A User Prediction and Identification System for Tor Networks Using ARIMA Model

Tetsuya Oda[1(✉)], Miralda Cuka[3], Ryoichiro Obukata[3], Makoto Ikeda[2], and Leonard Barolli[2]

[1] Department of Information and Computer Engineering, Okayama University of Science (OUS), 1-1 Ridaicho, Kita-ku, Okayama 700–0005, Japan
oda.tetsuya.fit@gmail.com

[2] Department of Information and Communication Engineering, Fukuoka Institute of Technology (FIT), 3-30-1 Wajiro-Higashi, Higashi-Ku, Fukuoka 811–0295, Japan
makoto.ikd@acm.org, barolli@fit.ac.jp

[3] Graduate School of Engineering, Fukuoka Institute of Technology (FIT), 3-30-1 Wajiro-Higashi, Higashi-Ku, Fukuoka 811–0295, Japan
mcuka91@gmail.com, obukenkyuu@gmail.com

Abstract. Due to the amount of anonymity afforded to users of the Tor infrastructure, Tor has become a useful tool for malicious users. With Tor, the users are able to compromise the non-repudiation principle of computer security. Also, the potentially hackers may launch attacks such as DDoS or identity theft behind Tor. For this reason, there are needed new systems and models to detect the intrusion in Tor networks. In this paper, we present the application of Autoregression Integrated Moving Average (ARIMA) for prediction of user behavior in Tor networks. We constructed a Tor server and a Deep Web browser (Tor client) in our laboratory. Then, the client sends the data browsing to the Tor server using the Tor network. We used Wireshark Network Analyzer to get the data and then used the ARIMA model to make the prediction. The simulation results show that proposed system has a good prediction of user behavior in Tor networks.

1 Introduction

The Onion Router (Tor) [1,2] is an implementation of an Onion Routing network, where users expect a large degree of privacy. This privacy is reflected in the perfect forward secrecy exhibited by Tor connections such that traffic captured at any single network location during transit only uncovers the previous and next waypoints [3]. Due to the amount of anonymity afforded to users of the Tor infrastructure, Tor has become a useful tool for malicious users. With Tor, the users are able to compromise the non-repudiation principle of computer security. Also, the potentially hackers may launch attacks such as DDoS or identity theft behind Tor.

L. Barolli et al. (eds.), *Advances in Internetworking, Data & Web Technologies*,
Lecture Notes on Data Engineering and Communications Technologies 6,
DOI 10.1007/978-3-319-59463-7_9

The Tor has been designed to make it possible for users to surf the Internet anonymously, so their activities and location cannot be discovered by government agencies, corporations, or anyone else. Compared with other anonymizers Tor is more popular and has more visibility in the academic and hacker communities. Tor is a low-latency, circuit-based and privacy-preserving anonymizing platform and network. It is one of several systems that have been developed to provide Internet users with a high level of privacy and anonymity in order to cope with the censorship measures taken by authorities and to protect against the constantly increasing threats to these two key security properties.

There are two main approaches to the design of Intrusion Detection Systems (IDSs). In a misuse detection based IDS, intrusions are detected by looking for activities that correspond to known signatures of intrusion or vulnerabilities. On the other hand, anomaly detection based IDS detects intrusions by searching for abnormal network traffic. The abnormal traffic pattern can be defined either as the violation of accepted thresholds for the legitimate profile developed for the normal behavior.

In [4], the authors designed and implemented TorWard, which integrates an Intrusion Detection System (IDS) at Tor exit routers for Tor malicious traffic discovery and classification. The system can avoid legal and administrative complaints and allows the investigation to be performed in a sensitive environment such as a university campus. An IDS is used to discover and classify malicious traffic. The authors performed comprehensive analysis and extensive real-world experiments to validate the feasibility and effectiveness of TorWard.

One of the most commonly used approaches in expert system based on intrusion detection is a rule-based analysis using soft computing techniques such Fuzzy Logic (FL), Artificial Neural Networks (ANNs), Probabilistic Reasoning (PR), and Genetic Algorithms (GAs). They are good approaches capable of finding patterns for abnormal and normal behavior. In some studies, the NNs have been implemented with the capability to detect normal and attack connections [5].

In [6], a specific combination of two NN learning algorithms, the Error Backpropagation and the Levenberg-Marquardt algorithm, is used to train an artificial NN to model the boundaries of the clusters of recorded normal behavior. It is shown that the training dataset, consisting of a combination of recorded normal instances and artificially generated intrusion instances, successfully guides the NN towards learning the complex and irregular cluster boundary in a multidimensional space. The performance of the system is tested on unseen network data containing various intrusion attacks [6].

In [7] is presented a NN-based intrusion detection method for the internet-based attacks on a computer network. The IDSs have been created to predict and thwart current and future attacks. The NNs are used to identify and predict unusual activities in the system. In particular, feed-forward NNs with the Backpropagation training algorithm were employed and the training and testing data were obtained from the Defense Advanced Research Projects Agency (DARPA)

intrusion detection evaluation data sets. The experimental results on real-data showed promising results on detection intrusion systems using NNs.

In [8], the authors deal with packet behavior as parameters in anomaly intrusion detection. The proposed IDS uses a Back-propagation Artificial Neural Network (ANN) to learn system's behavior. The authors used the KDD'99 data set for experiments and the obtained satisfying results.

In [9] is presented a deep learning approach for network intrusion detection system. The proposed system use self-taught learning, a deep learning technique based on sparse auto-encoder and soft-max regression, to develop an NIDS. The experimental results on the test data showed promising results on detection intrusion systems using NIDS.

In this paper, we present the application of Autoregression Integrated Moving Average (ARIMA) model for user behavior in Tor networks. We used the ARIMA and constructed a Tor server and a Deep Web browser (client) in our laboratory. Then, the client sends the data browsing to the Tor server using the Tor network. We used Wireshark Network Analyzer to get the data and then use the ARIMA to make the prediction. For evaluation we considered Number of Packets (NoP) metric. We present some simulation results considering Tor client.

The structure of the paper is as follows. In Sect. 2, we present a short description of Deep Web and Tor. In Sect. 3, we give an overview of ARIMA. In Sect. 4, we present an overview of R. In Sect. 5, we present the proposed model. In Sect. 6, we discuss the simulation results. Finally, conclusions and future work are given in Sect. 7.

2 Deep Web and Tor Overview

2.1 Deep Web

The Deep Web (also called the Deepnet, Invisible Web or Hidden Web) is the portion of World Wide Web content that is not indexed by standard search engines [10,11]. Most of the Web's information is far from the search sites and standard search engines do not find it. Traditional search engines cannot see or retrieve content in the Deep Web. The portion of the Web that is indexed by standard search engines is known as the Surface Web. Now, the Deep Web is several orders of magnitude larger than the Surface Web. The most famous of the deep web browsers is called Tor.

The Deep Web is both surprising and sinister and accounts for in excess of 90% of the overall Internet [12]. The Google and other search engines deal only with the indexed surface web. The deep-dark web hosts illegal markets, such as the Silk Road, malware emporiums, illegal pornography, and covert meeting places and messaging services. The pervasiveness of the Internet provides easy access to darkweb sites from anywhere in the world. The growth of the dark web has been paralleled by an increasing number of anonymity web-overlay services, such as Tor, which allow criminals, terrorists, hackers, paedophiles and the like to shop and communicate with impunity. Law enforcement and security agencies have had only very limited success in combating and containing this dark menace.

2.2 Tor

Tor is a low-latency, circuit-based and privacy-preserving anonymizing platform and network. It is one of several systems that have been developed to provide Internet users with a high level of privacy and anonymity in order to cope with the censorship measures taken by authorities and to protect against the constantly increasing threats to these two key security properties [13–16].

The Tor main design goals are to prevent attackers from linking communication partners, or from linking multiple communications to or from a single user. Tor relies on a distributed overlay network and onion routing to anonymize TCP-based applications like web browsing, secure shell, or peer-to-peer communications.

The Tor network is composed of the Tor client, an entry/guard node, several relays and the exit node. The Tor client is a software, installed on each Tor user's device. It enables user to create a Tor anonymizing circuit and to handle all the cryptographic keys, needed to communicate with all nodes within the circuit. The Entry Node is the first node in the circuit that receives the client request and forwards it to the second relay in the network. The Exit Node is the last Tor-relay in the circuit. Once the connection request leaves the entry node, it will be forwarded, through relays in the circuit, all the way to the exit node. The latter receives the request and relays it to the final destination.

When a client wants to communicate with a server via Tor, he selects n nodes of the Tor system (where n is typically 3) and builds a circuit using those selected nodes. Messages are then encrypted n times using the following onion encryption scheme. The messages are first encrypted with the key shared with the last node (called the exit node of the circuit) and subsequently with the shared keys of the intermediate node. As a result of this onion routing, each intermediate node only knows its predecessor and successor, but no other nodes of the circuit. In addition, the onion encryption ensures that only the last node is able to recover the original message.

A Tor client typically uses multiple simultaneous circuits. As a result, all streams of a user are multiplexed over these circuits. For example, a BitTorrent user can use one of the circuits for his connections to the tracker and other circuits for his connections to the peers.

3 ARIMA

The basic models of time series proposed by Box and Jenkins include Autoregression Model, Moving Average Model, Auto-regression Moving Average Model and Autoregression Integrated Moving Average Model. The Autoregressive Moving Average Model (ARMA) is a relatively mature model which contains Autoregressive (AR) Model, Moving Average (MA) Model and ARMA Model. ARMA (p, q) Model is expressed as follows:

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \epsilon_t - \theta_1 \epsilon_{t-1} - \theta_2 \epsilon_{t-2} \cdots - \theta_\eta \epsilon_{t-\eta} \quad (1)$$

where y_t is forecasting value of time, $t, \phi_i, \theta_j = (i = 1, 2, \cdots, p; j = 1, 2, \cdots, q)$ are model parameters, ϵ_t is the random process of white noise whose mean value is zero, p and q are orders of the model [17]. The ARIMA model is the extension of ARMA model. The ARMA model is usually used to dispose stable time series. If the series are non-stationary, it can be transfonned into a stationary time series using the d-th difference process, d is usually zero, one or two. Then, the series after difference is modeled by ARMA. The whole above process is called ARIMA. The prediction process of ARIMA model is as follows.

- Stationary Identification of Sequence: Check series' stationary with test methods of ADF root of unity.
- Series' Stationary Processing: If data series are unstable, we need to conduct difference processing until the data after disposed meet stationary condition. The order of difference is d when time series are stable.
- The Estimation of Parameters: We need check whether it has statistical significance.
- Conduct Hypothesis Testing: We need to judge whether residual sequence of the model is white noise.
- Conduct Forecasting Analysis: The forecasting analysis are conducted by using models that has been checked to be qualified.

4 The R Environment

The R is an integrated suite of software facilities for data manipulation, calculation and graphical display [18]. Among other things it has the following features.

- An effective data handling and storage facility.
- A suite of operators for calculations on arrays, in particular matrices.
- A large, coherent, integrated collection of intermediate tools for data analysis.
- Graphical facilities for data analysis and display either directly at the computer or on hardcopy.
- A well developed, simple and effective programming language (called "S") which includes conditionals, loops, user defined recursive functions and input and output facilities. Indeed most of the system supplied functions are themselves written in the S language.

The term "environment" is intended to characterize it as a fully planned and coherent system, rather than an incremental accretion of very specific and inflexible tools, as is frequently the case with other data analysis software. R is very much a vehicle for newly developing methods of interactive data analysis. It has developed rapidly, and has been extended by a large collection of packages. However, most programs written in R are essentially ephemeral, written for a single piece of data analysis.

Many people use R as a statistics system. The R is an environment within which many classical and modern statistical techniques have been implemented. A few of these are built into the base R environment, but many are supplied as packages.

5 Proposed Intrusion Detection Model for Tor Networks

The proposed system model is shown in Fig. 1. We call this system: User Behavior Prediction System using ARIMA (UBPS-ARIMA). We used UBPS-ARIMA and constructed a Tor server and a Deep Web browser (Tor client) in our laboratory. Then, the client sends the data browsing to the Tor server using the Tor network. We used Wireshark Network Analyzer [19] to get the data. The data are stored in the log files. The system runs until the number of loops is achieved.

The data in the log files are considered as old data and the current data are the input of ARIMA model. The UBPS-ARIMA can predict the user behavior using these data.

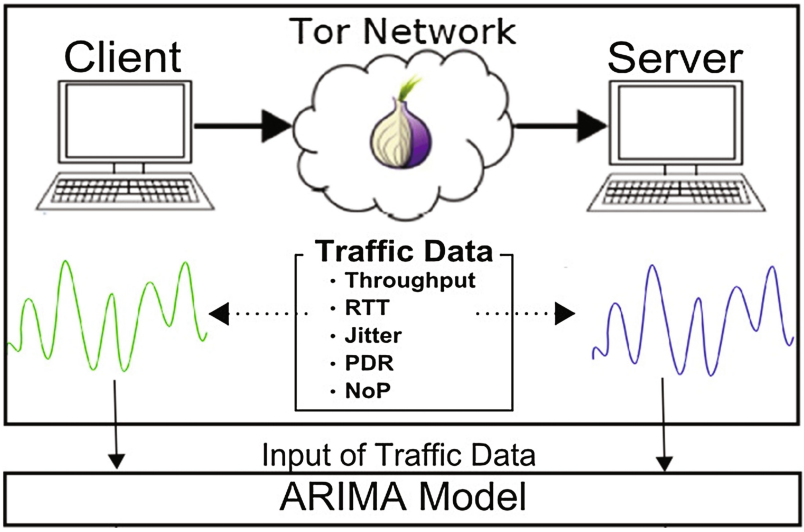


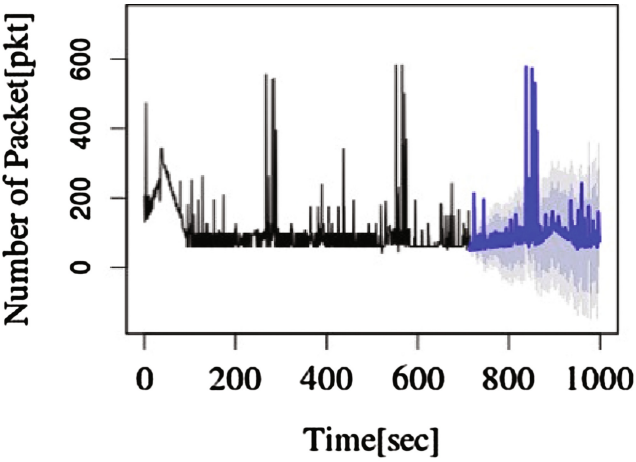
Fig. 1. Proposed system model.

6 Simulation Results

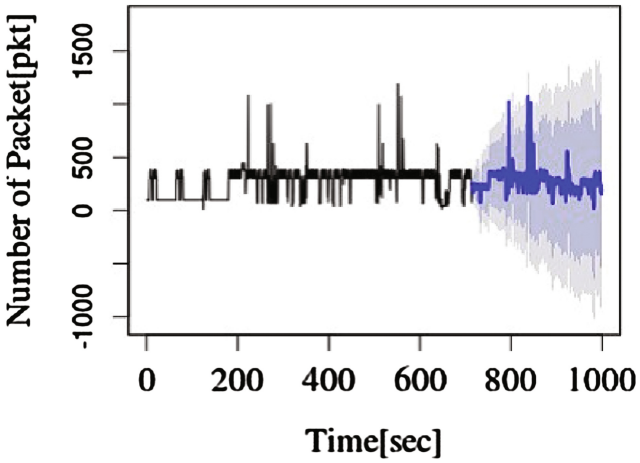
We carried out some simulations using the UBPS-ARIMA. In Fig. 2, we show the Number of Packet (NoP) parameter for Tor client and Tor server. The simulation parameters are shown in Table 1. We ran the simulation 10000 times. In Fig. 2(a) are shown the data for Tor client and in Fig. 2(b) for Tor server. As evaluation parameter, we used the NoP. From 0 [sec] to 700 [sec] are shown the training data. Then, from 701 [sec] to 1000 [sec] are shown the predicted data. The simulation results show that UBPS-ARIMA model has a good prediction.

Table 1. Simulation parameters.

Parameters	Values
Number of training data	1000
Number of measurements	400
Predictive model	ARIMA model



(a) Tor client



(b) Tor server

Fig. 2. Simulation results.

7 Conclusions

The Tor network has become a useful tool for malicious users. With Tor, the users are able to compromise the non-repudiation principle of computer security. Also, the potentially hackers may launch attacks such as DDoS or identity theft behind Tor. For this reason, there are needed new systems and models to detect the intrusion in Tor networks.

In this paper, we presented the application of ARIMA for prediction of user behavior in Tor networks. We used ARIMA model and constructed a Tor server and a Deep Web browser. Then, the client sent the data browsing to the Tor server using the Tor network. We used Wireshark Network Analyzer to get the data and then used the ARIMA to make prediction of user behavior. The simulation results show that UBPS-ARIMA has a good prediction of user behavior.

References

1. Tor Project Web Site. http://www.torproject.org/
2. Dingledine, R., Mathewson, N., Syverson, P.: Deploying low-latency anonymity: design challenges and social factors. IEEE Secur. Priv. **5**(5), 83–87 (2007)
3. Dingledine, R., Mathewson, N., Syverson, P.: Tor: the second-generation Onion Router. In: Proceedings of the 13th Conference on USENIX Security Symposium (SSYM-2004), vol. 13, p. 21 (2004)
4. Ling, Z., Luo, J., Wu, K., Yu, W., Fu, X.: TorWard: discovery of malicious traffic over Tor. In: Proceedings of IEEE INFOCOM 2014, pp. 1402–1410, April 2014
5. Reddy, E.K.: Neural networks for intrusion detection and its applications. In: Proceedings of the World Congress on Engineering 2013 Vol. II, WCE-2013, July 2013
6. Linda, O., Vollmer, T., Manic, M.: Neural network based intrusion detection system for critical infrastructures. In: Proceedings of International Joint Conference on Neural Networks (IJCNN-2009), pp. 1827–1834, June 2009
7. Shum, J., Malki, H.A.: Network intrusion detection system using neural networks. In: Proceedings of Fourth International Conference on Natural Computation (ICNC-2008), pp. 242–246, October 2008
8. Al-Janabi, S.T.F., Saeed, H.A.: A neural network based anomaly intrusion detection system. In: Developments in E-systems Engineering (DeSE), pp. 221–226, December 2011
9. Niyaz, Q., Sun, W., Javaid, A.Y., Alam, M.: A deep learning approach for network intrusion detection system. In: Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (Formerly BIONETICS), BICT-15, vol. 15, pp. 21–26 (2015)
10. Lang Hong, J.: Deep web data extraction. In: Proceedings of IEEE International Conference on Systems Man and Cybernetics (SMC-2010), pp. 3420–3427, October 2010
11. Singh, M.P.: Deep web structure. IEEE Internet Comput. **6**(5), 4–5 (2002)
12. Stupples, D.: Security challenge of Tor and the deep web. In: 8th International Conference for Internet Technology and Secured Transactions (ICITST-2013), December 2013
13. Biryukov, A.: Trawling for Tor hidden services: detection, measurement, deanonymization. In: Proceedings of IEEE Symposium on Security and Privacy (SP-2013), pp. 80–94, November 2013

14. Dhungel, P., Steiner, M., Rimac, I., Hilt, V., Ross, K.W.: Waiting for anonymity: understanding delays in the Tor overlay. In: Proceedings of IEEE Tenth International Conference on Peer-to-Peer Computing (P2P-2010), pp. 1–4, August 2010
15. Xin, L., Neng, W.: Design improvement for Tor against low-cost traffic attack and low-resource routing attack. In: Proceedings of WRI International Conference on Communications and Mobile Computing (CMC-2009), pp. 549–554, January 2009
16. Syverson, P.: A peel of onion. In: Proceedings of ACSAC-2011, pp. 123–135, December 2011
17. Min, Y., Bin, W., Liang-Ii, Z., Xi, C.: Wind speed forecasting based on EEMD and ARIMA. In: Chinese Automation Congress (CAC-2015), pp. 1299–1302 (2015)
18. The R Project for Statistical Computing. http://www.r-project.org/
19. WireShark Web Site. http://www.wireshark.org/