P2P Based Anonymous Data Collection System

Zirui Zhao, Jinyang Li, Tai-Sheng Cheng
University of Illinois at Urbana-Champaign

1 Introduction

To improve product quality and user experiences, more and more companies are collecting user's anonymous data. Typically, it will be three steps for collecting anonymous data [10]. First, gather user's insensitive information on client side; Second, send the information back to company's database or other cloud platform [10]. Third, analyze collected data. For each of step, there is possibility of leaking user's sensitive data. In the first step, the client side software or JavaScript may collect user's privacy actively, which is known as malware or spyware. In the second step, user may expose sensitive information such as ISP or Geolocation due to the network transmission which can be passively collected. In the third step, the companies may find correlation between the anonymous data they collected with other open data to reveal user's identity [3]. To protect people from leaking their sensitive information, we have software analysis to deal with malware and spyware in the first stage [6] and differentiate privacy [3] in the third stage. However, we do not have many solutions to prevent data leak during the second stage.

Sensitive information can be leaked easily in the second stage, via device fingerprints [13]. Typically, device fingerprints can be acquired passively through browser's behavior or even lower layers of TCP/IP model [9] [8]. When the user connects directly to the company's server to submit anonymous data, the user's public IP address can easily be acquired in transport layer and the physical location will be exposed by querying IP Geolocation database [7]. Considering most companies are using HTTP or HTTPS protocol for data transmission, they can easily acquire more sensitive data by checking HTTP header. For example, "X-Forwarded-For" field tells the company if the user is using proxy and what the original IP is, and "User-Agent" field tells which browser and operating system the user is using. When it comes to web applications, there will be more ways to track the user down. The web application provider can use DNS Leak to find out which DNS server the user is using and use WebRTC to acquire the user's internal IP address. Besides, the JavaScript's behavior, screen resolution, Flash and Java plugins are other side-channels can be used by the snoopers [1]. According to Mozilla's report, "83.6% of the browsers seen had a unique fingerprint; among those with Flash or Java enabled, 94.2%" [1], which does not include cookies. These fingerprint data can be used to link with the collected anonymous data. Attackers can easily analyze users preferences or behaviors according to their location or language based on these so called anonymous data, which may lead to specially targeted scam and virus, or Advanced Persistent Threat (APT) [4] such as social engineering.

Traditional ways of privacy protection in second phase are using VPN or Tor to hide the user's real identity. However, these techniques which try to hide user's IP are not enough. By using device fingerprint, the company can still track the user behind the Tor [12]. To solve the problem, we introduce our P2P based anonymous data collection system. Instead of trying to hide the identity, we decouple the relation

between device fingerprint and collected data. Tracking the origin of anonymous data will be like finding needle in a haystack.

In our P2P based anonymous data collection system, each peer will hold a queue for each registered application. The queue is filled with fake data specially generated and encrypted with the company's public key. When an anonymous data record is generated by the user, it will be encrypted with company's public key and pushed into the queue then pop an old record out. Each peer will randomly pick a data record and broadcast it to other peers after a certain period of time. So, the queue will be filled with the mix of user's own data, other users's data, and fake data generated by the company gradually. Because the whole procedure picks data record randomly, people cannot know where a data record is from. While submitting the data record to the company, the queue will be sent in a random order. This mechanism decouples the user's device fingerprint with the data submitted.

To prevent bad peer who may inspect or malicious tamper other user's data, the data record contains anonymous data and its hash value, then it is encrypted by company's public key. Also, each peer has a speed limiter to block the bad peer who may spam garbage into the P2P network. On the company side, the server will decrypt data record with their private key, filter all the fake data and duplicated data based on its hash value. A possible optimization for the system is running this P2P network as a system level service, to avoid each application implements an independent P2P service and occupy a network port.

2 Related Work

2.1 Device Fingerprint

The first empirical study of fingerprint was conducted by Eckersley. He developed a way to generate fingerprint based on browser's extension [5]. The website, panopticlick.eff.org, maintained by him, collected more than 50,000 users browsers' information. Based on those data, his method achieved more than 18bits entropy, which means it can identify at least 262,144 browsers before two browsers are considered as the same one on average. Besides using front-end technologies generate fingerprint. Fingerprints can also be generated from lower layer of network stack. Christoph did large-scale study on passive 802.11 device fingerprint [8]. Therefore, the device fingerprint problem occurs both in web applications and traditional desktop software.

2.2 Tor

Tor is an anonymity network software which helps people hiding their real identity. A Tor connection has three Tor nodes involved [2], each node does not know where the packet is come from or where it is going because the packet is encrypted by these nodes' public key. This mechanism ensures anonymity. However, Tor does not work well in this scenario. Yi Shi proposed a way to create user's fingerprinting by monitoring incoming and outcoming packets in 2009 [11]. After Yi, Tao Wang improved the quality of fingerprint classification results by introducing new data processing way and fingerprinting metrics in 2013 [12].

References

[1] https://wiki.mozilla.org/fingerprinting.

[2] https://www.torproject.org/.

[3] Bittau, A., Erlingsson, U., Maniatis, P., Mironov, I., Raghunathan, A., Lie, D., Rudominer, M., Kode, U., Tinnes, J., and Seefeld, B. Prochlo: Strong privacy for analytics in the crowd. In Proceedings of the 26th Symposium on Operating Systems Principles (2017), ACM, pp. 441–459.

[4] Daly, M. K. Advanced persistent threat. Usenix, Nov 4, 4 (2009), 2013–2016.

[5] Eckersley, P. How unique is your web browser? In International Symposium on Privacy Enhancing Technologies Symposium (2010), Springer, pp. 1–18.

[6] Herley, C. E., Keogh, B. W., Hulett, A. M., Marinescu, A. M., Williams, J. S., and Nurilov, S. Spyware detection mechanism, Apr. 28 2015. US Patent 9,021,590.

[7] Katz-Bassett, E., John, J. P., Krishnamurthy, A., Wetherall, D., Anderson, T., and Chawathe, Y. Towards ip geolocation using delay and topology measurements. In Proceedings of the 6th ACM SIGCOMM conference on Internet measurement (2006), ACM, pp. 71–84.

[8] Neumann, C., Heen, O., and Onno, S. An empirical study of passive 802.11 device fingerprinting. In Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on (2012), IEEE, pp. 593–602.

[9] Nikiforakis, N., Kapravelos, A., Joosen, W., Kruegel, C., Piessens, F., and Vigna, G. Cookieless monster: Exploring the ecosystem of web-based device fingerprinting. In Security and privacy (SP), 2013 IEEE symposium on (2013), IEEE, pp. 541–555.

[10] Papadimitriou, A., Bhagwan, R., Chandran, N., Ramjee, R., Haeberlen, A., Singh, H., Modi, A., and Badrinarayanan, S. Big data analytics over encrypted datasets with seabed. In OSDI (2016), pp. 587–602.

[11] Shi, Y., and Matsuura, K. Fingerprinting attack on the tor anonymity system. In International Conference on Information and Communications Security (2009), Springer, pp. 425–438.

[12] Wang, T., and Goldberg, I. Improved website fingerprinting on tor. In Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society (2013), ACM, pp. 201–212.

[13] Yen, T.-F., Xie, Y., Yu, F., Yu, R. P., and Abadi, M. Host fingerprinting and tracking on the web: Privacy and security implications. In NDSS (2012), vol. 62, Citeseer, p. 66.