Data minimisation of DNS traffic

The DNS Privacy solutions presented here ensure that DNS queries made by an individual end user can't be observed by eavesdroppers as they pass across the Internet. Only the operators of DNS privacy servers have access to the details of the queries. For operational reasons such as monitoring server performance or detecting and mitigating attacks operators need to keep logs of the DNS queries they see; in some circumstances they may need to share those logs with other operators. To preserve end user privacy, as RFC6973[17] observes it is important that the data in these logs limits the identifiability of end users; more generally, that the data in the logs is kept to the minimum required for purpose, a process the RFC terms *data minimisation*.

Data minimising a trace or logs of network traffic therefore includes ensuring the recorded data does not contain privacy-sensitive information. This is typically personal data, or data that can be used to link a record to an individual, but may also include revealing other confidential information, for example on the structure of an internal corporate network. In the case of identifiability, this means that if individual user identifiers cannot be omitted altogether, pseudonyms should be used instead.

The problem of effectively ensuring that DNS query logs do not contain privacy-sensitive information is not one that currently has a generally agreed solution. This page gives an overview of current approaches to identifier pseudonymisation. As RFC7626[16] makes clear, the big privacy risk in DNS is connecting DNS queries to an individual, so at present the main focus is on pseudonymising client IP addresses (though, of course, the MAC address, VLAN identifier and ARP data may be useful in particularly localised environments).

Terminology

Pseudonymising a dataset is generally done using either anonymisation or pseudonomysation. This discussion uses the definitions from RFC6973[17] Sectio n 3, with additional observations from van Dijkhuizen et al.[1]

- *Anonymisation*. To enable anonymity of an individual, there must exist a set of individuals that appear to have the same attribute(s) as the individual. To the attacker or the observer, these individuals must appear indistinguishable from each other.
- *Pseudonymisation*. The true identity is deterministically replaced with an alternate identity (a pseudonym). When the pseudonymisation schema is known, the process can be reversed, so the original identity becomes known again.

In practice there is a fine line between the two; for example, how to categorise a deterministic algorithm for pseudonymisation of IP addresses that produces a group of pseudonyms for a single given address?

History

Awareness of the need for pseudonymising data, so that significant corpuses of captures could be shared for research purposes, sparked research into particularly IP address pseudonymisation in the late 1990s/early 2000s. Several techniques reflecting different requirements for the pseudonymised addresses and different performance/resource tradeoffs emerged over the course of the decade. Developments over the last decade have been both a blessing and a curse; the large increase in size between an IPv4 and an IPv6 address, for example, renders some techniques (in particular TSA) impractical, but also makes available a much larger amount of input entropy. Pseudonymised IPv6 addresses are therefore much better placed to resist brute force re-identification attacks than IPv4 addresses. Several authors (e.g. Brenker & Arnes[11]) have observed that today any IPv4 address pseudonymisation is vulnerable to a brute force attack, particularly if an attacker is capable of ensuring packets are captured by the target and the attacker can send forged traffic with arbitrary source and destination addresses to the target thus permitting an attack along the lines of a cryptographic chosen plaintext attack.

Categorising techniques for anonymising logs

The ways in which data may be pseudonymised can be classified into some broad categories.

- *Replacement*. A one-to-one replacement of a field to a new value of the same type, for example using a regular expression.
- *Filtering*. Removing (and thus truncating) or replacing data in a field. Field data can be overwritten, often with zeros, either partially (*grey marking*) or completely (*black marking*).
- *Generalisation*. Data is replaced by more general data with reduced specificity. One example would be to replace all TCP/UDP port numbers with one of two fixed values indicating whether the original port was ephemeral (>=1024) or non-ephemeral (>1024). Another example, *precision degredation*, reduces the accuracy of e.g. a numeric value or a timestamp.
- *Enumeration*. With data from a well-ordered set, replace the first data item data using a random initial value and then allocate ordered values for subsequent data items. When used with timestamp data, this preserves ordering but loses precision and distance.
- *Reordering/shuffling*. Preserving the original data, but rearranging its order, often in a random manner.
- *Random substitution*. As replacement, but using randomly generated replacement values.
- *Cryptographic permutation*. Using a permutation function, such as a hash function or cryptographic block cipher, to generate a replacement de-identified value.

This list is derived from RFC6235 and van Dijkhuizen et al.[1]

A pseudonymising technique may also have properties desirable in a particular application:

- *Format-preserving encryption*. Normally when encrypting, the original data length and patterns in the data should be hidden from an attacker. Some applications of data minimisation, such as network capture pseudonymisation, require that the pseudonymised data is of the same form as the original data, to allow the data to be parsed in the same way as the original.
- *Prefix preservation*. Values such as IP addresses and MAC addresses contain prefix information that can be valuable in analysis, e.g. manufacturer ID in MAC addresses, subnet in IP addresses. Prefix preservation ensures that prefixes are pseudonymised consistently; e.g. if two IP addresses are from the same subnet, prefix preserving pseudonymising will ensure that their pseudonymised counterparts will also share a subnet. Prefix preservation may be fixed (the length of the prefix to be preserved must be set by the user in advance), or general (if two addresses share any length of prefix bits in common, their pseudonymised counterparts will also have the same length of prefix bits in common).

Notable pseudonymising techniques

Google Analytics non-prefix filtering

Since May 2010, Google Analytics has provided a facility that allows website owners to request that all their users IP addresses are pseudonymised within Google Analytics processing. This very basic pseudonymisation simply sets to zero the least significant 8 bits of IPv4 addresses, and the least significant 80 bits of IPv6 addresses. The level of pseudonymisation this produces is perhaps questionable. There are some analysis results[13] which suggest that the impact of this on reducing the accuracy of determining the user's location from their IP address is less than might be hoped; the average discrepancy in identification of the user city for UK users is no more than 17%. *Anonymisation: Format-preserving, Filtering (grey marking)*.

dnswasher

Since 2006, PowerDNS have included a data minimisation tool dnswasher[14] with their PowerDNS product. This is a PCAP filter that performs a one-to-one mapping of end user IP addresses with an pseudonymised address. A table of user IP addresses and their de-identified counterparts is kept; the first IPv4 user addresses is translated to 0.0.0.1, the second to 0.0.0.2 and so on. The de-identified address therefore depends on the order that addresses arrive in the input, and running over a large amount of data the address translation tables can grow to a significant size. *Anonymisation: Format-preserving, Enumeration*.

Prefix-preserving map

Used in TCPdpriv[2], this algorithm stores a set of original and pseudonymised IP address pairs. When a new IP address arrives, it is compared with previous addresses to determine the longest prefix match. The new address is pseudonymised by using the same prefix, with the remainder of the address pseudonymised with a random value. The use of a random value means that TCPdrpiv is not deterministic; different pseudonymised values will be generated on each run. The need to store previous addresses means that TCPdpriv has significant and unbounded memory requirements, and because of the need to allocated pseudonymised addresses sequentially cannot be used in parallel processing. *Anonymisation: Format-preserving, prefix preservation (general), replacement, random substitution*.

Cryptographic Prefix-Preserving Pseudonymisation

Cryptographic prefix-preserving pseudonymisation was originally proposed as an improvement to the prefix-preserving map implemented in TCPdpriv, described in Xu et al[3] and implemented in the Crypto-PAn tool[4]. Crypto-PAn is now frequently used as an acronym for the algorithm. Initially it was described for IPv4 addresses only; extension for IPv6 addresses was proposed in Harvan & Schonwalder[5] and implemented in snmpdump[6]. This uses a cryptographic algorithm rather than a random value, and thus pseudonymity is determined uniquely by the encryption key, and is deterministic. It requires a separate AES encryption for each output bit, so has a non-trivial calculation overhead. This can be mitigated to some extent (for IPv4, at least) by pre-calculating results for some number of prefix bits. *Pseudonymisation: Format-preserving, prefix preservation (general), cryptographic permutation.*

Top-hash Subtree-replicated Anonymisation

Proposed in Ramaswamy & Wolf[7], Top-hash Subtree-replicated Anonymisation (TSA) originated in response to the requirement for faster processing than Crypto-PAn. It used hashing for the most significant byte of an IPv4 address, and a pre-calculated binary tree structure for the remainder of the address. To save memory space, replication is used within the tree structure, reducing the size of the pre-calculated structures to a few Mb for IPv4 addresses. Address pseudonymisation is done via hash and table lookup, and so requires minimal computation. However, due to the much increased address space for IPv6, TSA is not memory efficient for IPv6. *Pseudonymisation: Format-preserving, prefix preservation (general), cryptographic permutation.*

ipcipher

A recently-released proposal from PowerDNS[8], ipcipher is a simple pseudonymisation technique for IPv4 and IPv6 addresses. IPv6 addresses are encrypted directly with AES-128 using a key (which may be derived from a passphrase). IPv4 addresses are similarly encrypted, but using a recently proposed (and confusingly closely named) encryption ipcypher[9] suitable for 32bit block lengths. However, the author of ipcrypt has since indicated[10] that it has low security, and further analysis has revealed it is vulnerable to attack. At the time of writing, progress on ipcipher appears to have stalled. *Pseudony misation: Format-preserving, cryptographic permutation*.

Bloom filters

van Rijswijk-Deij et al[15] have recently described work using Bloom filters to categorise query traffic and record the traffic as the state of multiple filters. By this means, it is possible to determine with a high probability if, for example, a particular query was made, but the set of queries made cannot be recovered from the filter. Similarly, by mixing queries from a sufficient number of users in a single filter, it becomes practically impossible to determine if a particular user performed a particular query. Large numbers of queries can be tracked in a memory-efficient way. As filter status is stored, this approach cannot be used to regenerate traffic, and so cannot be used with tools used to process live traffic. *Anonymisation: Generalisation*.

Other data minimisation considerations

(A placeholder list).

TTL/Hoplimit (if present) can be used to fingerprint client OS.

MAC address/VLAN.

DNS ID lack of randomisation ditto.

All queries down a single TCP stream must come from the same host.

References

1. Niels van Dijkhuizen and Jeroen van der Ham. 2018. A Survey of Network Traffic Anonymisation Techniques and Implementations. ACM Comput. Surv. 51, 3, Article 52 (May 2018), 27 pages. DOI: https://doi.org/10.1145/3182660
2. G. Minshall, Ipsilon Networks, Inc., TCPDRIV, Oct 2005. http://ita.ee.lbl.gov/html/contrib/tcpdpriv.html
3. J. Xu, J. Fan, M. Ammar, and S. B. Moon, Prefix-Preserving IP Address Anonymization: Measurement-Based Security Evaluation and a New Cryptography-Based Scheme, Proceedings of the IEEE International Conference on Network Protocols, 2002. DOI: https://doi.org/10.1016/j.comnet.2004.03.033. http://an.kaist.ac.kr/~sbmoon/paper/intl-journal/2004-cn-anon.pdf
4. CryptoPan page at Georgia Tech. https://www.cc.gatech.edu/computing/Telecomm/projects/cryptopan/
5. M. Harvan and J. Schonwalder. Prefix- and lexicographical-order-preserving IP address anonymization. In Proceedings of the 10th IEEE/IFIP Network Operations and Management Symposium 2006 (NOMS'06). 519–526. DOI: http://dx.doi.org/10.1109/NOMS.2006.1687580. http://mharvan.net/talks/noms-ip_anon.pdf
6. J. Schonwalder, snmpdump, 2004, https://github.com/schoenw/snmpdump
7. R. Ramaswamy and T. Wolf, "High-Speed Prefix-Preserving IP Address Anonymization for Passive Measurement Systems," in *IEEE/ACM Transactions on Networking*, vol. 15, no. 1, pp. 26-39, Feb. 2007. DOI: https://doi.org/10.1109/TNET.2006.890128. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.84.7417&rep=rep1&type=pdf
8. A. Hubert, PowerDNS, Feb 2018. https://github.com/PowerDNS/ipcipher. See also On IP address encryption: security analysis with respect for privacy, Medium.com, May 2018 by the same author.
9. J.-P. Aumasson, Mar. 2015, https://github.com/veorq/ipcrypt
10. J.-P. Aumasson, Re: [Cfrg] Analysis of ipcrypt?, Feb. 2018. https://www.ietf.org/mail-archive/web/cfrg/current/msg09494.html. See also J. Donenfeld, Feb. 2018. https://www.ietf.org/mail-archive/web/cfrg/current/msg09495.html
11. T. Brekne and A. Årnes. "Circumventing IP-address pseudonymization." *Communications and Computer Networks* (2005). https://pdfs.semanticscholar.org/7b34/12c951cebe71cd2cddac5fda164fb2138a44.pdf
12. IP Anonymization in Analytics, https://support.google.com/analytics/answer/2763052?hl=en
13. Huiyan, Anonymize IP Geolocation Accuracy Impact Assessment (blog posting), May 2017.https://www.conversionworks.co.uk/blog/2017/05/19/anonymize-ip-geo-impact-test/
14. dnswasher - A PowerDNS nameserver debugging tool, 2006
15. R. van Rijswijk-Deij, M. Bomhoff and R. Dolmans, Let a Thousand Filters Bloom: DNS-Based Threat Monitoring That Respects User Privacy, TNC18, June 2018, https://tnc18.geant.org/core/presentation/127
16. S. Bortzmeyer, DNS Privacy Considerations, RFC7626, https://datatracker.ietf.org/doc/rfc7626/
17. A. Cooper, H. Tschofenig, B. Aboba, J. Peterson, J. Morris, M. Hansen and R. Smith, Privacy Considerations for Internet Protocols, RFC6973, https://datatracker.ietf.org/doc/rfc6973/