use the defence dataset to train the Random Forest algorithm.

9. Defence SVM: Develop the SVM algorithm using defines datasets.

10. WTFPAD KNN: Use a modified dataset from WTFPAD to train the KNN algorithm.

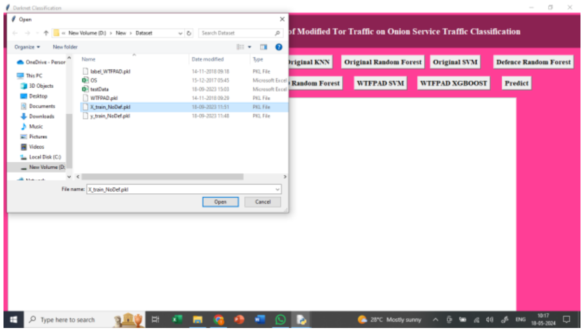
11. WTFPAD Random Forest:Use a modified dataset from WTFPAD to train the Random Forest algorithm.

12. WTFPAD SVM:Use an original dataset without defence to train the SVM algorithm.

13. WTFPAD ADABOOST: Use a modified dataset from WTFPAD to train the ADABOOST algorithm.

14. Predict**:** This module reads test networks data and uses the ADABOOST algorithm extension to categorize network traffic as either Tor or Onion applications. Test results are displayed in square brackets, and the anticipated service type is displayed following the arrow sign.

RESULT AND DISCUSSION

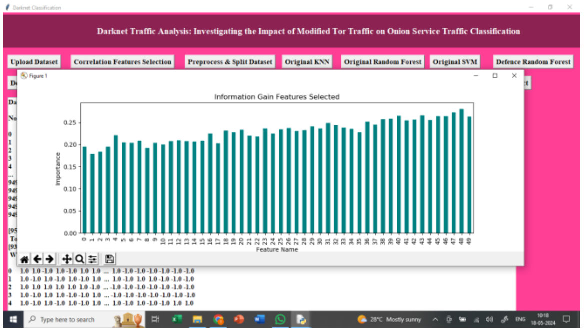


Name	Date modified	Type
label_WTFPAD.pkl	14-11-2018 09:18	PKL File
OS	15-12-2017 05:45	Microsoft Excel
testData	18-09-2023 15:03	Microsoft Excel
WTFPAD.pkl	14-11-2018 09:29	PKL File
X_train_NoDef.pkl	18-09-2023 11:51	PKL File
y_train_NoDef.pkl	18-09-2023 11:48	PKL File

File name: X_train_NoDef.pkl

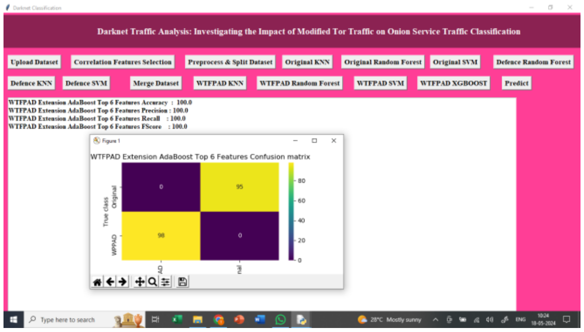
The aforementioned result loads and displays the original TOR dataset with NO

DEFENCE, followed by the WTFPAD dataset.



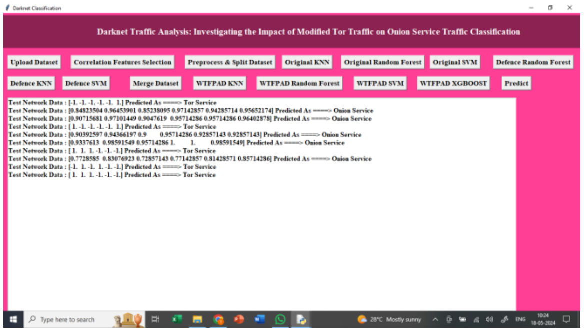
```
0   1.0  1.0 -1.0  1.0 -1.0  1.0  1.0  ...  1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0
1   1.0 -1.0  1.0 -1.0  1.0 -1.0  1.0  ...  1.0 -1.0  1.0  1.0 -1.0 -1.0 -1.0
2   1.0  1.0  1.0  1.0  1.0  1.0 -1.0  ... -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0
3   1.0 -1.0  1.0  1.0 -1.0 -1.0  1.0  ... -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0
4   1.0 -1.0  1.0 -1.0  1.0 -1.0  1.0  ...  1.0 -1.0  1.0 -1.0 -1.0  1.0  1.0
```

The x-axis in the following graph shows the names of the characteristics, while the y-axis shows the significance or pertinent score value.



```
WTFPAD Extension AdaBoost Top 6 Features Accuracy  :  100.0
WTFPAD Extension AdaBoost Top 6 Features Precision : 100.0
WTFPAD Extension AdaBoost Top 6 Features Recall    : 100.0
WTFPAD Extension AdaBoost Top 6 Features FScore    : 100.0
```

In the given result, 100% accuracy was obtained after training the ADABOOST algorithm.



```
Test Network Data : [-1. -1. -1. -1. -1.  1.] Predicted As ====> Tor Service
Test Network Data : [0.84823504 0.96453901 0.85238095 0.97142857 0.94285714 0.95652174] Predicted As ====> Onion Service
Test Network Data : [0.90715681 0.97101449 0.9047619  0.95714286 0.95714286 0.96402878] Predicted As ====> Onion Service
Test Network Data : [ 1. -1. -1. -1. -1.  1.] Predicted As ====> Tor Service
Test Network Data : [0.90392597 0.94366197 0.9        0.95714286 0.92857143 0.92857143] Predicted As ====> Onion Service
Test Network Data : [0.9337613  0.98591549 0.95714286 1.         1.         0.98591549] Predicted As ====> Onion Service
Test Network Data : [ 1.  1.  1. -1. -1. -1.] Predicted As ====> Tor Service
Test Network Data : [0.7728585  0.83076923 0.72857143 0.77142857 0.81428571 0.85714286] Predicted As ====> Onion Service
Test Network Data : [-1.  1. -1.  1. -1. -1.] Predicted As ====> Tor Service
Test Network Data : [ 1.  1.  1. -1. -1. -1.] Predicted As ====> Tor Service
```

In the result above, test network data is read, and network traffic is then classified as Tor or Onion services using the extension ADABOOST algorithm. Test results are displayed in square brackets, and the anticipated service type is displayed following the arrow sign.

CONCLUSION

Three research issues about the classification of traffic on the onion service were addressed in this study. We assessed the suitability of autonomous machine learning systems for separating Tor traffic from Onion Service traffic. From each traffic trace, we retrieved fifty characteristics, which we then fed into machine learning classifiers. Our findings demonstrated that the KNN, RF, and SVM classification algorithms can separate Tor traffic from Onion Service traffic with 99% accuracy. Then, we attempted to determine if the classification of Tor traffic is impacted by cutting-edge Website Fingerprinting defines. We assessed the effects of these defences on the Onion Service traffic categorization. These defines incorporate various alterations in an attempt to disguise information leakage from traffic.Our tests demonstrated that our feature set, when paired with the aforementioned classifiers, lowers the performance for classifying Onion Service data. We did note, though, that the altered

Tor traffic can still be identified. In addition, we employed three feature selection criteria to determine the best features for this task: Fisher Score, Pearson's correlation, and information gain. These key characteristics were able to distinguish between Tor and Onion Service traffic with >98% accuracy. When improved Tor traffic footprints were applied, however, they were unable to provide such positive outcomes.

REFERENCES

[1] R. Dingledine, N. Mathewson, and P. Syverson, ‘‘Tor: The second generation onion router,’’ in Proc. 13th USENIX Secure. Symp. (SSYM), San Diego, CA, USA, Aug. 2004, pp. 303–320.

[2] M. Al Sabah, K. Bauer, and I. Goldberg, ‘‘Enhancing Tor’s performance using real-time traffic classification,’’ in Proc. ACM Conf. Comput. Commun. Secure. (CCS), New York, NY, USA, Oct. 2012, pp. 73–84.

[3] A. H. Lashkari, G. D. Gil, M. S. I. Mamun, and A. A. Ghorbani, ‘‘Characterization of Tor traffic using time based features,’’ in Proc. 3rd Int. Conf. Inf. Syst. Secure. Privacy (ICISSP), Porto, Portugal, Feb. 2017, pp. 253–262.

[4] M. Kim and A. Anpalagan, ‘‘Tor traffic classification from raw packet header using convolutional neural network,’’ in Proc. 1st IEEE Int. Conf. Know. Innova. Invention (ICKII), Jeju Island, South Korea, Jul. 2018, pp. 187–190.

[5] G. He, M. Yang, J. Luo, and X. Gu, ‘‘Inferring application type information from Tor encrypted traffic,’’ in Proc. 2nd Int. Conf. Adv. Cloud Big Data (CBD), Washington, DC, USA, Nov. 2014, pp. 220–227.

[6] A. Monteiro, D. Cuonzo, G. Aceto, and A. Escape, ‘‘Anonymity services tor, I2P, Mononym: Classifying in the dark (web),’’ IEEE Trans. Dependable Secure Comput., vol. 17, no. 3, pp. 662–675, May 2020.

[7] (May 2017). Wry Ransomware Analysis. Accessed: Apr. 26, 2023. [Online]. Available: https://www.secureworks.com/research/wcryransomware-analysis

[8] (Jul. 2019). Keeping a Hidden Identity: Mirai C&Cs in Tor Network. Accessed: Apr. 26, 2023. [Online]. Available: https://blog.trendmicro. com/trendlabs-security-intelligence/keeping-a-hidden-identity-mirai-ccsin-tor-network/

[9] (Nov. 2014). Global Action Against Dark Markets on Tor Network. Accessed: Aug. 4, 2020. [Online]. Available: https://www.europol. europa.eu/newsroom/news/global-action-against-dark-markets-tornetwork

[10] M. Juarez, M. Imani, M. Perry, C. Diaz, and M. Wright, ''Toward an efficient website fingerprinting defence,'' in Proc. 21st Eur. Symp. Res. Comput. Secure. (ESORICS), Heraklion, Greece, Sep. 2016, pp. 27–46.

[11] T. Wang and I. Goldberg, ''Walkie-talkie: An efficient defence against passive website fingerprinting attacks,'' in Proc. 26th USENIX Secure. Symp. (SEC), Vancouver, BC, Canada, Aug. 2017, pp. 1375–1390.

[12] W. De la Cadena, A. Mitzva, J. Hiller, J. Pennekamp, S. Reuter, J. Filter, T. Engel, K. Wehrle, and A. Panchenko, ''Traffic Sliver: Fighting website fingerprinting attacks with traffic splitting,'' in Proc. ACM SIGSAC Conf. Comput. Commun. Secure. (CCS), New York, NY, USA, Nov. 2020, pp. 1971–1985.

[13] J. Hayes and G. Danez is, ''k-fingerprinting: A robust scalable website fingerprinting technique,'' in Proc. 25th USENIX Conf. Secure. Symp. (SEC), Austin, TX, USA, Aug. 2016, pp. 1187–1203.

[14] X. Bai, Y. Zhang, and X. Niu, ''Traffic identification of Tor and web mix,'' in Proc. 8th Int. Conf. Intel. Syst. Design Appl. (ISDA), Kaohsiung, Taiwan, vol. 1, Nov. 2008, pp. 548–551.

[15] O. Berthold, H. Federate, and S. Koppell, ''Web Mixes: A system for anonymous and unobservable Internet access,'' in Proc. Int. Workshop Design Issues Anonymity Unobservability, in Lecture Notes in Computer Science, vol. 2009, H. Federate, Ed., Berkeley, CA, USA, Jul. 2000, pp. 115–129.

[16] B. Zan tout and R. Harat, "I2P data communication system", *Proc. 10th Int. Conf. Newt. (ICN)*, pp. 401-409, Jan. 2011.

[17] P. Surinam, M. Imani, M. Juarez and M. Wright, "Deep fingerprinting: Undermining website fingerprinting defences with deep learning", *Proc. ACM SIGSAC Conf. Comput. Commun. Secure. (CCS)*, pp. 1928-1943, Oct. 2018.

[18] R. Overdorf, M. Juárez, G. Acar, R. Greenstead and C. Díaz, "How unique is you. Onion?: An analysis of the fingerprint

ability of Tor onion services", *Proc. ACM SIGSAC Conf. Comput. Commun. Secure. (CCS)*, pp. 2021-2036, Oct. 2017.

[19] I. H. Witten, E. Frank and M. A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, San Francisco, CA, USA: Morgan Kaufmann, 2011.

[20] X. He, D. Cai and P. Niyogi, "Laplacian score for feature selection", *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, pp. 507-514, Dec. 2005.

[21] M. Gan and L. Zhang, "Iteratively local Fisher score for feature selection", *Appl. Intel.*, vol. 51, pp. 6167-6181, Aug. 2021.