Task 1.2.4.1 Explore existing state-of-the-art anonymization tools starting with CryptoPan and ONTAS

We completed this task in the fourth quarter of the first year of the award. We will continue to update the survey if and as more tools emerge, but this exploration is sufficient to establish the current state of the art.

In completing this task, we were able to draw on a recent and thorough survey of trace anonymization tools by Van Dijkhuizen and Van Der Ham[1]. They discuss the important algorithms and tools that are reported in the literature. Many of the algorithms they discuss were embodied in code but are no longer supported or available. It is useful to note them for completeness, but in many cases have been supplanted by later options.

We also depended on an earlier survey by Tan, Yeo et al [2], from the group at Dartmouth that supports CRAWDAD. They provide some valuable additional background and discuss some additional tools.

For those who want to dig deeper into the various software packages that have been developed, these two papers are an excellent place to start. In particular, we refer the reader to Section 4.3 of the paper by Van Dijkhuizen and Van Der Ham, where Figure 1 provides a summary of the features and status of a number of systems, including some we do not list below, for space reasons.

Here we provide only a brief summary of the important schemes we have identified. The following table draws heavily on their work, with the addition of a few additional schemes not discussed in their papers—in particular ONTAS and PINQ.

In the table, two of the columns have the following meaning:

- Scope refers to the number of fields in the packet. Some schemes deal only with the IP header, some with higher-level fields and/or MAC addresses.
- Method refers specifically to how the IP fields are anonymized. Schemes that anonymize other packet fields have a distinct method for each field. We identify the method they use for IP values as a quick if incomplete way to compare the schemes. In particular, some schemes are *prefix-preserving*, which means that after anonymization IP addresses that had the same prefix in the raw data now have the same prefix in the anonymized version.

Name	Scope	Method	Status
TCPdpriv[1]	IP header only	Prefix preserving	Very early proposal. Not in use.
CryptoPan [3][4][5]	IP address only	Prefix preserving	In active use. Considered state of the art.
Tcprewrite [6]	Layer 2-4	Randomize	Part of tcpreplay. Appears to be actively supported.
TCPmkpub[7]	Layers 2-4	Mixed. See discussion.	Inactive. Old tar file available.
AAPI & Anontool [8]	Extensible. Up to application layer	CryptoPan	Not supported as of 2015.
FLAIM [9]	Extensible. Up to application layer	Multiple	Last release 2008.
PktAnon [10]	Extensible	??	Last release 2011.
Traceanon [11]	Layer 3	Prefix substitution or CryptoPan	Last updated 2010
PCAPAnon [12]	Part of PCAPLib.	??	Production code not available.
TraceWrangler [13]	Advanced layer 3 support	??	Currently supported.
TCPurify [14]	Layer 3	Prefix-revealing, hash of "rest"	Last update 2008. No longer available on github.
ONTAS [15]	Layer2-3	Prefix-revealing	Active support
PINQ [16]	Multiple layers	Differential privacy	Long abandoned.

Discussion

Some schemes are intended for off-line post-processing of a trace. For example, TraceWrangler is a PCAP editor that takes input from sources like Wireshark. Others, like PktAnon and ONTAS are designed to perform the anonymization online as the trace is captured. Online schemes require a great deal of attention to performance, and may compromise the anonymity scheme (as in ONTAS) for performance.

The list reveals a pattern widely observed across academic research, and network measurement in particular with its need for ongoing efforts. Many projects with creative ideas are implemented, but not sustained. The space is littered with expired projects. The important active projects:

- CryptoPan is widely used, both as code and as an algorithm.
- Tcpreplay and TraceWrangler are PCAP editor tools, rather than bulk anonymization tools.

[1] TCPDpriv was written by Greg Minshall in 1996. It appears that the original description is no longer available online. There are many detailed descriptions in secondary sources on the Web.

- ONTAS is a high-performance, line-rate, zero-copy anonymizer. The performance demands limit what transforms can be performed.

A few of the tools warrant some further discussion:

TCPmkpub: This system is interesting for its “split” approach to anonymizing IP addressed. It assumes the trace was captured at the exit point to an edge network, so there would be internal addresses (which might require greater privacy protection) and external IP addresses. TCPmkpub uses CryptoPan to anonymize the external addresses, but for the internal addresses, the algorithm uses a table of prefixes with lengths. The prefix part is mapped to an identifier unique to the prefix, and the host part is anonymized by a pseudo-random permutation that is considered harder to break than CryptoPan.

ONTAS: ONTAS is significant because of its performance objective. The IP anonymization scheme is as follows: for specific prefixes, it preserves the prefix and hashes the rest of the address. Otherwise, it leaves the address visible. As an example of where this approach might be useful, if a trace is captured at the exit point from a campus network, ONTAS could be configured to transform the prefixes associated with the campus, and leave the external addresses intact. This would resemble what TCPmkpub does, at line rate.

PINQ: PINQ is distinctive and worthy of attention because it uses differential privacy (DP) to protect PII in packet traces and flow data. It does not reveal anonymized fields in the trace records, but instead allows the researcher to run queries on the raw data. The queries are run by a trusted agent with access to the raw data. The paper does not discuss whether some “pre-anonymization” of the data would reduce the level of trust required of the query agent while still permitting useful queries. The paper provides several examples of how to combine the basic query types of a typical DP query software package to create some complex queries. They make the point that the ability to use DP to execute useful queries depends on the ingenuity of the programmer. Since the basic building blocks of DP may not be a familiar starting point for typical query composition, the explicit examples in this paper are valuable. The actual PINQ code has not been available for some time, but as a part of Task 1.3.2 we intend to see if we can preproduce some of these queries using modern DP software.

Bibliography

[1] N. V. Dijkhuizen and J. V. D. Ham, “A Survey of Network Traffic Anonymisation Techniques and Implementations,” *ACM Comput. Surv.*, vol. 51, no. 3, pp. 1–27, May 2019, doi: 10.1145/3182660.

[2] Tan, Keren, Yeo, Jihwang, Locasto Michael, and Kotz, David, “Catch, Clean, and Release: A Survey of Obstacles and Opportunities for Network Trace Sanitization,” in *Privacy-Aware Knowledge Discovery*, 0 ed., F. Bonchi and E. Ferrari, Eds. CRC Press, 2010, pp. 139–170. doi: 10.1201/b10373-13.

[3] J. Fan, J. Xu, M. H. Ammar, and S. B. Moon, “Prefix-preserving IP address anonymization: measurement-based security evaluation and a new cryptography-based scheme,” *Comput. Netw.*, vol. 46, no. 2, pp. 253–272, 2004.

[4] T. Brekne, A. Årnes, and A. Øslebø, “Anonymization of ip traffic monitoring data: Attacks on two prefix-preserving anonymization schemes and some proposed remedies,” in *International Workshop on Privacy Enhancing Technologies*, 2005, pp. 179–196.

[5] S. E. Coull, C. V. Wright, A. D. Keromytis, F. Monrose, and M. K. Reiter, “Taming the devil: Techniques for evaluating anonymized network data,” 2008.

[6] Klassen, Fred, “Tcpreplay - Pcap editing and replaying utilities.” http://tcpreplay.appneta.com/ (accessed Jul. 02, 2022).

[7] R. Pang, M. Allman, V. Paxson, and J. Lee, “The devil and packet trace anonymization,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 1, pp. 29–38, Jan. 2006, doi: 10.1145/1111322.1111330.

[8] D. Koukis, S. Antonatos, D. Antoniades, E. P. Markatos, and P. Trimintzios, “A generic anonymization framework for network traffic,” in *2006 IEEE International Conference on Communications*, 2006, vol. 5, pp. 2302–2309.

[9] A. J. Slagell, K. Lakkaraju, and K. Luo, “FLAIM: A Multi-level Anonymization Framework for Computer and Network Logs.,” in *LISA*, 2006, vol. 6, pp. 3–8.

[10] T. Gamer, C. Mayer, and M. Schöller, “Pktanon–a generic framework for profile-based traffic anonymization,” 2008.

[11] “TraceAnon – libtrace.” https://wand.net.nz/trac/libtrace/wiki/TraceAnon (accessed Jul. 02, 2022).

[12] Y.-D. Lin, P.-C. Lin, S.-H. Wang, I.-W. Chen, and Y.-C. Lai, “Pcaplib: A system of extracting, classifying, and anonymizing real packet traces,” *IEEE Syst. J.*, vol. 10, no. 2, pp. 520–531, 2014.

[13] “TraceWrangler - Packet Capture Toolkit.” https://www.tracewrangler.com/ (accessed Jul. 02, 2022).

[14] M. Peuhkuri, “A method to compress and anonymize packet traces,” in *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, 2001, pp. 257–261.

[15] H. Kim and A. Gupta, “ONTAS: Flexible and Scalable Online Network Traffic Anonymization System,” in *Proceedings of the 2019 Workshop on Network Meets AI & ML - NetAI’19*, Beijing, China, 2019, pp. 15–21. doi: 10.1145/3341216.3342208.

[16] F. McSherry and R. Mahajan, “Differentially-private network trace analysis,” p. 12.