Network Trace Anonymization Using a Prefix-Preserving Condensation-Based Technique (Short paper)

Ahmed Aleroud[1(✉)], Zhiyuan Chen[2], and George Karabatis[2]

[1] Department of Computer Information Systems, Yarmouk University, Irbid 21163, Jordan
ahmed.aleroud@yu.edu.jo
[2] Department of Information Systems, University of Maryland,
Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250, USA
{zhchen,georgek}@umbc.edu

Abstract. This paper proposes a method to anonymize network trace data by utilizing a novel perturbation technique that has strong privacy guarantee and at the same time preserves data utility. The resulting dataset can be used for security analysis, retaining the utility of the original dataset, without revealing sensitive information. Our method utilizes a condensation based approach with strong privacy guarantees, suited for cloud environments. Experiments show that the method performs better than existing anonymization techniques in terms of privacy-utility trade off, and it surpasses existing techniques in attack prediction accuracy.

Keywords: Privacy preserving · Data mining · Intrusion detection · Anonymization · Network traces

1 Introduction and Background

Sharing network trace data to create models that identify security attacks is a very sensitive issue for any organization as everyone prefers to access real (not synthetic) network trace datasets for security analysis, but nobody wants to reveal internal information to the public. Organizations use public or private clouds to store data (resources) which may contain sensitive information that may be detrimental if an adversary has access to it. Real-world traces usually contain sensitive information, e.g. host addresses, emails, personal web-pages, and even authentication keys [1]. These traces must be first "anonymized" to eliminate any private information before they can be shared among researchers. Additionally, anonymization should maintain both utility and the privacy of the trace. Network traces consist of a packet header and payload. The header contains information about source and destination Internet Protocol (IP) addresses, port numbers, protocol, type of packet, and other fields which must be anonymized before sharing. Network data sources such as RIPE (European IP Network Coordination Centre) [2], Route Views [3], and the Center for Applied Internet Data Analysis (CAIDA) [4] provides anonymized collected traces to the research community. A simple approach to

C. Debruyne et al. (Eds.): OTM 2016 Conferences, LNCS 10033, pp. 934–942, 2016.
DOI: 10.1007/978-3-319-48472-3_59

anonymize IP addresses is to map each IP into a random 32-bit address [5], yet it results in losing the prefix relationship between IP addresses, which may be important for clustering based on such relationships. Therefore, it is highly desirable to preserve the prefix when anonymizing IPs. In addition, the existing techniques do not handle data injection attacks. This is the act of injecting information to be logged, so that it may be recognized after the anonymization process [6]. Data injection attacks are executed by sending IP packets with random source and destination IP addresses, spoofed IP addresses, or forged packet headers that appear in the anonymized traces [7]. The adversary may then uncover the mapping between the injected plain text and the anonymized address (see [8, 9]). Our technique anonymizes network trace data, and generates a dataset that can be deployed in clouds, and freely distributed to organizations for testing intrusion detection techniques without revealing sensitive information. The proposed algorithm has strong privacy protection and high utility. Related work includes several methods of data obfuscation to achieve network traces anonymization. Black marker is the most trivial method that replaces all IP addresses with a constant [10]. Enumeration is another method that can be applied to well-ordered sets. It replaces a value with a random one for the first field and continues with a higher value for each following field [10]. Permutation approaches are used for anonymizing the IP and MAC addresses. A random permutation is applied to map non-anonymized addresses to a set of possible addresses. Shuffling re-arranges pieces of data within a field (an example tool is PktAnon [11]). Truncation techniques can anonymize the IP and MAC addresses by deleting a portion of the data. Truncation replaces the n least significant bits of a field with zeros. This technique effectively makes an end-point non identifiable [7]. Other approaches such as Random Substitution randomize data to provide no link between observations [7], and random time shifting [12]. Generalization approaches replace data with more general data [10], e.g. by grouping of TCP/UDP port numbers using a fixed value for both categories [10]. Finally, Prefix-preserving pseudonymization techniques are similar to the permutation ones, however, they preserve the prefixes and cryptographic keys are used to keep the mapping consistent [13–15]. In the rest of the paper Sect. 2 describes our anonymization approach. In Sect. 3 experiments and analysis are discussed, and conclusions are in Sect. 4.

2 Research Methodology

We address two shortcomings in existing work on anonymizing network trace data [10, 16–19]. First, the lack of a formal and strong privacy preservation model and secondly, the exposure to attacks such as injection attacks [6, 20]. We utilize an improved version of K-anonymity [21] and we identify the features (attributes) in network traces that need to be anonymized and also are important for intrusion detection. Source and destination IP addresses reveal information that may lead to the discovery of a user identity. The destination IP addresses can be used by attackers to launch attacks. On the other hand, IP addresses can be used in intrusion detection algorithms as well. For example, if we know that attacks often originate from specific IP addresses then we can

identify future attacks from the same addresses. Thus we need to carefully balance the need for privacy protection and intrusion detection when anonymizing IP addresses.

2.1 Anonymization of IP Addresses

The following two stages describe anonymization of IP addresses:

- Encrypt/permute the leading digits of the IP addresses (network number). Intrusion detection methods can still use the leading portion of the IP addresses. Attackers may discover the subnet of a host but the next stage prevents identifying the host.
- For the remaining digits of the IP (host number part), cluster these addresses, and randomize addresses in the same cluster (exact IP address cannot be located).

Figure 1 summarizes the steps needed to anonymize the IP address. The dataset D is divided into n datasets, such that D_i contains flows with label L_i where each label can be an attack or a benign activity. Then permute the leading digits of the IP addresses (network number) using prefix preserving permutation function. The IP addresses are then clustered into k clusters based on their least significant digits (host number). The mean for the least significant digits of IPs in the same cluster (host number) is calculated. Then the least significant digits of IPs (the last three digits) in each cluster are replaced using the computed statistics. Figure 2 demonstrates IP anonymization.

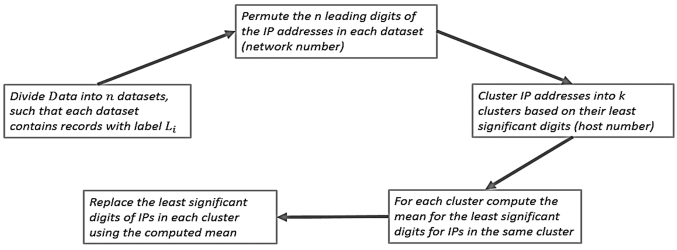

Fig. 1. Steps to anonymize IP addresses

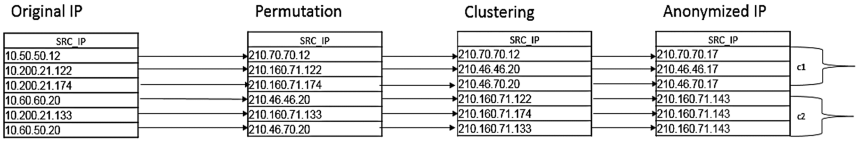

Fig. 2. IP anonymization example

2.2 Anonymizing Non-IP Features

We utilize a condensation-based approach to perform anonymization on non-IP features. The original condensation method utilizes the distribution of the original data to generate a synthetic dataset [22]. The condensation method has some similarity with K-anonymity. It creates groups where records are k- indistinguishable. However, instead of masking values, random synthetic data is generated [22]. We apply two modifications to the original condensation algorithm.

- First, we implement a per class condensation mechanism on network traces. The typical condensation algorithm does not consider the differences between classes to perform the de-identification. In general, there is a significant difference between the behavior of network attackers and other users and such differences need to be captured in the anonymized data
- Second, the typical condensation algorithm picks cluster centers randomly, which may lead to inferior clusters. Instead, we utilize k-means clustering algorithm which is relatively efficient in terms of within-class variance [23].

The first two steps in anonymizing non-IP features are similar to those for IP addresses. Figure 3 shows two more steps to anonymize such feature, first the clusters are sorted in ascending order of cluster size. For each cluster C_j that contains less than k records, k- $|C_j|$ records are selected if they are the closest to the center of C_i that lies in a cluster in which the number of records is greater than k.

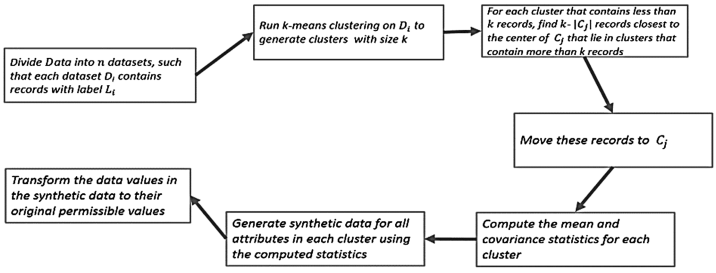

Fig. 3. Anonymization of non-IP features

Algorithm1: Generating Synthetic data (C_j)

1. Use Principal Component Analysis to shift the data in the cluster in a new space $(C_j \rightarrow Z)$, by creating independent components $Z_1, Z_2, \ldots Z_p$
2. For each independent component Z_i
   Generate a random data Z_i' with normal distribution such that $|Z_i'|=|Z_i'|$; $\mu_{Z_i'}=\mu_{Z_i}$; and $\sigma_{Z_i'}=\sigma_{Z_i}$;
3. Combine $Z_1', Z_2', \ldots Z_p'$ into one dataset Z_1' in an orderly manner
4. Using Reverse PCA shift Z' to the original space
   $Z' \rightarrow C_j'$
   Return C_j'

The selected records are then moved to C_j. In addition, synthetic data is generated using Algorithm 1. For each cluster C_j the data is shifted into a new space a using Principal Component Analysis. In the new space Z_1, Z_2, Z_p are independent components. Then, a random data $Z_i^{'}$ with similar statistical features of Z_i is generated. Finally, $Z_1^{'}, Z_2^{'}, Z_3^{'}$ are combined into one dataset.

3 Experiments and Evaluation

While it is straightforward to simulate the generation of benign and suspicious traces using tools such as softflowd [24] by listening to network interface, this approach does not capture all intended characteristics of benign activities. PREDICT (A Protected REpository for Defense of Infrastructure against Cyber Threats) has shared real-world datasets for cyber security research. We use packet captures from the 2013 National Collegiate Cyber Defense Competition (nccdc.org) in our experiments. We created a tool to generate benign and suspicious flows from packets. The flow information for training and evaluation is shown in Table 1.

Table 1. The characteristics of the dataset used in our experiments

Features	Types of activities
• I_{src}, Source IP	• Benign
• I_{dst}, Destination IP	• Potentially Bad Traffic
• P_{src}, Source port	• Attempted User Privilege Gain
• P_{dst}, Destination port	• Generic Protocol Command Decode
• Prot, Protocol	• Attempted Information Leak
• Pckts, Number of packets in the flow	• Web Application Attack
• Octs, Number of octets in the flow	• Detection of a Network Scan
• T_{start}, Start time of the flow	• Access to Vulnerable Web Application
• T_{end}, End time of the flow	*Number of selected flows 400893 70 % is used for training, 30 % for testing*
• D, Flow duration	

3.1 Evaluation Measures

Privacy: Conditional privacy is used to measure the privacy of anonymized traces [25]. It is an average measure of privacy that was originally proposed in the context of distribution reconstruction after additive perturbation. The measure is based on differential entropy of the random variable. The differential entropy of A given B = b is

$$h(A|B) = -\int_{\Omega_{A,B}} f_{A,B}(a,b) log_2 f_{A|B=b}(a) da\, db \tag{1}$$

Where A is a random variable that describes the data, and B is the variable that gives information on A.$\Omega_{A,B}$ identifies the domain of A and B. Therefore, the average conditional privacy of A given B, is:

$$\prod(A|B) = 2^{h(A|B)} \tag{2}$$

Accuracy: Several accuracy measures are used to validate the effectiveness of our anonymization algorithms such as TP Rate, FP Rate, Precision, Recall, F-Measure, MCC (Matthews correlation coefficient), and ROC (Receiver Operating Characteristic) Area. We prove that our model works reliably and is able to produce satisfactory classification accuracy values while preserving privacy.

3.2 Results

Figure 4 shows the conditional privacy measures for the de-identified dataset using different techniques when changing the values of K.

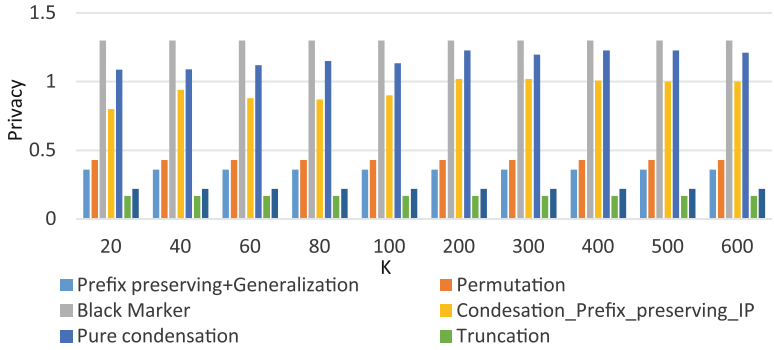

Fig. 4. Conditional privacy using different anonymization methods/PREDICT dataset

Our algorithm performed better than most existing techniques. While the Black Marker technique performs better in terms of privacy, we show that it has low accuracy values. We utilized two types of condensation approaches. First, we performed the main condensation without preserving the Prefix of IP. Secondly, we performed condensation with prefix preserving anonymization on IP addresses. It is obvious that the main condensation attains higher privacy values than prefix-preserving condensation. To evaluate if our approach can differentiate between benign activities and attacks on anonymized data, we ran an experiment to compare accuracy before and after anonymization using K-Nearest Neighbor classifier (Table 2). We also compared our approach with existing anonymization techniques such as Black Marker, Permutation and Truncation. Our approach to anonymize IP addresses using prefix preserving technique, in addition to the conventional condensation for non-IP features leads to no significant information loss. Compared to other techniques, it achieves higher attack detection rate and lower false positives with both classifiers.

Table 2. PREDICT Data-KNN classification on anonymized data

	TP Rate	FP Rate	P	R	F-Measure	ROC Area	PRC Area	Class
Original	0.78	0.03	0.96	0.78	0.86	0.87	0.87	Attack
	0.96	0.21	0.77	0.96	0.86	0.87	0.76	Normal
	0.86	**0.11**	**0.88**	**0.86**	**0.86**	**0.87**	**0.82**	Avg
Condensation-Per Class	0.66	0.00	1.00	0.66	0.79	0.83	0.81	Attack
	1.00	0.33	0.78	1.00	0.87	0.83	0.78	Normal
	0.84	**0.18**	**0.88**	**0.84**	**0.84**	**0.83**	**0.79**	Avg
Condensation All Class	0.47	0.10	0.46	0.47	0.47	0.68	0.30	Attack
	0.89	0.30	0.89	0.89	0.89	0.68	0.89	Normal
	0.82	**0.20**	**0.82**	**0.82**	**0.82**	**0.68**	**0.79**	Avg
Pure Condensation	0.69	0.31	0.70	0.69	0.69	0.68	0.64	Attack
	0.68	0.30	0.67	0.68	0.67	0.68	0.60	Normal
	0.68	**0.31**	**0.68**	**0.68**	**0.68**	**0.68**	**0.63**	Avg
Prefix-preserving (IP) & Generalization (other features)	0.51	0.03	0.94	0.51	0.66	0.73	0.75	Attack
	0.96	0.48	0.60	0.96	0.74	0.73	0.60	Normal
	0.71	**0.23**	**0.79**	**0.71**	**0.70**	**0.73**	**0.68**	Avg
Permutation	0.38	0.05	0.95	0.38	0.54	0.66	0.83	Attack
	0.94	0.61	0.33	0.94	0.49	0.66	0.32	Normal
	0.52	**0.19**	**0.80**	**0.52**	**0.53**	**0.66**	**0.70**	Avg
Truncation	0.78	0.39	0.98	0.78	0.87	0.69	0.98	Attack
	0.60	0.21	0.05	0.60	0.10	0.69	0.04	Normal
	0.78	**0.39**	**0.96**	**0.78**	**0.85**	**0.69**	**0.96**	Avg

4 Conclusions

Our method anonymizes network traces using prefix preserving and condensation. It utilizes a novel perturbation technique with a very strong privacy guarantee and preserves data utility. In addition, it clusters flows based on their features. Each cluster contains flows with similar features. Our experiments show that our method performs better than existing ones in terms of privacy-utility tradeoff. We plan to investigate other privacy techniques such as differential privacy which has stronger privacy guarantee against injection attacks. Scalability comparison with other methods and large scale parallelizable experiments using Hadoop are possible extensions of this work.

Acknowledgement. This work is partially supported by MITRE-USM FFRDC under grant 11183.

References

1. Pang, R., Allman, M., Paxson, V., Lee, J.: The devil and packet trace anonymization. ACM SIGCOMM Comput. Commun. Rev. **36**, 29–38 (2006)
2. RIPE (RIPE Network Coordination Centre). https://www.ripe.net/
3. University of Oregon Route Views Project. http://www.routeviews.org/
4. Corporative Association for Internet Data Analysis (CAIDA). https://www.caida.org/tools/taxonomy/anonymization.xml

5. Xu, J., Fan, J., Ammar, M., Moon, S.B.: On the design and performance of prefix-preserving IP traffic trace anonymization. In: Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement, pp. 263–266. ACM (2001)
6. Burkhart, M., Schatzmann, D., Trammell, B., Boschi, E., Plattner, B.: The role of network trace anonymization under attack. ACM SIGCOMM Comput. Commun. Rev. **40**, 5–11 (2010)
7. Brekne, T., Årnes, A., Øslebø, A.: Anonymization of IP traffic monitoring data: attacks on two prefix-preserving anonymization schemes and some proposed remedies. In: Danezis, G., Martin, D. (eds.) PET 2005. LNCS, vol. 3856, pp. 179–196. Springer, Heidelberg (2006)
8. Chaum, D.L.: Untraceable electronic mail, return addresses, and digital pseudonyms. Commun. ACM **24**, 84–90 (1981)
9. Raymond, J.-F.: Traffic analysis: protocols, attacks, design issues, and open problems. In: Federrath, H. (ed.) Designing Privacy Enhancing Technologies. LNCS, vol. 2009, pp. 10–29. Springer, Heidelberg (2001)
10. Slagell, A.J., Li, Y., Luo, K.: Sharing network logs for computer forensics: a new tool for the anonymization of netflow records. In: Workshop of the 1st International Conference on Security and Privacy for Emerging Areas in Communication Networks, pp. 37–42. IEEE (2005)
11. Gamer, T., Mayer, C., Schöller, M.: PktAnon–A Generic Framework for Profile-based Traffic Anonymization. PIK-Praxis der Informationsverarbeitung und Kommunikation **31**, 76–81 (2008)
12. Qardaji, W., Li, N.: Anonymizing network traces with temporal pseudonym consistency. In: 2012 32nd International Conference on Distributed Computing Systems Workshops, pp. 622–633. IEEE (2012)
13. Peuhkuri, M.: A method to compress and anonymize packet traces. In: Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement, pp. 257–261. ACM (2001)
14. Fan, J., Xu, J., Ammar, M.H., Moon, S.B.: Prefix-preserving IP address anonymization: measurement-based security evaluation and a new cryptography-based scheme. Comput. Netw. **46**, 253–272 (2004)
15. Riboni, D., Villani, A., Vitali, D., Bettini, C., Mancini, L.V.: Obfuscation of sensitive data in network flows. In: Proceedings of IEEE INFOCOM, pp. 2372–2380. IEEE (2012)
16. Slagell, A., Yurcik, W.: Sharing computer network logs for security and privacy: a motivation for new methodologies of anonymization. In: Workshop of the 1st International Conference on Security and Privacy for Emerging Areas in Communication Networks, pp. 80–89. IEEE (2005)
17. Slagell, A., Wang, J., Yurcik, W.: Network log anonymization: application of crypto-pan to cisco netflows. In: Proceedings of the Workshop on Secure Knowledge Management (2004)
18. Shebaro, B., Crandall, J.R.: Privacy-preserving network flow recording. Digital Invest. **8**, S90–S100 (2011)
19. Koukis, D., Antonatos, S., Antoniades, D., Markatos, E.P., Trimintzios, P.: A generic anonymization framework for network traffic. In: IEEE International Conference on Communications, pp. 2302–2309. IEEE (2006)
20. King, J., Lakkaraju, K., Slagell, A.: A taxonomy and adversarial model for attacks against network log anonymization. In: Proceedings of the ACM symposium on Applied Computing, pp. 1286–1293. ACM (2009)
21. Sweeney, L.: k-anonymity: a model for protecting privacy. Int. J. Uncertainty Fuzziness Knowl. Based Syst. **10**, 557–570 (2002)

22. Aggarwal, C.C., Yu, P.S.: A condensation approach to privacy preserving data mining. In: Bertino, E., Christodoulakis, S., Plexousakis, D., Christophides, V., Koubarakis, M., Böhm, K., Ferrari, E. (eds.) EDBT 2004. LNCS, vol. 2992, pp. 183–199. Springer, Heidelberg (2004). doi:10.1007/978-3-540-24741-8_12
23. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Oakland, CA, USA, pp. 281–297 (1967)
24. Miller, D.: Softflowd: A Flow-Based Network Traffic Analyser (2013). Mindrot.org
25. Agrawal, D., Aggarwal, C.C.: On the design and quantification of privacy preserving data mining algorithms. In: Proceedings of the Twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pp. 247–255. ACM (2001)