Network Working Group
Internet-Draft
Intended status: Informational
Expires: October 26, 2019

C. Wood
Apple Inc.

April 24, 2019

Linkable Identifiers
draft-wood-linkable-identifiers-01

Abstract

Rotating public identifiers is encouraged as best practice as a means of protecting endpoint privacy. For example, regular MAC address randomization helps mitigate device tracking across time and space. Other protocols beyond those in the link layer also have public identifiers or parameters that should rotate over time, in unison with coupled protocol identifiers, and perhaps with application level identifiers. This document surveys such privacy-related identifiers exposed by common Internet protocols at various layers in a network stack. It provides advice for rotating linked identifiers such that privacy violations do not occur from rotating one identifier while neglecting to rotate coupled identifiers.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

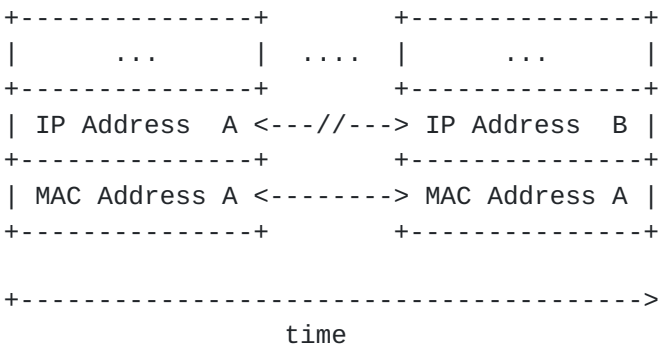
This Internet-Draft will expire on October 26, 2019.

Table of Contents

1. Introduction

[RFC6973] defines the correlation of information relevant to or associated with a specific user as a significant attack on privacy. Different layers of the network stack use identifiers to uniquely address hosts or information flows. To mitigate the privacy concern, many standards suggest randomizing or otherwise rotating such identifiers on a regular basis. For example, a MAC address may be used to link otherwise unrelated network packets to a single device. Rotating the MAC address prevents this association at the link layer. However, when multiple identifiers are simultaneously present on different layers of the stack, breaking the association at any individual layer might be insufficient to disassociate a host from their network traffic. Linkability can also occur across protocols and/or across layers. For example, TLS connections are commonly preceded by DNS queries for a particular endpoint (host name), e.g. example.com. Moreover, in the TLS handshake, this same host name is sent in cleartext in the Server Name Indication extension. Thus, observing either the DNS query or TLS SNI reveals information about the other. Similarly, while an IP address of a device may rotate, if web browser cookies do not, then, a website can track the various IP addresses of a given cookie over time.

Huitema et al. [I-D.ietf-dnssd-privacy] say, "it is important that the obfuscation of instance names is performed at the right time, and that the obfuscated names change in synchrony with other identifiers, such as MAC Addresses, IP Addresses or host names." Consider the following example where this advice is not followed, wherein an IP address is changed yet the MAC address is not.



A network adversary may trivially link these packets based on their common MAC address and continue to associate traffic with this particular host based on IP address B even if the MAC address eventually changes in the future. In this document, we outline simple rules that SHOULD be followed by protocol implementations to avoid such linkability. We then survey protocols developed inside the IETF and out, and identify their sticky identifiers. Results were obtained by analyzing protocol documentation and specifications, and also scanning packet traces captured from protocols in practice on common systems.

2. Sticky Protocol Identifiers

In this section, we survey existing protocols developed inside and out of the IETF, and identify sticky protocol identifiers for each. A sticky identifier is one that persists across logically grouped data exchanges between a client and server. This may include state-generating servers or, commonly, client algorithm, software configuration, or device-specific fields. We categorize surveyed protocols by the OSI layer at which they operate. Specifically, we focus on Link, Internet, Transport, Session, and Application layers. (Our taxonomy may not match traditional OSI models, though we consider it sufficiently representative.)

2.1. Internet and Link Layer

- Ethernet, 802.11, and Bluetooth: MAC addresses are fixed to specific devices. Unless frequently rotated, they are sticky identifiers. Simply rotating the MAC address may or may not be sufficient depending on other information sent at the protocol

layer with the a (rotated) MAC address. For example, in 802.11, frames have an incrementing sequence number and if the sequence number is not reset in unison with a MAC address change, the sequence number can be used to re-correlate randomized MAC addresses.

- IPv4 and IPv6: Static or infrequently rotating addresses are sticky identifiers when exposed on the network. Privacy Extensions for Stateless Address Autoconfiguration [RFC4941] enhance IPv6 client privacy by, e.g., issuing new IPv6 /64 prefixes every day. The 64-bit IID suffix remains random to deter linkability.

- IKEv2: Initiator Security Parameters Indexes (SPIs) are used as connection identifiers instead of IP addresses. They are required to rotate for each new SA.

2.2. Transport and Session Layer

- TCP [RFC0793]: TCP source ports may be sticky if reused across senders. For example, most operating systems allocate allocate ephemeral (short lived) ports to each new connection. Per IANA allocations, ephemeral ports range from 49152 to 65535 (2^15+2^14 to 2^16-1) [http://www.iana.org/assignments/port-numbers]. However, this does not prevent an application from re-using port across connections. Destination are also intentionally sticky, since they identify services offered by endpoints. Therefore, reusing a destination port does not lead to decreased linkability. Moreover, with TCP Fast Open (TFO) [RFC7413], servers give clients plaintext cookies that must be re-used when resuming a TCP+TFO connection. Clients do not modify these server cookies, which therefore means they can be tracked.

- MPTCP [RFC6824]: Connection tokens or IDs are explicitly used to link MPTCP subflows between IP address pairs. These tokens are only exposed during flow management operations, e.g., when creating new subflows. Normal data transfer uses TCP sequence numbers to bypass middlebox interference and an additional data sequence number (DSN) TCP option to allow receivers to deal with out-of-order subflow packet arrival. The union of packet DSNs across subflows should yield a contiguous packet number sequence.

- TLS [RFC5246] [RFC8446]: Prior to TLS 1.3, significant information is exposed during TLS handshakes, including: session identifiers (or re-used PSK identifiers in TLS 1.3), timestamps, random nonces, supported ciphersuites, certificates, and extensions. Many of these are common across all TLS clients - specifically, ciphersuites, nonces, and timestamps. However, others may persist

across active sessions, including: session identifiers (in TLS 1.2 and earlier versions) and re-used PSK identifiers (in TLS 1.3). Without rotation, these re-used identifers are sticky.

- DTLS [RFC6347]: Datagram TLS is a slightly modified variant of TLS aimed to run over datagram protocols such as UDP. In addition to identifiers exposed via TLS, DTLS adds cookie-based denial-of-service countermeasures. Servers issue stateless cookies to clients during a handshake, which must be replayed in cleartext by clients to prove ownership of its IP address. (This is similar to TFO cookies described above.) Additionally, DTLS is considering support of a static connection identifier (CID) [I-D.ietf-tls-dtls-connection-id], which permits client address mobility. CIDs are specifically designed to not change across addresses.

- QUIC [I-D.ietf-quic-transport]: QUIC is another secure transport protocol originally developed by Google and now being standardized by the IETF. IETF-QUIC [I-D.ietf-quic-transport] uses TLS 1.3 for its handshake. In addition to identifiers exposed by TLS 1.3, QUIC has its own connection identifier (CID) used to permit address mobility.

2.3. Application Layer:

- HTTP [RFC2616]: While HTTP is a stateless protocol, it enables applications to define state-keeping mechanisms in header fields. The fields might carry the state itself or tokens pointing to state kept at the endpoints. The Cookie header field [RFC6265] is de-facto the mechanism for web applications to uniquely identify their clients by generating a token and instructing the client to attach to any future requests. The ETag header field [RFC7232] enables applications to uniquely reference a resource which the client may cache. Applications may return unique reference tokens to distinct clients.

- DNS [RFC1035]: SRV records often contain human-readable information specific to particular devices, clients, or users. For example, printers may advertise its services with SRV records that contain a human-readable instance name. These are often not rotated as services change.

- NTP [RFC5905]: By default, mode 3 for NTP - client to server - sends several source-specific fields in the clear to NTP servers, including: timestamps, poll, and precision. These fields should be left empty or randomized as per [I-D.ietf-ntp-data-minimization]. Other fields that may link to

clients include: Stratum, Root Delay, Root Dispersion, Ref ID, Ref Timestamp, Origin Timestamp, and Receive Timestamp.

3. Identifier Scope and Threat Model

Not all packet identifiers are visible end-to-end in a client-server interaction. For example, MAC addresses are only visible to those with physical access to the medium - the local subnet for Ethernet and proximity for Wi-Fi; we will consider both of these "on-path" for the sake of this analysis. IP addresses are only visible between endpoints. (In systems such as Tor, source and destination addresses change at each circuit hop.) Thus, identifier linkability depends on the threat model under consideration. Off-path adversaries, e.g. those without physical access to the medium, are not considered a problem since they do not have access to packets in flight. On-path adversaries may exist at various locations relative to an endpoint (sender or receiver) on a path, e.g., in a local subnet, as an intermediate router or middlebox between two endpoints, or as a TLS terminating reverse proxy. In this document, we categorize these three types of adversaries as follows:

1. Local: An on-path adversary belonging to the same local subnet as an endpoint, e.g., a switch.

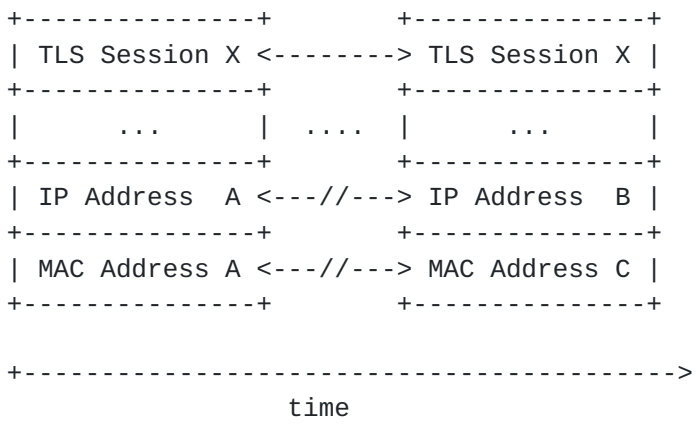
2. Intermediate: An on-path adversary that observes datagrams in flight but does not terminate a (TCP or TLS) connection, e.g., a middlebox or performance enhancing proxy (PEP).

3. Terminator: An on-path adversary that terminates a connection, e.g., a TLS- terminating reverse proxy. Note that there can be distinct terminators for individual layers of network stack. E.g., one for TLS and another for HTTP.

The scope of an identifier includes are all other protocols and layers observable by the same adversary.

4. Limiting Linkable Identifiers

The introductory example illustrating packet linkability using MAC addresses is one of many possible ways in which an attacker may link packets. As another hypothetical example, assume that IP address and MAC addresses were properly rotated, whereas TLS session identifiers were reused over time, as shown below.



Despite rotating all protocol identifiers beneath TLS, a static session identifier makes packet linkability trivial. Thus, a strict, yet safe rule for removing packet linkability is to rotate all linked identifiers in unison. Unfortunately, this strategy is problematic in practice. It would imply terminating active connections whenever an identifier changes (otherwise, linkability remains trivial). For example, if MAC addresses are rotated on a regular basis, e.g., every 15 minutes, then connection lifetimes would be limited to this window.

A more sensible policy would be to restrict identifier rotation to layers which are exposed to the same adversary. For example, origin MAC addresses may not be visible to the destination. In this case, rotating IP addresses and TLS session identifiers is not required to prevent packet linkability by an adversary who does not see the origin MAC address. A realistic threat model is one in which IP- to TLS-layer information is exposed to the same on-path adversary. Identifiers beneath IP are visible to local adversaries, which may not be an issue, and those above TLS are visible to authenticated peers.

4.1. Time and Path Linkability

There are multiple dimensions along which identifiers may be linked: (1) time, as identifiers are used and re-used by senders, and (2) space, as identifiers are duplicated across multiple disjoint network paths, possibly by different protocols. We refer to these dimensions as time and path linkability, respectively.

Time linkability is arguably simpler to mitigate, since new connections over time may opt to use new identifiers. For example, instead of resuming a TLS session with an existing session ID, a client may initiate a fresh handshake. As a simple rule, if an identifier in the same scope changes, endpoints SHOULD use fresh identifiers for all other protocols in that scope. This means that,

for identifiers visible to intermediate adversaries, new TLS sessions SHOULD be initiated from an endpoint with a fresh IP address and TCP source port. Note that clients behind NATs may not need to generate a fresh IP address, as they enjoy some measure of anonymity by design. If local adversaries were considered part of the threat model, then a fresh MAC address may also be needed.

In contrast, path linkability is more difficult to achieve, as it requires using fresh identifiers for each protocol field. This may not always be technically feasible. For example, DNS query names are also intentionally used as the TLS SNI. Moreover, protocols such as QUIC explicitly try to enable path linkability via connection-level identifiers (CIDs) to support multihoming or mobile endpoints. This makes path linkability impossible to mitigate. However, as multiple, disjoint paths may be operated by different entities (e.g., ISPs), collusion may be less common.

5. Timing Considerations

Advice in this document SHOULD NOT be interpreted as guarantees for preventing linkability. Rather, it aims to increase linkability complexity. It is difficult to prevent path-linkability without modifying protocols above the layer at which identifiers rotate. For example, assuming MPTCP subflows were unlinkable across paths, shared transport state controlling the rate of data transmission may be sufficient to link these flows.

6. IANA Considerations

This document has no request to IANA.

7. Security Considerations

This document does not introduce any new security protocol.

8. Privacy Considerations

This document describes considerations and suggestions for improving privacy in the context of many IETF protocols. It does not introduce any new features or protocol behavior that would adversely impact privacy.

9. Acknowledgments

The authors thank Martin Thompson and Brian Trammell for comments on earlier versions of this document.

10. Normative References

[I-D.ietf-dnssd-privacy]
Huitema, C. and D. Kaiser, "Privacy Extensions for DNS-SD", draft-ietf-dnssd-privacy-05 (work in progress), October 2018.

[I-D.ietf-ntp-data-minimization]
Franke, D. and A. Malhotra, "NTP Client Data Minimization", draft-ietf-ntp-data-minimization-04 (work in progress), March 2019.

[I-D.ietf-quic-transport]
Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", draft-ietf-quic-transport-20 (work in progress), April 2019.

[I-D.ietf-tls-dtls-connection-id]
Rescorla, E., Tschofenig, H., and T. Fossati, "Connection Identifiers for DTLS 1.2", draft-ietf-tls-dtls-connection-id-04 (work in progress), March 2019.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <https://www.rfc-editor.org/info/rfc793>.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[RFC2508] Casner, S. and V. Jacobson, "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links", RFC 2508, DOI 10.17487/RFC2508, February 1999, <https://www.rfc-editor.org/info/rfc2508>.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, DOI 10.17487/RFC2616, June 1999, <https://www.rfc-editor.org/info/rfc2616>.

[RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, DOI 10.17487/RFC4941, September 2007, <https://www.rfc-editor.org/info/rfc4941>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <https://www.rfc-editor.org/info/rfc5246>.

[RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <https://www.rfc-editor.org/info/rfc5905>.

[RFC6265] Barth, A., "HTTP State Management Mechanism", RFC 6265, DOI 10.17487/RFC6265, April 2011, <https://www.rfc-editor.org/info/rfc6265>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <https://www.rfc-editor.org/info/rfc6347>.

[RFC6824] Ford, A., Raiciu, C., Handley, M., and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses", RFC 6824, DOI 10.17487/RFC6824, January 2013, <https://www.rfc-editor.org/info/rfc6824>.

[RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <https://www.rfc-editor.org/info/rfc6973>.

[RFC7232] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests", RFC 7232, DOI 10.17487/RFC7232, June 2014, <https://www.rfc-editor.org/info/rfc7232>.

[RFC7413] Cheng, Y., Chu, J., Radhakrishnan, S., and A. Jain, "TCP Fast Open", RFC 7413, DOI 10.17487/RFC7413, December 2014, <https://www.rfc-editor.org/info/rfc7413>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

Author's Address

Christopher A. Wood
Apple Inc.
One Apple Park Way
Cupertino, California 95014
United States of America

Email: cawood@apple.com