How can network traffic lie?

Milan Cermak and Petr Velan

Institute of Computer Science, Masaryk University, Brno, Czech Republic
cermak@ics.muni.cz, velan@ics.muni.cz

Summary

Network traffic is considered to be a trusted data source for digital forensics and incident response, which is typically summarized by the phrase: "The network does not lie." However, this phrase is accurate only if we fully control the packet trace from capture to analysis. In other cases (e.g., external file source, improper handling, or access by unauthorized personnel), it must be considered that the trace file may have been modified, and some data or network connections may have been removed, added, or changed. These modifications are likely to be small but could have a significant impact on the analysis outcome.

The first part of the poster discusses manipulation methods and presents approaches to their identification based on the context of different protocols and connection types. We primarily focus on unwanted changes (i.e., we do not consider, for example, anonymization) whose presence is unknown to the analyst in advance. For each manipulation method, we list possible indicators that we have identified based on the analysis of individual protocols and our personal experience. The second part of the poster presents tools that can be used to manipulate the packet trace. We aim to show that general manipulation may be performed in a simple way by using these tools. However, if perfect data consistency is to be maintained, the manipulation requires significant expertise and attention to detail. In most cases, we may assume that the manipulation will not be carried out perfectly and preserve some indicators. Awareness of these indicators is crucial to ensure that unwanted packet trace manipulation does not remain hidden.

The PDF version of this poster is available at:
https://is.muni.cz/publication/2267297/2023-DFRWS-EU-how-can-network-traffic-lie-poster.pdf

Reasons and indicators of packet trace modification

In general, we may divide unwanted reasons for packet trace manipulation into intended and unintended. Unintended reasons are caused by improper data handling (e.g., splitting and joining different trace files), and their signs are typically more significant. In contrast, intended reasons are typically less noticeable and aim to influence analysis and the whole investigation. However, network traffic is quite complex and full of various dependencies and interrelated actions (e.g., domain translation before web connection). It is these relations that allow us to recognize possible manipulation.

Evidence hiding

An attempt to erase traces of selected network activity or modify network traffic so that it does not contain key characteristics.

False evidence

An attempt to insert additional network connections or edit existing ones to provide inaccurate leads.

Remove

Erasing one or several network connections or deleting packet field data.

Several connections	All connections between defined hosts
All connections from/to source host	All connection from/to destination host

Add

Injecting one or several network connections.

Several connections between existing hosts	Connections from/to new destination host
Connections from/to new source host	

Edit

Changing packet fields of captured network connections.

Addresses	Transport, internet, and link layer packet fields
Application data	Timestamps

Improper data handling

Performing operations such as splitting, merging, or even minor modifications (e.g., timestamp normalization) may cause an unintentional error that can negatively affect the packet trace.

Remove indicators

DNS: Translation of unvisited domain(s)
The trace contains a translation of the domain (corresponding to removed connections), whereas no related network connection is initiated after the translation.

FTP: Only data transfer or control connection
Only one type of connection (control at port 21 and data transfer at port 20) is deleted, while the other remains in the trace file. Similar for other protocols such as VoIP.

HTTP(S): Requests for external resources
Web pages typically consist of many external resources (e.g., javascript, CSS) whose request can be preserved in the trace file.

TLS: Certificate transparency check
For lesser-known domains, we may see traffic related to a certificate transparency check for a certificate not present in the previous traffic.

TCP/UDP: Incomplete or broken connections
If the entire connection is not correctly removed (e.g., due to time-based removal), packets from the connection start or end (as well as fragments) may be preserved.

IPv4/IPv6: Sudden drop in traffic for a given address
If connections are deleted for a specific time window, we can see a temporary drop in traffic for a given source address.

IPv4/IPv6: Only IPv4 or IPv6 connections in a dual-stack network
Although a host communicates primarily through IPv4, in the case of the dual-stack, it also creates IPv6 connections (locally or outside the network) and vice versa.

ICMP: Host unreachable message with destination host address
If the destination host is unavailable at some time, the client may receive an ICMP "Host unreachable" message from the router containing the IP address of the destination host.

DHCP: Offer message for a host without connections
All DHCP communication is carried in a link layer, which can be preserved as long as IP connections are deleted.

ARP: Translation of a local host address
If connections of two hosts within the local network are deleted, the ARP traffic related to IP to MAC address translation may be preserved.

ARP: Responses from a host without connections
If all hosts' IP connections are removed, responses to ARP requests from a router or other local network hosts may be preserved.

Add indicators

DNS: Missing translation of visited domain(s)
The packet trace misses a translation of the domain related to the destination host from added connections.

HTTP(S): Missing requests for external resources
Web pages typically consist of many external resources (e.g., javascript, CSS) whose request should be preserved in the trace file.

HTTP: Different user agent
Related network connections should originate from the same application that the user agent identifies.

HTTP: Unvisited host from referer
When a web link is visited, the HTTP header contains information about the name of the web page from which the link was visited.

TLS: Different ciphersuite list
Both the client and the server should preserve the offered cipher suite lists in all their connections in the packet trace.

TCP: Various window sizes for the same IP address
The window size should be unchanged for the entire packet trace or changed with a longer-term effect.

TCP: Various maximum segment sizes for the same IP address
The maximum segment size value should be unchanged for the entire packet trace or changed with a longer-term effect.

IPv4/IPv6: Various TTLs for the same IP address
The TTL value should be unchanged for the entire packet trace or changed with a longer-term effect.

IPv4/IPv6: Inconsistent interpacket gaps
All connections between two hosts in a packet trace should preserve similar interpacket gaps with minimal deviation.

IPv4/IPv6: Different capture snaplen
If a capture with a different snaplen is added, it may contain larger or truncated packets whose length is inconsistent with the original packet trace.

ARP: Missing translation of a local host address
If the communication occurs in the local network, the corresponding ARP traffic should be visible (if other ARP traffic is captured).

ARP: Missing responses from a host
The host in the local network should continuously respond to ARP requests from other hosts in the network.

ETHERNET: Different addresses for the same IP address
Although MAC addresses may change over time, it is not common for multiple MAC addresses to be used simultaneously or reverted quickly.

Edit indicators

DNS: Missing translation of visited domain(s)
The packet trace misses a translation of the domain related to the destination host address.

DNS: Translation of unvisited domain(s)
The trace contains a translation of the domain, whereas no related network connection is initiated after the translation.

DNS/TLS/HTTP: Inconsistent hostname in related connections
Changing the hostname within only one connection may cause that other related connections preserve the original hostname (e.g., in domain translation, SNI in TLS).

TLS: Invalid certificate
Changing TLS certificate values (e.g., subject or validity) causes that the certificate is no longer valid.

NTP/HTTP: Date field values inconsistency
NTP and HTTP protocols may contain time information that should correspond approximately to the observed connection time.

TCP: Inconsistent protocol behaviour
The characteristics of the captured traffic for a given connection will not match the values changed in the protocol fields (e.g., windows size, defined flags).

IPv4/IPv6/TCP/UDP: Invalid checksum
Any change in packet fields will cause invalid checksum, which needs to be recalculated when changed.

IPv4/IPv6: Sudden drop in traffic for a given address
If addresses are changed for a specific time window, we can see a temporary drop in traffic for a given source address.

IPv4/IPv6: Various protocol field values for the same IP address
Protocol field values should be unchanged for the entire packet trace or changed with a longer-term effect.

IPv4/IPv6: Inconsistent protocol behaviour
The characteristics of the captured traffic for a given connection will not match the values changed in the protocol fields (e.g., fragmenting).

IPv4/IPv6: Invalid data at packet end
Depending on the manipulation method, keeping the original packet length may be necessary, leading to additional zeros or other random characters at the packet end.

IPv4/IPv6: Incorrect packet length
The change in the data can reduce or increase the application layer, and the resulting length may not match the length defined in the IPv4/IPv6 header.

ETHERNET/IPv4/IPv6: Preceeding connections
Timestamp changing for a connection may lead to a state where the connection precedes another one that is typically before such connection (e.g., DNS before HTTP).

ICMP: Host unreachable message with destination host address
If the destination host is unavailable at some time, the client may receive an ICMP "Host unreachable" message from the router containing an IP address of the destination host.

DHCP: Offer message for a host without connections
All DHCP communication is carried in a link layer, which can be preserved if only IP connections are deleted.

ARP: Translation of a local host address
If the addresses of hosts in the local network are changed, the original ARP traffic related to the translation of IP to MAC address may be preserved.

ARP: Responses from a host without connections
If host address is changed in all connections, responses to ARP requests from a router or other local network hosts with the original address may be preserved.

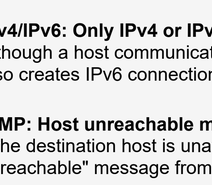
Tools for Packet Trace Manipulation

Packet traces can be manipulated either manually (on a byte level) or with more complex tools that allow us to change individual packet fields and maintain data integrity efficiently. This section briefly introduces the available tools and approaches that can be used for this purpose. We focus only on manipulation, thus tools aimed at anonymization are not listed here. We also do not include packet crafting tools (typically used for testing network services). We view them as a source of external data that can be added to the capture (e.g., using the "mergecap" tool).

GUI editors

Wireshark (version <= 1.12.*)

The packet editing functionality is available only in the legacy version of Wireshark with GTK-based GUI (the latest version from 2015). It enables editing packet fields, including application data. However, changes cannot be made in bulk. Also checksums are not automatically recalculated.
https://www.wireshark.org/

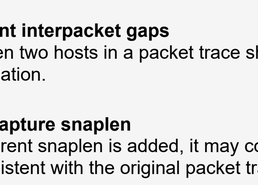


Colasoft Packet Builder

Freeware tool from 2016 providing a well-arranged GUI for packet fields editing, including application data, with automated checksums recalculation. In addition, it enables the insertion of additional network traffic. However, bulk editing is not available.
https://www.colasoft.com/download/products/download_packet_builder.php

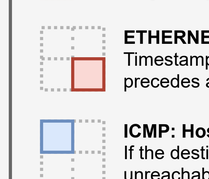
Netdude

An open-source tool without active development (last update 2010) for inspection, analysis, and manipulation of packet trace files. Nevertheless, the application data can only be edited as a whole, using HEX or text editor, without breaking it down into individual fields. In addition, bulk editing and checksums recalculation is not available.
https://netdude.sourceforge.net/



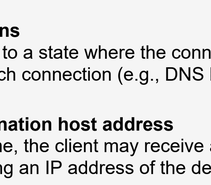
WireEdit

An actively developed paid editor of packet traces, allowing simple editing of all packet layers and automated maintenance of binary data integrity. Compared to other tools, it also provides functionality for bulk editing.
https://omnipacket.com/wireedit



Ostinato

An actively developed paid tool primarily designed for network testing that also provides editing capabilities. Editing is provided as "Find & Replace" on individual packet fields or application data (as a whole) while recalculating checksums (if the option is selected).
https://ostinato.org/



Command line tools

Editcap

The tool is part of the toolkit installed together with Wireshark. Regarding packet trace manipulation, it primarily allows packet capture splitting and time adjustment.
https://www.wireshark.org/docs/man-pages/editcap.html

Mergecap

The tool is part of the toolkit installed together with Wireshark. It serves primarily to merge packet traces (add/inject) while ensuring correct packet ordering by timestamp.
https://www.wireshark.org/docs/man-pages/mergecap.html

Tcprewrite (and tcpreplay-edit)

A tool for complex manipulation of packet traces that allows (bulk) editing of packet fields (addresses, TTL, ToS, and others) with automated checksums recalculation. However, it does not provide functionality for application data editing.
https://tcpreplay.appneta.com/

Bittwiste

An open-source tool without active development (last update 2012). Compared to tcprewrite, it provides a broader range of options for packet fields manipulation. However, it also does not provide functionality for application data editing.
https://bittwist.sourceforge.net/

Other approaches and tools

Programming

Using appropriate libraries (e.g., "Scapy" for Pyhon, "PcapPlusPlus" for C++), it is possible to write programs that modify a given trace and preserve the binary integrity of the data. The disadvantages are higher time and user knowledge requirements.
https://scapy.net/ or
https://pcapplusplus.github.io/

HEX editing

The packet trace can be edited directly at the byte level using any HEX editor (or command line tools like "sed"). The advantage is the possibility of bulk editing. However, only existing values can be replaced (i.e., the length of the changed fields must be preserved), and checksums are not automatically recalculated.

ID2T

An open-source tool for dataset creation using the insertion of synthetic network traffic. It supports the automatic adaptation of the inserted traffic according to the statistical properties of the original packet trace (e.g., timing, addresses). However, no other modification options are provided.
https://github.com/tklab-tud/ID2T

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research, Development and Education

MINISTRY OF EDUCATION, YOUTH AND SPORTS

This research was supported by ERDF "CyberSecurity, CyberCrime and Critical Information Infrastructures Center of Excellence" (No. CZ.02.1.01/0.0/0.0/16_019/0000822).