Detecting IP-tracking proof interfaces by looking for NATs

Aurélien Buchet
UCLouvain

Peter Snyder
Brave Software

Hamed Haddadi
Imperial College London
Brave Software

Cristel Pelsser
UCLouvain

***Abstract*—In this poster, we propose an approach based on short-lived random identifiers to allow applications to detect when multiple users share the same IP address such as when they are behind NATs. Using NATed interfaces could provide a cheap way to evade IP-based tracking as the traffic of all users is merged into a single IP flow. As a result, it is harder for trackers to single out (and so re-identify by IP address) users behind a NAT. For many years, there has been a race between web trackers trying to find techniques to monitor user behaviour online, and privacy researchers looking for solutions to avoid such tracking. Despite progresses in browser privacy-preserving techniques, IP tracking is still highly effective because current solutions to hide an IP address such as VPNs, or the Tor network, rely on external services and often induce a high cost in terms of performance. Our proposal could lead to solutions that are cheaper to deploy and don't affect the performance as much. We developed an Android application detecting when an IP address was shared by multiple devices and reported the availability of such interfaces. We show that it is possible to identify networks where multiple users share the same IP address. We also discuss how our system can be protected from potential attackers.**

I. Introduction

The goal of web tracking is to gather data about users' behaviour and browsing habits on the web. One way web trackers follow users is through their IP address. This is difficult to defend against at the application layer, as it has limited control over the network. In this paper, we present an idea to allow for a cheap way to hide the IP address from potential trackers. Richter et al. [22] studied the deployment of Carrier-Grade NATs (CGN) and showed that it is difficult to isolate and count users behind a CGN.

Our goal is (i) to propose a tool for the detection of multi-user NATs and deploy it (ii) to study how often devices have the opportunity to use an interface that is hard to uniquely identify to a user. We want to expose a minimal amount of information about the users of our tool while still collecting enough data to be able to perform our study. In our approach, we don't need accounts to authenticate users. However, we use IDs that are created locally and allow us to identify devices. We keep track of the results for each device so the system is not designed to provide full anonymity for the users. Here, we propose a design for a service allowing users to detect which of their interfaces, if any, is behind a NAT and if there are other users of the service using the same NAT. Because multiple users share the same public IP address, it can be difficult to identify individual users and track their activities. We implemented the service and show preliminary results of measurements from a small set of users.

978-3-903176-58-4 ©2023 IFIP

II. Background

Web Tracking. Tracking services usually use small JavaScript files or 1 pixels images embedded in pages that cause browsers to make requests to their servers [15]. They can then use different techniques to identify the user based on their requests. Well established techniques to track a user include the usage of cookies and browser fingerprinting [13]. Most of today's browsers include mechanisms to prevent the tracking of their users [26], [23], [6], [3]. Some extensions are also designed to protect a user's privacy on the web [1], [7], [8], [9]. This led to a race between trackers and blockers, to find new ways to identify users versus to protect their privacy. Recently, Mishra et al. [16] showed that it was possible to uniquely identify most users through a set of unique long-lived IP addresses that mapped directly to a device.

NATs. With the number of available IPv4 addresses rapidly reducing, some ISPs rely on CGNs to allow multiple users to share a single public IP address [14]. This is achieved through the use of Network Address Translation (NAT), which maps private IP addresses used on local networks to a public IP address used on the Internet. CGN works by assigning each user a unique private IP address on the local network, which is then translated to a public IP address when the user connects to the Internet. When multiple users connect to the Internet using the same public IP address, their traffic is distinguished by unique port numbers assigned by the NAT device. This allows the NAT device to route the traffic to the correct user on the local network.

Private networks. The Tor private network [5] provides anonymity by routing the Internet traffic through a series of servers, called nodes, that are operated by volunteers around the world. Each node only knows the IP address of the previous node and the next node in the chain, so no single node knows both the origin and the destination of the traffic. This is achieved by using cells that are encrypted once for every connection between Tor nodes leading to higher delays. Unfortunately, Tor usage may raise suspicions of the user's intent and is thus not recommended in some locations. For instance, connections to Tor bridges are blocked aggressively

in several countries such as China, Tanzania and Venezuela [24]. VPNs, or virtual private networks, are services that provide a secure and private connection between a device and the Internet. They work by creating an encrypted tunnel between the device and a remote server, which can be located anywhere in the world. When a piece of equipment connects to the Internet through a VPN, its Internet traffic is routed through this encrypted tunnel. The only public IP address that is seen by the destination is the one used by the VPN server. iCloud Private Relay [12] uses a two-layer architecture where the first layer is managed by Apple and has access to the public IP of the device while the second layer is operated by large CDNs and is responsible for decrypting the website names and completing the connection without access to the user's IP. All these solutions rely on intermediate nodes deployed to improve privacy whereas our solution is based on NATs that are often already used by the ISPs. The cost in terms of performance for the Tor network is not negligible. Indeed, the use of multiple relays and encryption layers can increase the latency significantly.

III. Design

The main contribution of this paper is the design and implementation of a client application with the corresponding server, allowing to detect multi-user NATs.

Our solution relies on an trusted server. The client application listens on all network interfaces to see when there is a change in the IP addresses. When such a change occurs, the device sends an HTTPS request to the server with a 64-bytes randomly chosen temporary ID. Each temporary ID has a limit, set randomly, in terms of the number of times they can be used so that they cannot be used to track the user. The user application keeps track of all the temporary IDs it has used. Upon reception of the request, the server sends an HTTPS reply containing the set of temporary IDs that have been previously sent by the same IP address. The IDs are only stored for several hours, on the server, as afterwards, the device that issued an ID has probably left the network. The client can compare this set of IDs with the ones it used. If there is at least one ID that was not used by this client, it must have been sent by another device using the same IP. It can then conclude that it is not alone behind the public IP address. The client stores the information on whether another device is detected and periodically sends aggregated reports to the server.

The reports need to be sent by the users as the server cannot know which temporary IDs are used by which devices and thus it cannot tell if there are multiple devices behind an IP address. Users can opt out of sending reports at any time, on the main page of the client application.

Tables I and II show how the collected data is stored. Each (temporary ID, IP) pair has an associated timestamp. Only IDs with a timestamp recent enough are sent back to the user. The results in the reports contain only a Boolean value indicating the presence of other devices. They also contain the device ID and the time of the measurement. There is no link between the device IDs and the temporary IDs used in the requests. The time could be used to correlate the data between the two tables if the correlation is made shortly after a temporary ID is sent. Correlation is only possible on our server. Further, as the timestamp of a temporary ID is replaced each time it is used on the same IP, the correlation between the two tables cannot be done on past data.

temporary ID	IP	timestamp
Temp A	10.10.10.10	2023-31-03 10:10:10
Temp B	10.10.10.10	2023-31-03 11:11:11
Temp C	20.20.20.20	2023-31-03 12:12:12

TABLE I: Stored identifiers format

deviceID	multiple users detected	time
A	false	2023-31-03 10:00:00
B	true	2023-31-03 10:00:00
C	false	2023-31-03 10:00:00

TABLE II: Stored results format

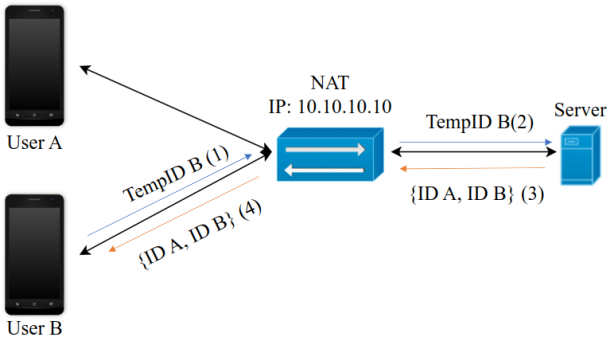

Fig. 1: Our proposed architecture

Figure 1 shows an example where after some time, the server has recorded a temporary ID from a User A through the NAT with address 10.10.10.10. Another user, B sends a request with another temporary ID through the same NAT. The server stores the new ID but cannot figure out if it was sent from the same user or another one as the IP is the same. He replies to the request with both temporary IDs. As user B knows that temporary ID B is the one that he used but doesn't have a record for temporary ID A, he concludes that there is another user behind the NAT.

The server is using the NestJS framework [17] and is implemented in about 200 lines of typescript. The client application is an Android application consisting of around 1200 lines of Java code. The results and temporary IDs are stored using a simple MySQL database [19] on the server and local files on the client application. All the traffic between the client and the server is encrypted with TLS [20].

IV. Evaluation

We conducted our study by distributing our application to 29 unique devices and collected data over a period of

4 weeks. During this time, we observed a total of 131 different IP addresses. On average, each device had access to approximately 4 IP addresses.

To determine if users shared IP addresses, we analysed the timing of their requests to the server and whether they detected the presence of other users. By considering that the result of each measurement holds until the next one, we estimated the time windows during which users shared an IP address. When considering all users, we found that, on average, they shared an IP address approximately 5% of the time.

It's important to note that this figure serves as a lower bound because our tool can only detect other users who are also using our application. It is possible that some users were behind a NAT with multiple other devices, but since they didn't interact with our server, their presence went unnoticed. Additionally, we explored the possibility of finding patterns in the time of day when IP address changes occurred for individual devices. Such patterns could indicate periods when users were likely behind a NAT, such as when using LTE during their journey from home to the workplace. However, we did not discover any significant patterns in our dataset.

V. Ethical Considerations

Our study follows the guidelines set by the Menlo Report [2]. We made sure that all the users involved were aware of the data that is collected. Our complete privacy policy is available online [4].

VI. Security Considerations

There are attacks possible against our system. Here, we present some of them and how they can be mitigated.

The biggest threat is when a tracker can detect that the server is queried and sends a request using the same IP in order to trick the application into detecting that there is another device. The request should therefore not be identifiable by an eavesdropper on the network. To achieve this, the IP of the server should not be uniquely identifiable which can be the case if we rely on a CDN to provide the service. The domain name and server name should not appear in cleartext when performing a request. The domain name should hence be retrieved using DNS over TLS [11] or DNS over HTTPS [10]. For the server's name, the TLS ClientHello should use the Encrypted Server Name Indication extension [21]. In the case a tracking entity still guesses that our server is being used, this entity could try to spoof the public IP address of a device to make the server record additional temporary identifiers and thus make a legitimate device believe that another device is present with the same IP. However, our proposal requires the establishment of TCP and TLS sessions before sending the device temporary IDs. This is hard to achieve by solely spoofing an IP. This attack often also requires the ability to hijack the route for the prefix of the victim.

VII. Discussion

Limitations As we don't store any link between the deviceID and the temporary identifiers, we cannot determine what are the IPs used by each user. Because the results are only linked to the deviceID and not to the IP address, we also cannot know which IP led to the detection of a NAT with multiple users.

Multi-devices users. Because users are not authenticated, it is not possible to distinguish devices belonging to different users. This means that in a home network with only one user but multiple devices, each device will detect that it is not the only one with the IP address, but the IP still maps to a unique user. Such users could retrieve all the temporary IDs on their devices to find out they are in fact alone behind their public IP address.

IPv6. While IPv6 allows to use privacy extensions [18], Saidi et al. showed that when using stateless address auto-configuration (SLAAC) [25], a single unsafe device is enough to track a whole client prefix by using the consistent MAC part of the addresses as an identifier. Detecting IPv6 NATs will remain relevant until privacy extensions become used by every device.

VIII. Conclusion

In this paper, we presented a service allowing users to determine if they have access to an interface that cannot be tracked by a classic IP tracker through the detection of other devices sharing the same IP. We implemented our solution and our results show that it can be used to detect NATs and identify if other users of the service are behind it. In the future, we will build up on this application to widen our user base and be able to perform larger-scale measurements. Once our measurements are done, we will extend the service to provide full anonymity to the users and inform them of their different interfaces such that they can select one behind a multi-user NAT.

References

[1] AdBlock. Adblock, 2018. https://getadblock.com/.

[2] Michael Bailey, David Dittrich, Erin Kenneally, and Doug Maughan. The menlo report. *IEEE Security & Privacy*, 10(2):71–75, 2012.

[3] Brave Privacy Team. Fingerprinting defenses 2.0, 2020.

[4] Aurélien Buchet. Multi-user nat detection privacy policy, 2023. https://aurelienbuchet.github.io/app_privacy_policy/.

[5] Roger Dingledine, Nick Mathewson, and Paul Syverson. *Tor: The Second-Generation Onion Router:*. Fort Belvoir, VA, Jan 2004.

[6] Arthur Edelstein. Protections against fingerprinting and cryptocurrency mining available in firefox nightly and beta, 2019. https://blog.mozilla.org/futurereleases/2019/04/09/protections-against-fingerprinting-and-cryptocurrency-mining-available-in-firefox-nightly-and-beta/.

[7] Electronic Frontier Foundation. Privacy badger, 2018. https://privacybadger.org/.

[8] Cliqz International GmbH. Ghostery, 2018. https://www.ghostery.com/.

[9] Eyeo GmbH. Adblock plus, 2018. https://adblockplus.org/.

[10] Paul E. Hoffman and Patrick McManus. DNS Queries over HTTPS (DoH). RFC 8484, October 2018.

[11] Zi Hu, Liang Zhu, John Heidemann, Allison Mankin, Duane Wessels, and Paul E. Hoffman. Specification for DNS over Transport Layer Security (TLS). RFC 7858, May 2016.

[12] Apple Inc. icloud private relay overview., 2021. https://www.apple.com/privacy/docs/iCloud_Private_Relay_Overview_Dec2021.PDF.

[13] Pierre Laperdrix, Nataliia Bielova, Benoit Baudry, and Gildas Avoine. Browser fingerprinting: A survey. *ACM Transactions on the Web (TWEB)*, 14(2):1–33, 2020.

[14] Ioana Livadariu, Karyn Benson, Ahmed Elmokashfi, Amogh Dhamdhere, and Alberto Dainotti. Inferring carrier-grade NAT deployment in the wild. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, page 2249–2257, Honolulu, HI, Apr 2018. IEEE.

[15] Jonathan R. Mayer and John C. Mitchell. Third-party web tracking: Policy and technology. In *2012 IEEE Symposium on Security and Privacy*, page 413–427, San Francisco, CA, USA, May 2012. IEEE.

[16] Vikas Mishra, Pierre Laperdrix, Antoine Vastel, Walter Rudametkin, Romain Rouvoy, and Martin Lopatka. Don't count me out: On the relevance of IP address in the tracking ecosystem. In *Proceedings of The Web Conference 2020*, page 808–815, Taipei Taiwan, Apr 2020. ACM.

[17] Kamil Mysliwiec. Nestjs - a progressive node.js framework, 2023. https://nestjs.com/.

[18] Thomas Narten, Richard Draves, and Suresh Krishnan. Privacy extensions for stateless address autoconfiguration in ipv6. Technical report, 2007.

[19] Oracle. Mysql, 2023.

[20] Eric Rescorla. The transport layer security (tls) protocol version 1.3. Technical report, 2018.

[21] Eric Rescorla, Kazuho Oku, Nick Sullivan, and Christopher A. Wood. TLS Encrypted Client Hello. Internet-Draft draft-ietf-tls-esni-16, Internet Engineering Task Force, April 2023. Work in Progress.

[22] Philipp Richter, Florian Wohlfart, Narseo Vallina-Rodriguez, Mark Allman, Randy Bush, Anja Feldmann, Christian Kreibich, Nicholas Weaver, and Vern Paxson. A multi-perspective analysis of carrier-grade nat deployment. In *Proceedings of the 2016 Internet Measurement Conference*, page 215–229, Santa Monica California USA, Nov 2016. ACM.

[23] Justin Schuh. Building a more private web, 2019. https://www.blog.google/products/chrome/building-a-more-private-web/.

[24] Ram Sundara Raman, Prerana Shenoy, Katharina Kohls, and Roya Ensafi. Censored planet: An internet-wide, longitudinal censorship observatory. In *proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, pages 49–66, 2020.

[25] Susan Thomson, Thomas Narten, and Tatuya Jinmei. Ipv6 stateless address autoconfiguration, rfc 2462. *Internet Engineering Task Force, Zeroconf Working Group*, 1998.

[26] John Wilander. Intelligent tracking prevention, 2017. https://webkit.org/blog/7675/intelligent-tracking-prevention/.