Poster: A Framework for Distributed Anonymous Data Collection and Feedback

Maxim Timchenko, Ari Trachtenberg
Electrical and Computer Engineering Department, Boston University
{maxvt, trachten}@bu.edu

***Abstract*—Diagnostic, usage, and statistical data collection occurs continuously in the background on our computers and smart devices. However, unless the data is particularly sensitive (say, for medical research) or there has been a recent and severe security failure covered by the media, the privacy and anonymity of the process or of the resulting data set are seldom given much thought by device owners. We thus propose and are in the process of implementing and evaluating a framework for non-realtime anonymous data collection, aggregation for analysis, and feedback. Departing from the usual "trusted core" approach, we aim to maintain the reporting agent's anonymity, even if the centralized part of the system is compromised. We design a peer-to-peer mix network tuned to carry data to a centralized repository while maintaining (i) source anonymity, (ii) privacy in transit, (iii) the ability to provide feedback from central server to source.**

I. Background

In practice, the anonymity and privacy of diagnostic data is often given lower priority than efficiency and simplicity of collection. A notable case in point are the crash submissions of some versions of Microsoft Windows, whose unencrypted contents have reportedly been intercepted and used for target reconnaissance [6].

A common implementation of a collection system contains a "trusted core", which receives and analyzes all the data in the system while also anonymizing and aggregating, as needed, to provide a sanitized version of the data to "untrusted applications". This approach carries the privacy risk of a central point of failure if the trusted core is compromised. Furthermore, since a direct connection is made between the reporter and the core, the IP address of the reporter could be recorded and associated with an anonymous (but typically fixed) machine identifier accompanying many diagnostic messages.

Recent work takes a more robust, if specialized, approach. By focusing on specific data types and introducing noise into the data, it is possible to achieve anonymity in the differential privacy context. For example, *PrivEx* [3] captures countable events (number of visits to particular websites, traffic volume, etc.) and uses cryptographic constructions, whereas *Non-tracking Web Analytics* [1] performs distributed aggregation. Another interesting approach is to rely on Tor for obfuscating the reporter-collector path, as done by *Anonygator* [5] as an intermediate step between the reporter and the aggregator, and *ANONIZE* [4] to hide the origin of a filled survey.

Finally, we observe that many of the described systems make no provision for a feedback channel (communication back to the reporter), which can be instrumental for providing meaningful value to the reporter in exchange for their data.

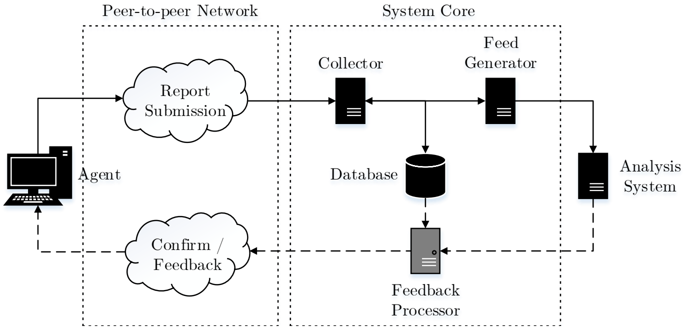

Fig. 1. Major components of the proposed system

II. Description

We consider the major features of common data collection streams (such as those used for diagnostic or threat assessment) that are not shared by general purpose communication streams, and propose a design that specifically takes advantage of those features rather than leveraging a general purpose anonymity network.

First of all, our streams are asymmetric. A sizable amount of data flows from the reporter, while the feedback might be absent entirely or contain a very small amount of data. This suggests using different algorithms in the reporting and feedback directions.

Secondly, the streams are typically loss tolerant. If a particular bug is common, it is likely that many customers will report it, so the loss of a single such report is not critical. This allows us to use simpler, stochastic algorithms that make local decisions but do not guarantee timely end-to-end delivery.

Finally, our data streams are not required to be realtime nor interactive. For majority of our use cases, a bounded delay is acceptable, letting us hold messages in mix pools for longer to increase the anonymity set.

A. Implementation

In the forward direction, messages are encrypted and passed through a peer-to-peer network of mixing relays. Communication between each pair of relays is encrypted as well, but we do not use onion encryption [7] or prenegotiated routes through the network. Instead, each relay chooses the next destination of a message, which might be either another relay or, with some fixed probability, the collector.

In the feedback direction, we offer a "mostly correct" binary feedback message (which could be a notification of delivery) through the use of public (or broadcast) Bloom filters. The size of many practical networks allows us to use a linearly scaling feedback channel with very little computational effort. We also allow the possibility of using free-form (non-binary) feedback; in this case, our Bloom filter indicates whether a free-form feedback file should be retrieved, with the tacit assumption that most interactions do not require such feedback.

We observe that for many use cases, reaching anonymity in the context of differential privacy is rather difficult, particularly when the data is not numerical / bucketable in nature. For example, the use case of collecting DNS traffic from hosts and analyzing it to detect potential malware activity takes, as input, domain names being resolved by the host. Trying to add noise to a domain name will likely result in a name that does not actually exist, and reducing the domain name to a value that can be added to a histogram will result in buckets containing both benign and malicious domain names, so detection will not be accurate.

III. Evaluation

There are many algorithm parameters that should be tuned based on the desired anonymity properties and the nature of the feedback traffic (frequency, size, regularity and so on). Fortunately, the simplicity of the algorithm leads to a straightforward analysis using queuing-theoric tools. We are building a message-level event-based network simulator of the protocol to validate the findings of the theoretical analysis and to test the reaction of the system to disruptive events, such as network partitioning. At this time, the forward path of the simulation is complete, and the feedback path is being written.

We are also characterizing the optimal loads for the proposed design. Small reports sent in a globally stable pattern tend to work best; large reports whose amounts can vary widely between different times on the network (for example, automated crash submissions for a single software product) can fare poorly.

From the security evaluation perspective, we consider three main threats: an attacker monitoring the system's traffic across a large section of the network, an attacker running a large number of relays (a Sybil attack), and an attacker compromising the "system core" and gaining access to the information contained in it. While protecting the reporting agent's anonymity is our main goal, we also consider potential disruption created by fake reports or by tampering with data and the potential to misuse the proposed system for unrelated goals, for example by attempting to use its UDP-based protocol as a distributed denial of service attack multiplier.

To test our design on a practical use case, we are writing an implementation of our design that collects DNS traffic to and from a host and detects queries likely to be related to malware activity. This use case requires at least some form of feedback (binary for coarse notification, free-form for reporting specific suspicious queries) to be able to alert the reporter when a suspicious pattern is detected. We intend to use an already existing machine learning-based analysis system EXPOSURE [2] to perform the detection. From the implementation perspective, capture and filtering of DNS packets at the host has been implemented using the open-source *libpcap* library, and end-to-end flow of data in the forward direction (single packets assembled into reports, forwarded to the collector and output to an aggregated file) has been achieved. Most of the protocol implementation, however, has been deferred until the network simulation shows satisfactory performance of the algorithms and the chosen parameter values, as the simulation allows much faster iteration and easier debugging.

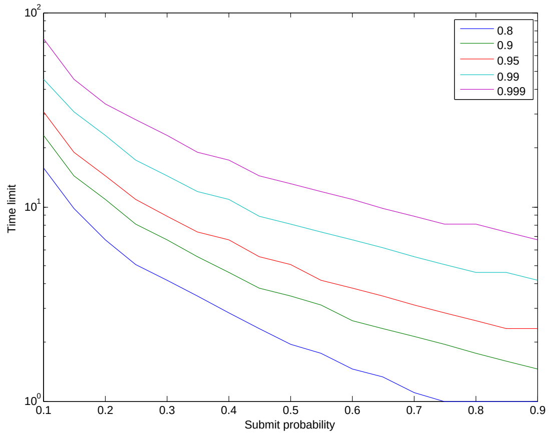

Fig. 2. Relationship between the probability of successful delivery and required time

References

[1] Istemi Ekin Akkus et al. "Non-tracking web analytics". In: *Proceedings of the 2012 ACM conference on Computer and communications security*. ACM. 2012, pp. 687–698.

[2] Leyla Bilge et al. "EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis." In: *NDSS*. 2011. URL: http://www.iseclab.org/papers/bilge-ndss11.pdf.

[3] Tariq Elahi, George Danezis, and Ian Goldberg. "PrivEx: Private Collection of Traffic Statistics for Anonymous Communication Networks". In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM. 2014, pp. 1068–1079.

[4] Susan Hohenberger, Steven Myers, Rafael Pass, et al. "ANONIZE: A Large-Scale Anonymous Survey System". In: *Security and Privacy (SP), 2014 IEEE Symposium on*. IEEE. 2014, pp. 375–389.

[5] Krishna PN Puttaswamy, Ranjita Bhagwan, and Venkata N Padmanabhan. "Anonygator: Privacy and integrity preserving data aggregation". In: *Middleware 2010*. Springer, 2010, pp. 85–106.

[6] SPIEGEL Staff. *Inside TAO: Documents Reveal Top NSA Hacking Unit*. Dec. 29, 2013. URL: http://www.spiegel.de/international/world/the-nsa-uses-powerful-toolbox-in-effort-to-spy-on-global-networks-a-940969-2.html (visited on 04/04/2015).

[7] Paul F Syverson, David M Goldschlag, and Michael G Reed. "Anonymous connections and onion routing". In: *Security and Privacy, 1997. Proceedings., 1997 IEEE Symposium on*. IEEE. 1997, pp. 44–54.