2018 15th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology (ISCISC)
August 28-29, 2018; Shahid Rajaee Teacher Training University - Tehran, Iran

An efficient user identification approach based on Netflow analysis

Atieh Bakhshandeh
Research Center for Developing Advanced Technologies
Tehran, Iran
bakhshandeh@rcdat.com

Zahra Eskandari
Research Center for Developing Advanced Technologies
Tehran, Iran
eskandari@rcdat.com

***Abstract*— with the advent of new technologies such as cloud- based services, smart phones, tablets and etc. users' connectivity to networks are inevitable. This will result in the generation of huge amount of traffic from the users' activities. For forensic examiners, this traffic is a critical source of information. In network forensics, focusing only on the IP addresses will result to evidence which is not confident as the account might have been compromised. Thus, the associated user is of more interest for forensic scientists rather than the IP address. Moreover, with the wide range of devices that a user may use (smart phone, tablet, laptop, etc.) and also the wide use of DHCP, the IP address is not a suitable identifier to distinguish users. This paper, proposes a method for efficiently identifying users of a network based on their behavior using the netflow traffic (which does not contain payloads). We extract a feature set from the flows of the network and use a random forest model to classify users. We have achieved the precision of 0.94 in the detection of users. The results show that this method can be effectively used by forensic scientists as they do not need to examine the whole traffic and only the reduced netflow traffic would be enough for investigation.**

Keywords— user profiling, forensics, netflow

I. INTRODUCTION

Profiling is defined as a process for identifying different users of a network based on personal traits and behavioral characteristics. User Profiling is used for many applications such as forensic applications, tracing user's activities on the Internet, controlling the use of shared resources of a network for security purposes and detecting intruders in a network based on their behavioral characteristics and preferences, i.e. if a user is logged in with another user's credentials, the system will be able to detect this incident. The practice of user profiling is based on some assumptions such as: 1) human behavior is predictable, 2) Intruders often do their malicious activities in a certain manner and can be distinguished from benign users, 3) The way users do their activities in the network relates to their personal characteristics.

There are many different methods for user profiling using various data sources like logs of the web traffic or packet traffic of a network. However, such methods are not practical due to their need for storing large amount of traffic and using heavy resources. Alternatively, there are some methods that use flows of a network instead of the whole traffic to identify different users. Flow is defined in RFC3917 as "a set of IP packets passing an observation point in the network during a certain time interval" [1]. These packets have common properties including source IP, destination IP, source port, destination port and protocol.

In order to fingerprint users using network flows, the general trend is to extract some features for each user and then train a classifier with a labeled data set using the extracted features. Classification is the process of categorizing data according to a predefined set of classes and assign each item the label of the proper class. In Section II, first, some of the works in this area are reviewed and after that we review the scheme proposed in [7] which we improve in our paper. Next, in Section III, we propose our method for achieving a significant improvement over the results presented in [7].

II. RELATED WORK

In [2], the authors showed that by mining solely NetFlow data belonging to an Internet Service Provider, an attacker is able to track users, and accurately estimate when they are connected to the network and which IP address they are using even if the users are behind a NAT. Their solution uses a series of properly trained HMMs(Hidden Markov Models) and cleverly combines them to maximize the success rate. However, as the authors reported it seems that solutions such as TOR or VPN tunneling may degrade the performance of their approach.

In [3], a netflow based flow analysis and monitoring system is proposed in which pattern matching is used to classify traffic types into normal and anomalous. Their data collection module receives and analyzes NetFlow-exported packets and inserts per flow record information into the Oracle database. A real-time anomalous traffic monitoring module with a stable matching pattern algorithm and two traffic statistic based intrusion detection algorithms (one algorithm is based on variance similarity while the other is based on Euclidean distance) are embedded in the system to detect worm and other malicious attacks. Also a proved join strategy is designed along with the two traffic statistic based intrusion detection algorithms. Their system is online and works in real time and its applications is mainly in intrusion detection and prevention systems.

In [4], the authors presented a new method for network access control which is based on user profiling. The BB-NAC(Behavior-Based Network Access Control) is a network access control mechanism which is based on behavior profiles rather than rules. In [4], an enhanced BB-NAC mechanism is proposed that fully optimizes the creation of

978-1-5386-7582-3/18/$31.00

clusters of behavior and overcomes the lack of automatic clustering with the original BB-NAC. They also present an incremental learning algorithm that automatically updates the behavior-based access control policies. They also show that the enhanced mechanism can differentiate between normal changes in the behavior profiles and attacks.

There are different approaches for detection of users using network flows. Some methods focus on the behavior of users; As an example in [5], the authors show that the length of flow traces of each user in one day and two weeks are correlated. On the other hand, Some methods are service/application based [6].

In [7], the authors proposed a supervised learning model that uses flow features to identify users within a given set. The authors name flow features as flow-bundle-features that can be derived from packet-level and flow-level features. Their features include two types of features: user features and host features. User features are characteristics of behavior of an individual user while host features are properties of the user's host platform. We have employed a similar approach as the scheme proposed in [7] with some modifications to make its performance better. As a result we have achieved a significant improvement over their results. Therefore, we give a brief overview of their approach here in A.

A. Review of the proposed scheme in [7]

First we re-define the concepts of packet-level features, flow-level features and flow-bundle-level features. Packet-level features are those that can be directly obtained by packets. Examples of packet-level features are source/destination IP addresses, source/destination ports and protocols. Flow-level features are features that are computed using all packets in a flow. Flow-bundle-features are features which are computed based on the values of flow-level features. Examples of flow-bundle features are average inter-flow-gap, flow rate(number of flows per hour), average flow duration. The paper uses flow-bundle-level features representing a group of N flows. The flow-bundle-level features f_i are derived from flow-level features. The authors uses a sliding window for bundling the flows. Suppose that flow records $1, 2, 3, ..., n$ are represented on a line and let W be a sliding window which covers N flows so that $N < n$. The N flows falling inside a window will form one sample ($S_i = \{f_{i1}, f_{i2}, ..., f_{in}\}$). The window W starts from the left end of the line covering the N flows ($1, 2, ..., N$), then gradually moves forward subsequently covering $(2, 3, ..., N + 1),(3, 4, ..., N + 2)$ and so on. The set of flows falling inside window frame in subsequent positions generate the set of samples $S_1, S_2, ..., S_3$. In this paper, the authors also divide types of features from another perspective. From this point of view, they divide features into host-based features and user-based features. We only work on user-based features and try to improve the results only using user-based features. User-based features introduced by the authors are described in the following.

- Destination IP Dispersion: It is the number of unique destination IP addresses visited in a window frame (bundle).
- Site of Interest: It is the destination IP that was most contacted by a user in a bundle of N flows.
- Destination IP Cohesion: It is the number of times the site of interest is visited in a window frame.
- Application Diversity: It is the number of unique destination ports in a bundle.
- Favorite application: It is the top destination port in a bundle.
- Browsing Duration: It is the median of flow duration in bundle of N flows.

After computing the above features and some other host-based features on a set of flows collected from a fixed set of users, the authors give these features to a classifier and train the classifier with them. Then, an unlabeled set of flows, again collected from the same set of users, is given as input to the classifier and the classifier predicts the user identity label for each of the flows. The authors use four different type of classification algorithms: K-nearest neighbor(for k=3 and k=7), random forest and C4.5 decision tree. The comparisons show that random forest outperforms other classification algorithms with the accuracy of 83%.

In this paper, we made two modifications to [7]. Our first modification is adding some extra features to the above-mentioned user-based features. The second and most effective modification is that we change the way the flows are grouped together. i.e. our bundles are based on time intervals rather than window frame. Precise description of our approach is presented in section III.

III. THE PROPOSED SCHEME

In this paper, we propose an improvement over the method presented in [7]. In this section, first we discuss the details of our data set and the standard format that is used for collecting our data in A. Then we articulate the features we extract from our data set in B. Finally, we discuss the details of our classification phase in C.

A. Data Set

We collected the data of our institute network for ten days and used this data as the data set. While collecting the data, we knew which flows belong to each user so we used this information for labeling the data set. We have 46 distinct users in our data set. The format of the data is according to Netflow standard V5. A brief overview of Netflow standard is given in the following.

1) Netflow Standard

NetFlow is a network protocol developed by Cisco for collecting IP traffic information and monitoring network traffic. Routers that have the NetFlow feature enabled generate NetFlow records. These records are exported from the router and collected using a NetFlow collector. The NetFlow collector then processes the data to perform the traffic analysis and presentation in a user- friendly format. NetFlow v1 was originally introduced in 1990 and has since evolved to NetFlow version 9. Today, the most common versions are v5 and v9. All the packets aggregated in a netflow record have five common properties: source address, destination address, source port, destination port and protocol. We use Netflow v5 in our work. The fields of Netflow which we require for our feature extraction are the following:

- Source address: IPv4 source address

- Destination address: IPv4 destination address
- Source port: TCP/UDP source port number
- Destination port: TCP/UDP destination port number
- Protocol: IP protocol byte
- Start Time: The start time of the flow
- End Time: The end time of the flow
- Input packets: The number of packets associated with an IP Flow that enter an interface
- Input bytes: The number of bytes associated with an IP Flow that enter an interface
- Output packets: The number of packets associated with an IP Flow that leave an interface
- Output bytes: The number of bytes associated with an IP Flow that leave an interface
- Time Duration: The difference of End time and start time of an IP flow

B. *Feature Extraction*

In our feature extraction phase, we first group the netflow records of a user into five-minute intervals. In other words, we use intervals of five minute length instead of a window frame of length N . Thus for each user, we group its flows as five minute intervals. Then, we compute features which are described in II.A over each five-minute interval for each user. Moreover, we add some extra features to that feature set. The final user-based feature set we compute are according to the following list.

- Browsing outgoing bytes: The sum of input bytes in a five minute interval.
- Browsing incoming bytes: The sum of output bytes in a five minute interval.
- Browsing outgoing packets: The sum of input packets in a five minute interval.
- Browsing incoming packets: The sum of output packets in a five minute interval.
- Destination IP Dispersion: The number of distinct destination IP addresses which the user connects to them in a five minute interval.
- Application Diversity: The number of distinct destination ports that the user connects to them in a five minute interval.
- Source port median: The median of source port in a five minute interval. The source port range in a client's connections is closely related to the operating system of the host for that client. Hence, median source port can identify operating system and kernel type of a user.
- Destination port median: The median of destination port in a five minute interval. The destination port range of a user will give information about the services the user use. Hence, destination port median can be a distinguishing feature regarding the services a user use.
- Favorite Application: The destination port that the user is connected to it, the maximum number of times in a five minute interval.
- Favorite source port: The source port that the user is connected to it the maximum number of times in a five minute interval.
- Browsing duration: The average of time duration of flows in a five minute interval. Note that in [7], this feature was the median of time duration and we change it to average to get more tangible results.
- Flow Count: The number of flows which belongs to a five minute interval.

C. *Classification*

After calculating the features described in B, we train a classifier using these features. At first, there were 46 distinct users in our data set. We have eliminated the users who does not have enough records (users with less than 150 records). In addition, we eliminate users with less than 20 records per day. As a result, 26 users were removed from our data set, thus we have 20 distinct users which indicates 20 distinct classes. The whole data set is collected within a 10 day interval. We divide our data set into two parts: training data and test data. We take 5 days for training and the remaining 5 days for test. Moreover, we only select those users which were in both the training data and test data. We choose a random forest model as our classifier. We train the random forest with the training data and using the features described in B and after that, we have labeled the test data using the prediction module. Our results are presented in section IV.

IV. EXPERIMENTAL RESULTS

In this section, the experimental results are provided and the model performance is evaluated using common metrics such as accuracy, precision, recall and F-measure. In order to define these metrics precisely, first we need to define some concepts. In classification, we have condition positive which indicates the number of real positive cases in the data. Similarly, there is a condition negative which is the number of real negative cases in the data. Based on these conditions, there will be four parameters after classification: true positive(TP) which is the number of positive examples labeled as such, true negative(TN) which is the number of negative examples labeled as such, false positive(FP) that is the number of negative examples labeled as positive and false negative(FN) that is the number of positive examples labeled as negative. We define accuracy, precision, recall and F-measure according to the following formulas.

- Accuracy: It is intuitively the proportion of the correct results of a classifier and is defined according to the following equation:

$$accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (1)$$

- Precision: Precision answers the question that out of all the examples the classifier labeled as positive, what fraction were correct? It is defined as the following equation:

$$precision = \frac{TP}{TP+FP} \quad (2)$$

- Recall: Recall answers the question that out of all real positive examples, what fraction did the

classifier labeled as positive? It is defined as the following equation:

$$recall = \frac{TP}{TP+FN} \quad (3)$$

- F-measure: It is a measure that combines precision and recall and is defined according to the following equation:

$$F = 2 \times \frac{precision \times recall}{precision + recall} \quad (4)$$

We calculate the above metrics for our method. We have also implemented the method proposed in [7] on our own data set and calculate the above mentioned metrics for their algorithm on our data. The results are shown in TABLE1.

It can be seen that there is significant improvement over the method used in [7]. There are two analysis behind this improvement. First, note that we add an extra feature which we named it flow count. This feature is the number of flows which fall into a five-minute interval. Fig. 1, 2 and 3 show the flow count as a function of 5-minute intervals in six different days for three different users. As it is illustrated, there exists almost a fixed pattern for each user. Thus, this feature is an efficient distinguishing feature**.**

Second, if we consider the behavior of users in 5 minute intervals which does not overlap each other, it will give a better sense of their behavior rather than considering the bundles of length 100 which overlap each other. Because when the bundles of length 100 overlap each other, it will cause them to be very similar to each other and it would not reflect a long term behavior of a user. Furthermore, these bundles do not reflect the amount of activity during time; as an example one user can generate one bundle in 5 minutes and another user may generate a similar bundle in 5 hours. Obviously their behaviors are not similar to each other while the bundle algorithm will identify them as one user because their bundles are similar to each other.

TABLE I. COMPARISON OF THERESULTS

	accuracy	precision	recall	F-measure
Our method	0.946084	0.944384	0.948497	0.94433
Method in [7]	0.5209767	0.4776182	0.47022881	0.47389470

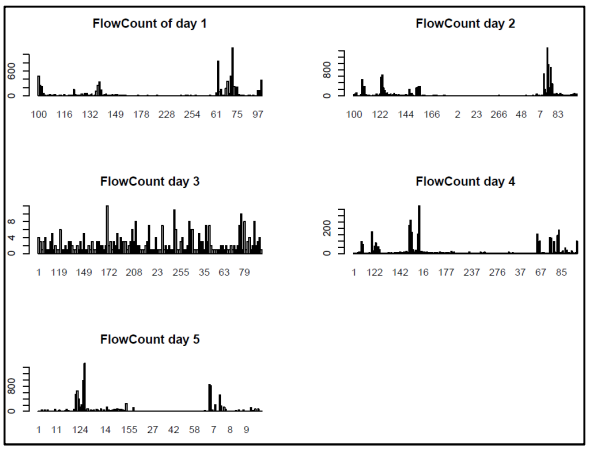

Fig. 1 Flow count of user A in 5 different days

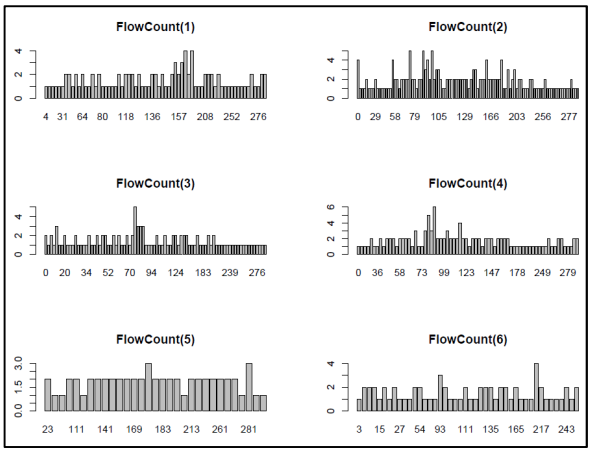

Fig. 2 Flow count of user B in 6 different days

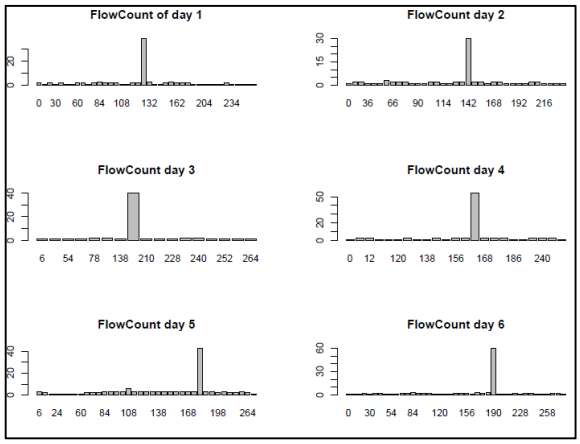

Fig. 3 Flow count of user C in 6 different days

V. CONCLUSION

In this paper, we proposed a new algorithm for detecting users of a network based on their personal characteristics and behavior. We use the netflow records of a network as input to our algorithm. We use the feature set proposed in [7] as well as some extra new features proposed by us. We create intervals of length five minute for each user's activity in the network and calculate the features within these intervals. The results show that this method can be effectively used by forensic scientists in collecting evidence. One challenge in this area is that a user behavior may vary during time and thus the model we have had created at first cannot detect these kind of users as time pass. A recommendation for future work can be a solution for the challenge of creating models that can model the long term behavior of users. Other suggestions for future work can be trying different values(other than 5 minutes) as time interval and selecting the best value. It is also a good idea to use feature selection approaches for choosing the best subset of features that is possible.

REFERENCES

[1] J. Quittek, JT. Zseby, B. Claise and S. Zander, "RFC 3917 - IPFIX Requirements". IETF. Retrieved 2010-02-11.

[2] N. V. Verde, G. Ateniese, E. Gabriell, L. V. Mancini and A. Spognardi, No NATd User left Behind: Fingerprinting Users behind NAT from NetFlow Records alone, Distributed Computing Systems (ICDCS), IEEE, 2014.

[3] L. Bin, L. Chuang, Q. Jian, H. Jianping and P. Ungsunan, A NetFlow based flow analysis and monitoring system in enterprise networks, Computer Networks 52 (2008) 10741092.

[4] V. Frias-Martinez, J. Sherrick, S. J. Stolfo and A. D. Keromytis, A Network Access Control Mechanism Based on Behavior Profiles, Computer Security Applications Conference, IEEE, 2009.

[5] Melnikov N, Scho¨ nwa¨lder J., Cybermetrics: user identification through network flow analysis, Lecture notes in computer science, vol.6155, Springer,16770, 2010.

[6] Perelman V, Melnikov N, Scho¨ nwa¨lder J. Flow signatures of popular applications,IEEE; 2011. p. 916.

[7] Vinupaul M.V, R. Bhattacharjee, Rajesh R. and G. S. Kumar, User Characterization through Network Flow Analysis, Data Science and Engineering (ICDSE), IEEE, 2016.

[8] Y. Song, S. J. Stolfo and T. Jebara, Markov Models for Network-Behavior Modeling and Anonymization, In Technical reportsColumbia University, http://hdl.handle.net/10022/AC:P:10682, 2011.