



pCapitan

User guide - network forensics in plain language.

Version 1.2.1 · October 2026 · GizmoGoose · gizmogoose.com



Contents

Welcome to pCapitan

- Installing
- Your first capture
- Finding your way around

Understanding a capture

- Overview
- Network map
- World map
- Traffic flow
- Timeline
- Protocols

Plain answers: Ask, diaries and replay

- Ask pCapitan
- Device diary
- Narrated replay
- How this works without AI
- Who is who
- Proxies and firewalls in the path
- Unusual captures

Digging into the details

- Conversations
- Packets
- Devices
- Destinations
- DNS
- Web (HTTP)
- Encryption (TLS and QUIC)

Filters

- Quick filters (no syntax needed)
- The filter language
- The visual filter builder
- The filter library
- Your own filters

Security and forensics

- Findings
- Forensics

IDS signatures (Snort / Suricata rules)

Kerberos / Active Directory

Wi-Fi (802.11)

Event timeline

Performance

Incident response

What happened?

Did the attack work?

Vulnerabilities

Respond & notes

The incident report

Decrypting traffic: the Decryption Vault

Where to find it

What the vault is

What can go in

How keys are matched to the capture

RSA key exchange and forward secrecy

The Encryption column

Wi-Fi and IPsec: a new decrypted capture

Following decrypted streams

Good practice

Why is it slow? Blame, compare, calls and logins

Who made it slow?

Compare captures

VoIP calls

Logins & routing

Which page for which problem

Keeping detection current: Updates & feeds

What ships with pCapitan and what you download

Threat feeds

CISA Known Exploited Vulnerabilities

 YARA rules

 Network owners (ASN)

Live capture

Live alerts

 Remote capture over SSH

Reports and exports

 Your branding on reports

♥♥ Vendor-safe copy

 Batch audit

Licence, trial and activation

[The 14-day free trial](#)

[Buying pCapitan](#)

[Activating](#)

[Moving to another computer](#)

[The licence check](#)

[Updates](#)

Troubleshooting and FAQ

[Keyboard shortcuts](#)

[Privacy and responsible use](#)

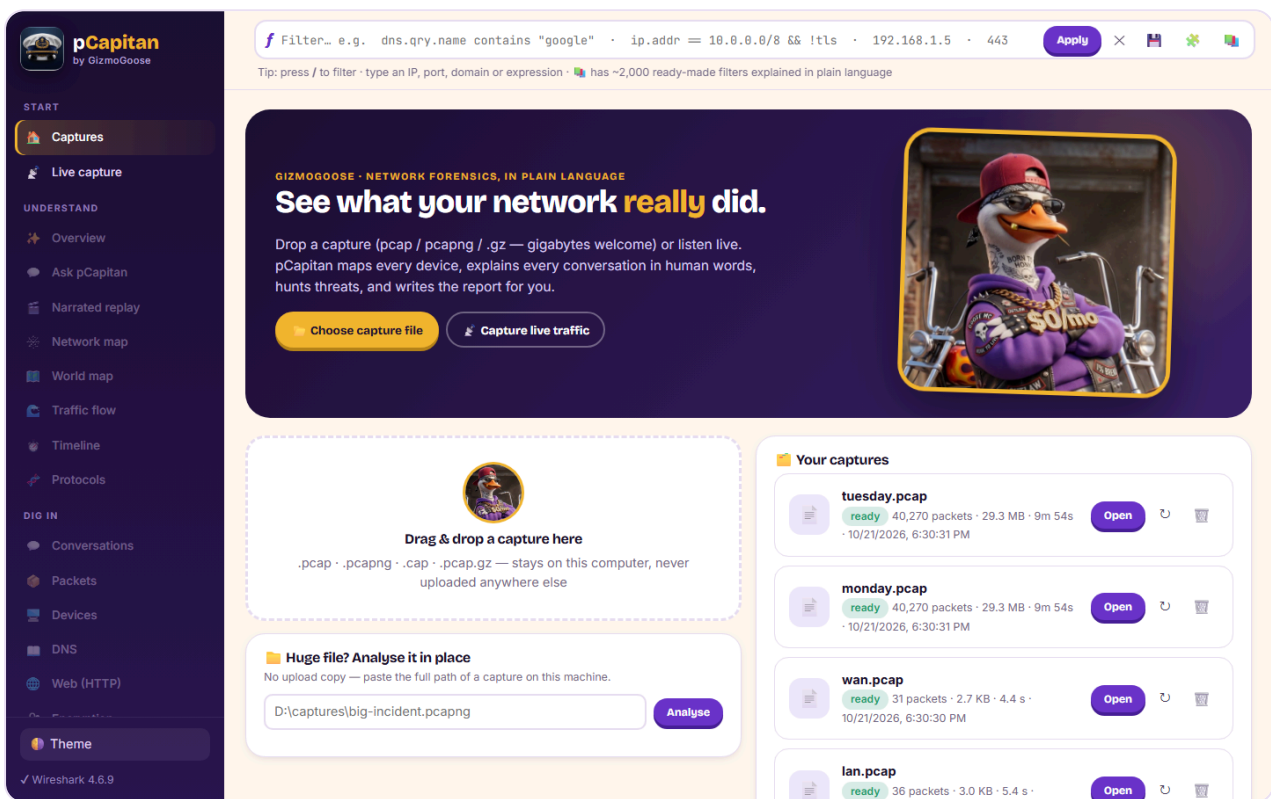
[Support](#)

Welcome to pCapitan

pCapitan (by GizmoGoose) opens network captures (`.pcap` , `.pcapng` , `.cap` , `.pcap.gz`) or listens to live traffic. It turns them into pictures and plain language:

- who was on the network;
- what each device did;
- which connections were risky;
- what you should do about it.

Everything runs **on your own computer**. Captures are never uploaded anywhere.



Installing

Download pCapitan from gizmogoose.com/tools/pcapitan. Every download starts with the 14-day free trial of every feature; a licence keeps it working afterwards.

System	What to download	How
Windows 10 / 11	pCapitan-<version>-Setup.exe (or the .msi)	Double-click and follow the wizard. Start pCapitan from the Start menu.
macOS 11+ (when available)	pCapitan-<version>.dmg	Open the DMG and drag pCapitan to <i>Applications</i> .
Ubuntu / Debian	pcapitan_<version>_amd64.deb	<code>sudo apt install . /pcapitan_<version>_amd64.deb</code> , then start <i>pCapitan</i> from the menu.
Any Linux	pCapitan-<version>-x86_64.AppImage	<code>chmod +x pCapitan-*.AppImage</code> and run it.

Optional: Wireshark (recommended)

pCapitan analyses captures entirely by itself. If **Wireshark** is installed (and **Npcap** on Windows), it adds:

- **Live capture** from your network adapters
- The full field-by-field **dissection tree** for any packet
- **File carving** (extracting files sent over HTTP, SMB, FTP and more)
- **Kerberos / Active Directory** analysis
- **TLS decryption** with a key-log file

Use **Wireshark 4.2 or later**; the latest version from [wireshark.org](https://www.wireshark.org) is recommended. Older versions (for example the 3.x that some Linux distributions still ship) may fail to start a live capture. On Windows, tick **Install Npcap** in the Wireshark installer.

The bottom-left corner of pCapitan shows whether Wireshark was found (✓ Wireshark 4.x). If your Wireshark is older than 4.2,  **Live capture** says so at the top of the page.

Your first capture

1. On **Captures** (the home screen), click **Choose capture file** or **drag a file onto the drop zone**.
2. pCapitan reads it and shows a progress bar. A typical file finishes in seconds. A multi-gigabyte file takes a few minutes, because every packet is decoded, tracked into a conversation and indexed for instant filtering.
3. When it is ready, the **Overview** opens automatically.

Huge files: use **Huge file? Analyse it in place** and paste the file's full path. pCapitan then reads the file where it is, instead of making an upload copy.

You can also double-click any `.pcap` or `.pcapng` file in Explorer or Finder to open it straight in pCapitan.

Where things are kept

Your captures, their analysis and your downloaded threat feeds are stored in a per-user folder:

- **Windows:** %LOCALAPPDATA%\pCapitan
- **macOS:** ~/Library/Application Support/pCapitan
- **Linux:** ~/.local/share/pCapitan

Delete a capture from the home screen with . Use  to analyse it again, for example after loading new threat feeds.

Finding your way around

The sidebar groups every screen by what you want to do:

Group	Screens
Start	Captures, Live capture
Understand	Overview, Network map, World map, Traffic flow, Timeline, Protocols
Dig in	Conversations, Packets, Devices, DNS, Web (HTTP), Encryption
Security	What happened?, Findings, Vulnerabilities, Respond & notes, Forensics, IDS signatures, Kerberos / AD, Wi-Fi, Event timeline, Performance
Tools	Filter library, Updates & feeds, Report & export, Help & guide

The **filter bar** at the top applies to *every* screen. Type `192.168.1.5`, `dns` or `tls.sni contains "google"`, press **Enter**, and the map, charts, tables and report all show only that traffic. Press **/** to jump to the filter bar from anywhere.

Words with a dotted underline (like *TTL* or *SNI*) have a plain-language explanation. Hover over them to read it.

Understanding a capture

Overview

The **Overview** is the one-page answer to "what happened on this network?".

The screenshot shows the pCapitan interface for a file named 'sample.pcap'. The left sidebar contains navigation options: START (Captures, Live capture), UNDERSTAND (Overview, Ask pCapitan, Narrated replay, Network map, World map, Traffic flow, Timeline, Protocols), and DIG IN (Conversations, Packets, Devices, DNS, Web (HTTP), Theme). The main area is titled 'OVERVIEW sample.pcap' and includes a filter bar at the top. Below the filter bar, a 'Risk: High' banner indicates 31 findings (6 critical, 12 high, 9 medium, 4 low). A 'What happened?' section provides a summary of the capture, mentioning data smuggling via DNS and various login attempts. Below this, a row of key statistics is displayed: 40,270 Packets, 28.7 MB Data, 9m 54s Duration, 9 / 16 Devices (local / Internet), 492 Conversations, and 6 Credentials exposed. The 'The story of this capture' section provides a detailed narrative of the capture, including the number of active devices, conversations, and the busiest device. The 'Top findings' section lists several critical issues, such as ARP spoofing, FTP credentials observed, HTTP Basic credentials observed, HTTP form credentials observed, SNMP v2c credentials observed, and PsExec remote execution.

- **Risk banner:** *High, Medium or Low*, based on the security findings. Click **Review** → to see them.
- **Key numbers:** packets, data volume, duration, local and internet devices, conversations, and exposed credentials.
- **The story of this capture:** a written summary in plain language, covering:
 - who was busiest;
 - what kind of traffic it was and how much was encrypted;
 - how much went to the internet;
 - network health;
 - the most important findings.

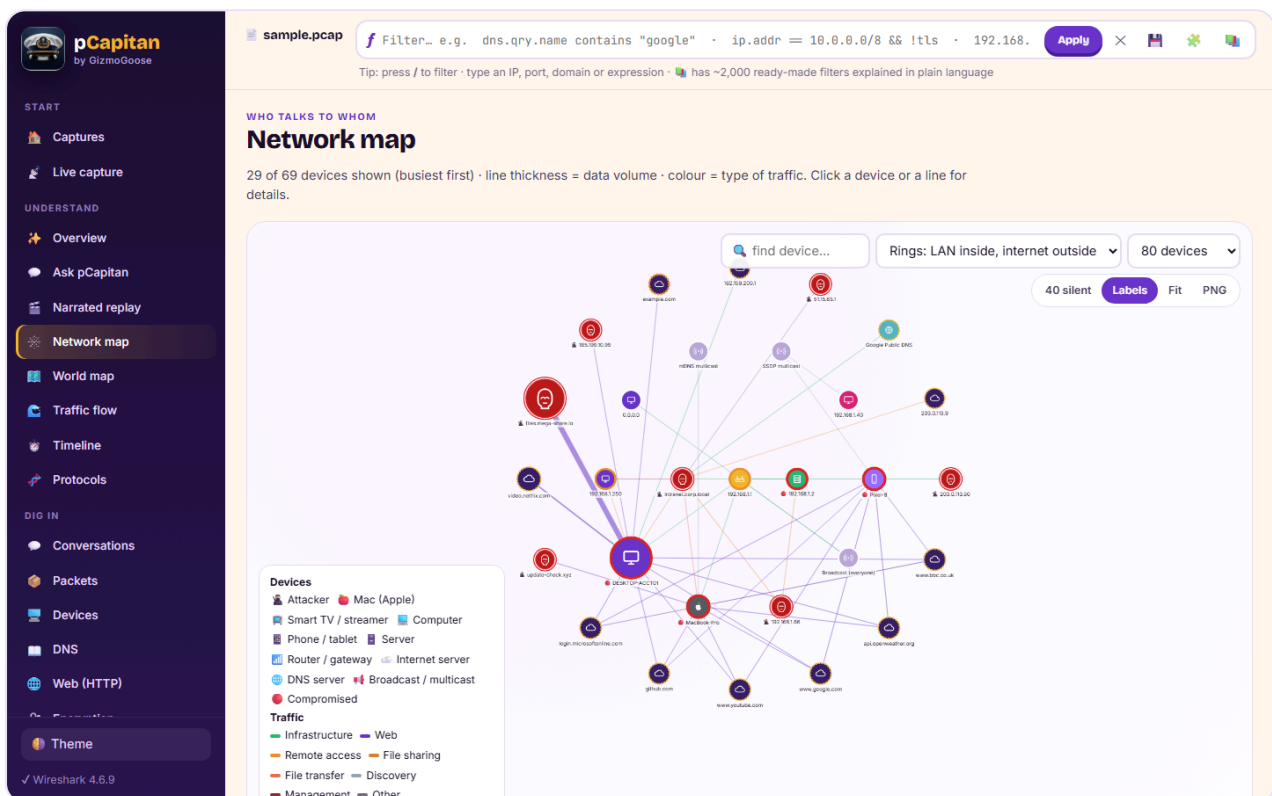
Click **Explain the current selection in depth** for a deeper narrative of whatever is currently filtered. - **Charts:** activity over time, traffic types, busiest devices, traffic direction, top encrypted sites, top DNS look-ups and connection health. **Click any slice or bar** to filter to it.

Network map

Every device is drawn as an icon:

-  computer
-  phone
-  server
-  router
-  internet server
-  DNS server
-  printer
-  camera
-  domain controller

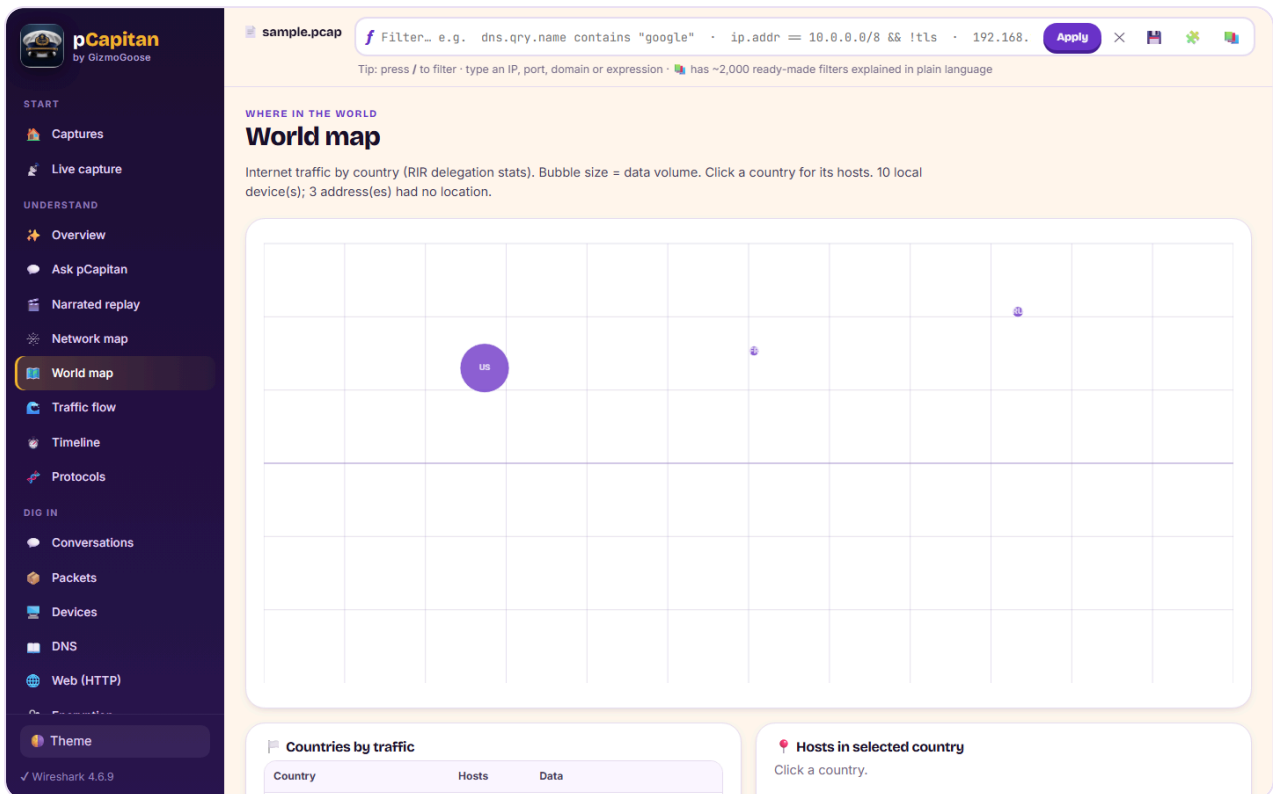
Lines show who talked to whom. **Thicker means more data**, and **colour shows the kind of traffic**. Internet hosts have a dashed gold border.



- **Hover** a device to highlight its neighbours. **Click** it for its full profile.
- **Click a line** to see what the two devices exchanged, filter to that pair, or hide it.
- **Layout:** *Rings* puts the LAN inside and the internet outside. *Organic*, *Hierarchy from gateway*, *Concentric*, *Circle* and *Grid* are also available.
- **Device count:** choose how many of the busiest devices to show. **"silent"** reveals addresses that were contacted but never answered, such as hosts probed by a scan.
- **PNG** saves the map as an image.

World map

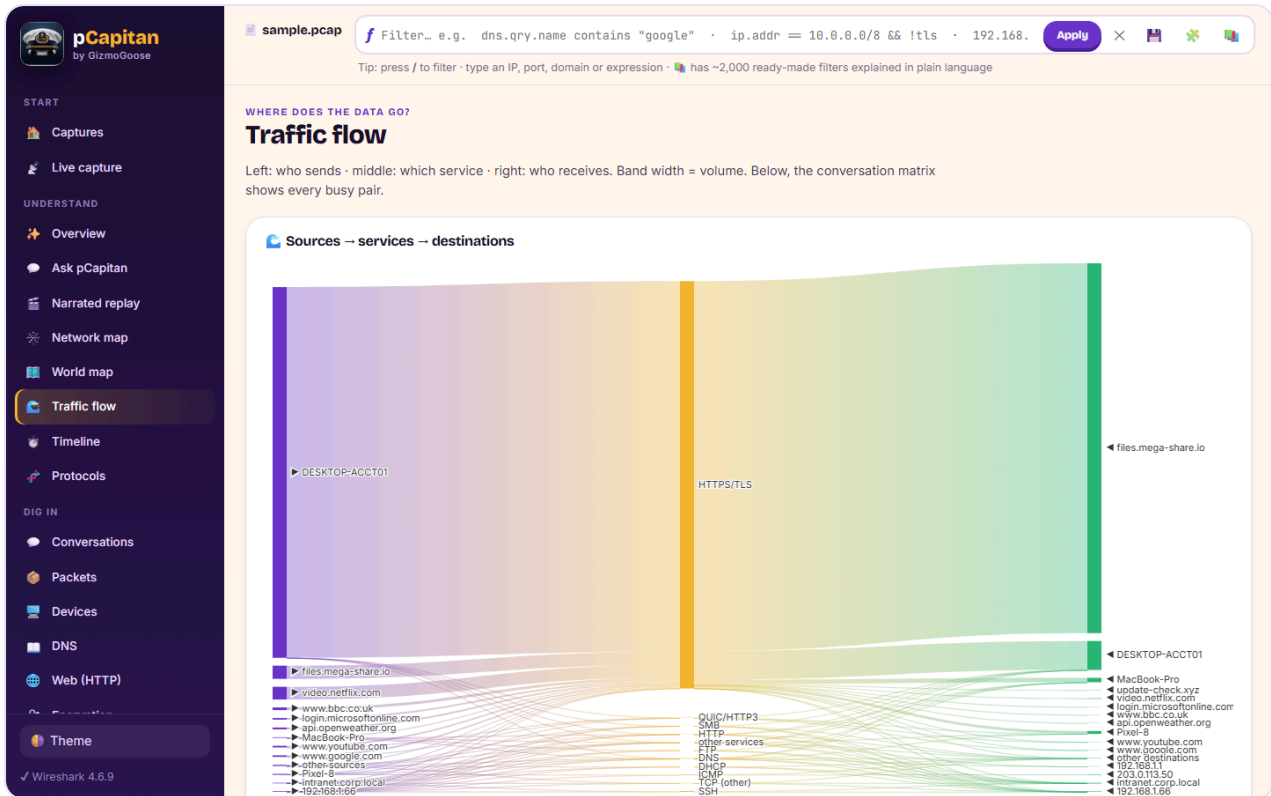
Shows which **countries** your internet traffic reached. Bubble size equals data volume. Click a country, or its row in the table, to list its hosts. **filter** limits the whole app to that country.



Country data comes from a local database built from the Regional Internet Registries, so no lookups are sent anywhere. Manage it in **Updates & feeds** → **Location database**.

Traffic flow

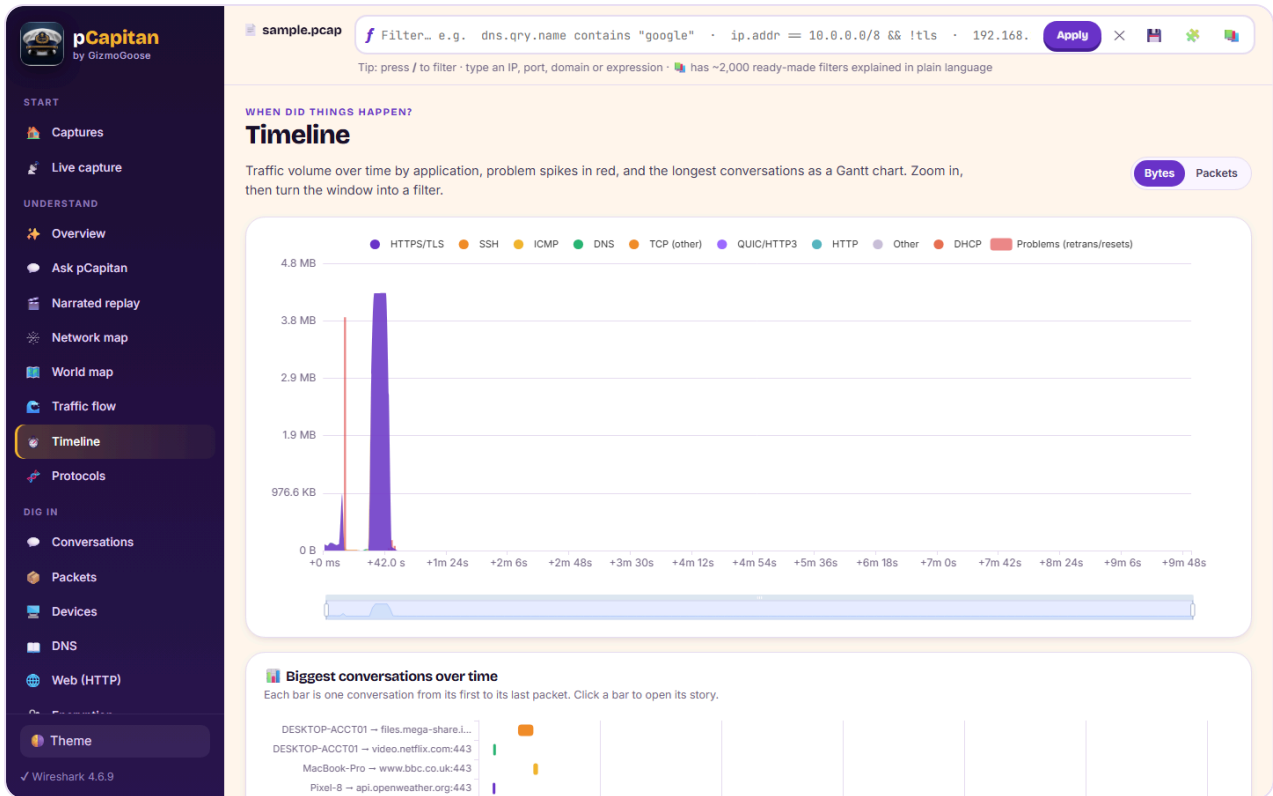
A **Sankey diagram**: senders on the left, services in the middle, receivers on the right. Band width equals volume. Underneath, the **conversation matrix** heat-maps every busy pair. Click a cell to filter that pair.



Timeline

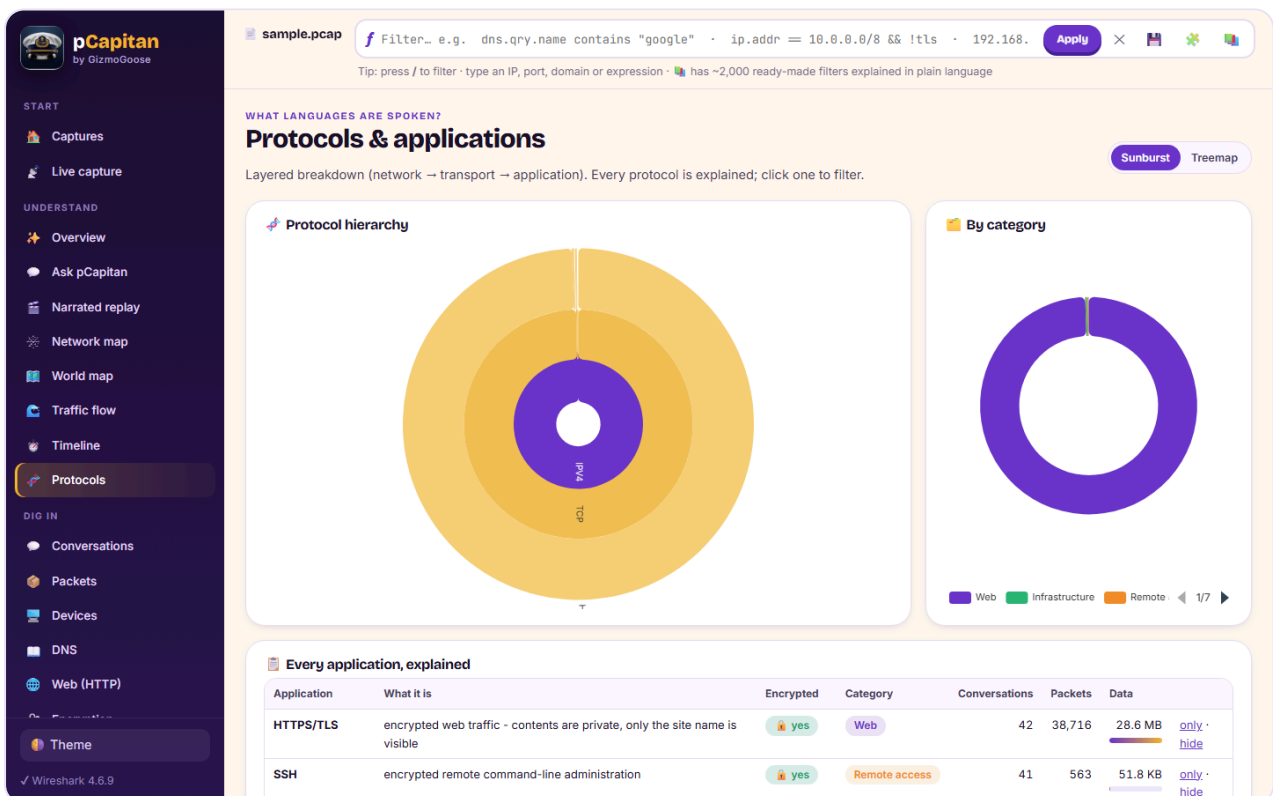
Traffic volume over time, stacked by application. Problem spikes (retransmissions and resets) appear as red bars.

- **Drag** the slider or zoom with the wheel. Then click **Filter to ...** to turn the time window into a filter.
- **Biggest conversations** is a Gantt chart. Click a bar to open that conversation's story.
- **Key events** plots DNS look-ups, sessions, credentials and alerts in time order.



Protocols

The layered breakdown: network, then transport, then application. View it as a **sunburst** or a **treemap**. The table lists **every application with a plain explanation** of what it is and whether it is encrypted. Use **only** or **hide** to filter.



Plain answers: Ask, diaries and replay

You do not have to know filters or protocols to get answers from a capture. 🗨️ **Ask pCapitan** answers questions typed in your own words, the 📖 **Device diary** tells one device's story chapter by chapter, and the 🎞️ **Narrated replay** plays the capture back moment by moment on the map. Every answer is worked out from the capture itself, and comes with the evidence.

Ask pCapitan

🗨️ **Ask pCapitan** (Understand group in the sidebar) has one question box. Type a question and press **Enter** or click **Ask**, or click one of the example questions under the box.

sample.pcap **Apply**

Tip: press / to filter · type an IP, port, domain or expression · 📖 has ~2,000 ready-made filters explained in plain language

QUESTIONS IN PLAIN ENGLISH

Ask pCapitan

Ask about this capture in your own words. The answer is computed from the capture itself, with the evidence and a filter to see the packets. Nothing leaves this computer and no AI is involved.

What happened? Is any device compromised? Were any passwords sent in clear text? What was uploaded to the internet? Why is the network slow?

Which device is the busiest? Which devices used SSH? Who used port 445? Were any programs downloaded? How many devices are there?

Which companies' networks did we reach?

What did 192.168.1.66 do?

 192.168.1.66 was active from 23:13:34 to 23:13:43 (9.3 seconds) in 1 stretch of activity. It sent 46.7 KB and received 35.2 KB. 4 moments need attention (marked 🚩).

When	What
23:13:34 - 23:13:43: something to look at	291 conversations with 42 devices: sent 46.7 KB, received 35.2 KB; connected to intranet.corp.local over SSH (41×): 29.6 KB up, 22.1 KB down; connected to intranet.corp.local over TCP (other) (176×): 9.3 KB up, 9.3 KB down, 176 failed; connected to intranet.corp.local over HTTP (8×): 2.6 KB up, 2.2 KB down

✓ Wireshark 4.6.9

These are the kinds of question it understands:

You want to know	Example questions	The answer
What happened	<i>What happened? · Is any device compromised?</i>	The summary from 🌐 What happened? , with each device's verdict.
Who a device talked to	<i>Who talked to 192.168.1.5?</i>	Its peers, busiest first, with conversations, data and services.
What a device did	<i>What did 192.168.1.66 do? · Tell me about 192.168.1.66</i>	Its diary, in short.
Readable passwords	<i>Were any passwords sent in clear text?</i>	Every login seen in readable form: when, protocol, user, from, to.
Uploads	<i>What was uploaded to the internet?</i>	The biggest uploads from local devices to the internet.
Websites	<i>Which websites did 192.168.1.10 visit?</i>	Site names from encrypted connections (SNI) and web requests.
Name look-ups	<i>Who looked up google.com? · Show the DNS traffic</i>	The devices that looked up that name, or the most looked-up names.
Failed look-ups	<i>Show DNS failures</i>	Names that could not be looked up (no such name, server failure, refused), most frequent first.
Services and companies	<i>Is YouTube being watched? · Which services did we reach?</i>	The addresses of that service with the devices that used it, or every destination with its company and CDN.
Your internet provider	<i>What stayed inside the ISP? · What went to the internet?</i>	How much traffic went to cache nodes inside your provider's network and how much went further.
Countries	<i>Which countries? · Who talks to Germany?</i>	Internet addresses by country, or the ones in the country you named.
VPNs	<i>Is there a VPN?</i>	OpenVPN, WireGuard, IPsec, L2TP and PPTP connections.
The capture itself	<i>How many packets? · When did the capture start?</i>	Packets, data, start and end time, and length.
A service or port	<i>Which devices used SSH? · Who used port 445?</i>	Each pair of devices that used it, with conversations and data.
Slowness	<i>Why is the network slow?</i>	The slowest request-and-answer conversations, and who made them slow.
The busiest device	<i>Which device is the busiest? · Who used the most bandwidth?</i>	The devices that sent and received the most data.
Programs	<i>Were any programs downloaded?</i>	Executable files seen in the traffic, and web requests for <code>.exe</code> , <code>.dll</code> , <code>.msi</code> or <code>.ps1</code> files.
Device count	<i>How many devices are there?</i>	Every device, local or internet, with its type.

You want to know	Example questions	The answer
Who owns an address	<i>Who owns 8.8.8.8?</i>	The address facts: type, standard, registry, country and network owner.
Network owners	<i>Which companies' networks did we reach?</i>	Each organisation's network reached, with the number of addresses, the data and examples.

A device can be named by its address or by a name pCapitan knows for it in this capture, such as its host name. If you name a device and the question matches nothing else, you get that device's diary.

Each answer contains:

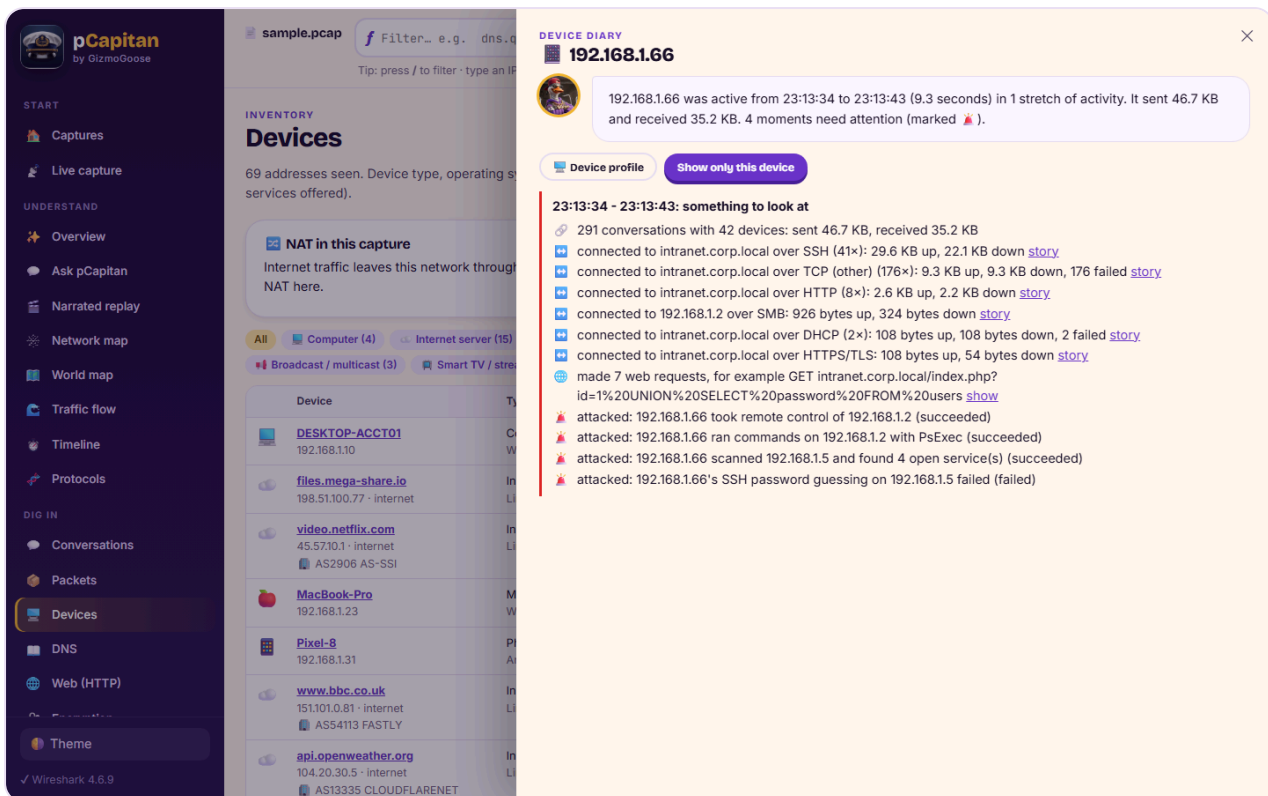
- **one sentence** that answers the question;
- an **evidence table** with the rows behind the sentence;
-  **Show these packets**, which applies a filter to the matching traffic;
- a link to go further:  **Its diary** for a device, or **Open the page** → for the page with the full detail (for example  **What happened?** or  **Devices**).

You can also type a **filter expression** in the box, for example `ip.addr == 10.0.0.1`, `http.user_agent contains "curl"` or `tcp.port == 8080 || udp.port == 53`. Anything that compares a field with a value, or a protocol name on its own such as `dns`, is treated as a filter: pCapitan answers *That is a filter* and offers  **Show these packets**.

When pCapitan does not understand a question, it says so. It lists what it can answer about and shows example questions you can click.

Device diary

The  **Device diary** tells what one device did, in time order. Open it with  **Diary** in the device drawer (click any device), or with  **Its diary** under an answer in Ask.



At the top is a summary: when the device was active, for how long, in how many stretches of activity, and how much it sent and received. If something needs attention, the summary says how many moments are marked 🚨.

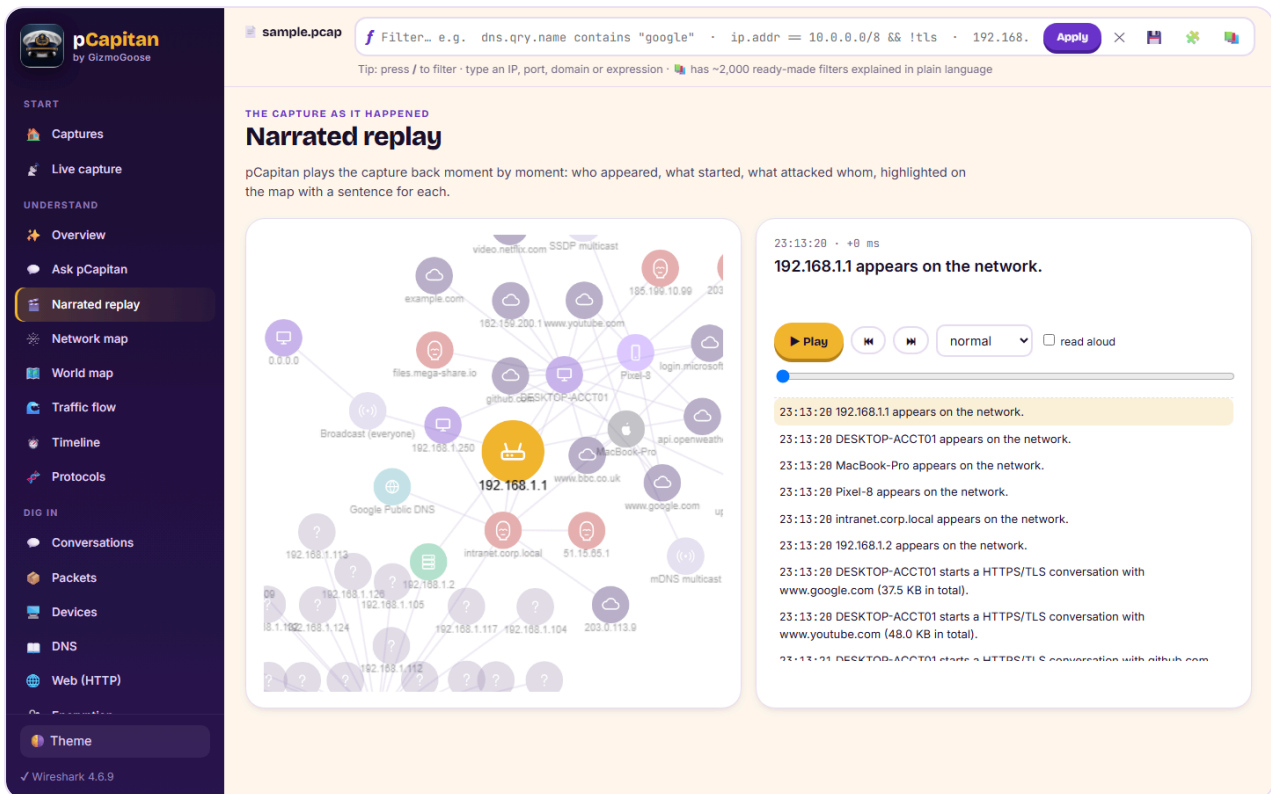
The diary is split into **chapters**. A new chapter starts after a quiet gap. The gap is one twelfth of the capture's length, but never less than 2 seconds and never more than 5 minutes. Each chapter has its start and end time; a chapter with a security event is headed *something to look at*.

Icon	Line
	The device asked for an address (DHCP), with the name it gave.
	How many names it looked up, with the first five. show filters to its look-ups.
	How many conversations it had, with how many devices, and the data sent and received.
	Its six biggest conversations in the chapter: with whom, which service, data up and down, and failed attempts. story opens the conversation's story.
	How many web requests it made, with an example. show filters to them.
	A login it sent in readable form.
	An attack it made, or one that targeted it, with the outcome.

The drawer also has **Device profile** and **Show only this device**.

Narrated replay

 **Narrated replay** (Understand group in the sidebar) plays the capture back as a series of moments. Each moment has its time, one sentence, and the devices involved.



The screenshot shows the pCapitan interface. On the left is a sidebar with a 'START' section containing 'Captures' and 'Live capture', and an 'UNDERSTAND' section containing 'Overview', 'Ask pCapitan', 'Narrated replay' (highlighted), 'Network map', 'World map', 'Traffic flow', 'Timeline', and 'Protocols'. Below this is a 'DIG IN' section with 'Conversations', 'Packets', 'Devices', 'DNS', 'Web (HTTP)', and 'Threats'. At the bottom of the sidebar is a 'Theme' button and a 'Wireshark 4.6.9' status bar. The main area has a filter bar at the top with a search box and an 'Apply' button. Below the filter bar is a section titled 'THE CAPTURE AS IT HAPPENED' with the heading 'Narrated replay'. A paragraph explains that pCapitan plays the capture back moment by moment, showing who appeared, what started, what attacked whom, highlighted on the map with a sentence for each. The main area is divided into two panels. The left panel shows a network map with various devices and connections, including 'video.netflix.com SSDP multicast', 'example.com', '192.168.1.250', 'files.mega-share.io', 'github.com', 'DESKTOP-ACCT01', 'Pixel-8', 'login.microsoft.com', 'api.openweath', 'MacBook-Pro', 'www.bbc.co.uk', 'www.google.com', 'intranet.corp.local', '51.15.95.1', 'mDNS multicast', 'Google Public DNS', '192.168.1.113', '192.168.1.124', '192.168.1.125', '192.168.1.105', '192.168.1.117', '192.168.1.104', '203.0.113.9', and '192.168.1.112'. The right panel shows a timeline of moments with a play button, a speed selector (normal), and a 'read aloud' checkbox. The timeline lists the following moments: '23:13:20 192.168.1.1 appears on the network.', '23:13:20 DESKTOP-ACCT01 appears on the network.', '23:13:20 MacBook-Pro appears on the network.', '23:13:20 Pixel-8 appears on the network.', '23:13:20 intranet.corp.local appears on the network.', '23:13:20 192.168.1.2 appears on the network.', '23:13:20 DESKTOP-ACCT01 starts a HTTPS/TLS conversation with www.google.com (37.5 KB in total).', '23:13:20 DESKTOP-ACCT01 starts a HTTPS/TLS conversation with www.youtube.com (48.0 KB in total).', and '23:13:21 DESKTOP-ACCT01 starts a HTTPS/TLS conversation with github.com'.

The moments are:

- **attacks and attempts:** every security finding that has an outcome, with the outcome;
- **devices appearing:** the first time each local device sends traffic (up to 40 devices);
- **big conversations:** the start of the 25 largest conversations, with their service and total size.

A replay holds up to 80 moments. If there are more, every attack and attempt is kept and the other moments are thinned out evenly.

On the **map**, the devices of the current moment are enlarged with a gold border, and the line between them is drawn as a gold arrow. For an attack or an attempt they turn red. The map zooms to them. Attackers are drawn with the  icon.

The controls:

- **► Play / || Pause** steps through the moments automatically. **⏮** and **⏭** go back or forward one moment.
- **Speed:** *slow* (5 seconds a moment), *normal* (3 seconds) or *fast* (1.5 seconds).
- **The slider** moves to any moment. **The list** under it shows every moment; click one to jump to it.
- **read aloud** speaks each sentence with the speech voice of your computer.
- For a big conversation,  **the story of this conversation** opens its story.

How this works without AI

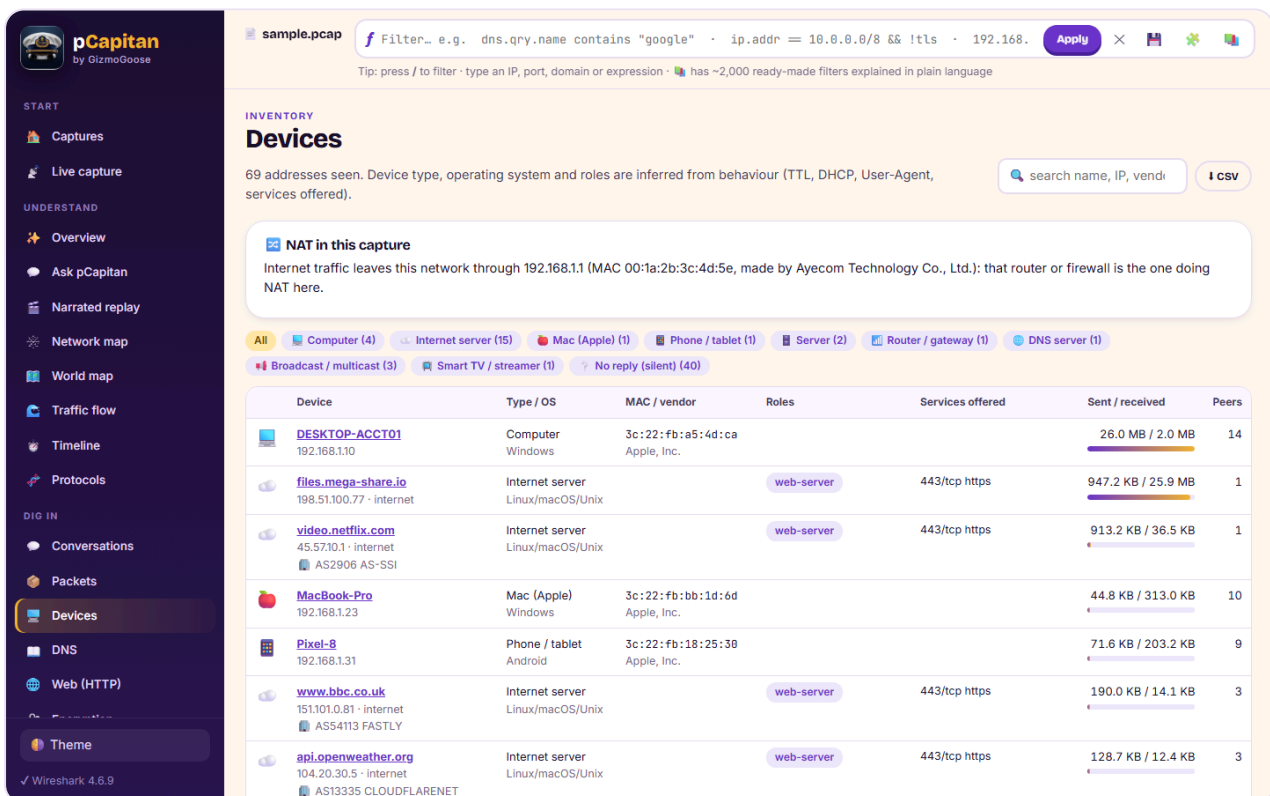
Who writes these answers? pCapitan does, with ordinary rules, not a language model.

- When you open a capture, pCapitan measures it and stores the results in a database of tables kept with that capture: devices, conversations, name look-ups, web requests, logins, findings and more.
- **Ask** compares your question with a fixed set of question patterns. It picks out the device, port, application or domain you named. It then runs a database query on the capture's tables and fills in a sentence template with the results.
- The **diary** and the **replay** are built from the same measured tables, with sentence templates.
- Nothing leaves your computer. No question or answer is sent anywhere.
- The same capture always gives the same answer to the same question.
- Every sentence has its evidence: the table under it, and a filter to the packets.

The limit: pCapitan understands only the kinds of question listed above. A question phrased in a way that matches none of the patterns is not understood. pCapitan then says so and shows examples.

Who is who

pCapitan works out what each device is from everything it reveals about itself, and shows it as an icon on the map and in  **Devices**:



sample.pcap Filter: e.g. dns.qry.name contains "google" · ip.addr == 10.0.0.0/8 && !tts · 192.168. Apply ×

Tip: press / to filter · type an IP, port, domain or expression · has ~2,000 ready-made filters explained in plain language

INVENTORY

Devices

69 addresses seen. Device type, operating system and roles are inferred from behaviour (TTL, DHCP, User-Agent, services offered).

search name, IP, vendor CSV

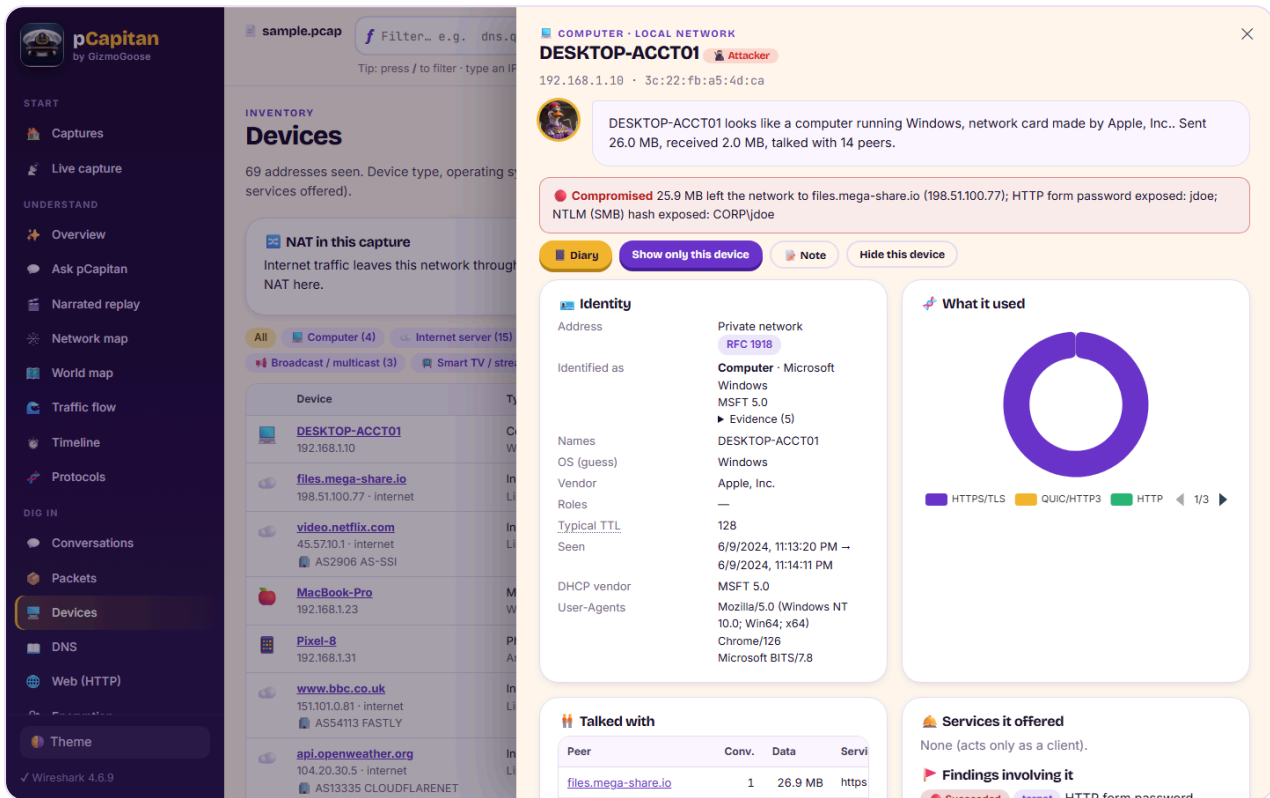
NAT in this capture
Internet traffic leaves this network through 192.168.1.1 (MAC 00:1a:2b:3c:4d:5e, made by Ayecom Technology Co., Ltd.): that router or firewall is the one doing NAT here.

All Computer (4) Internet server (15) Mac (Apple) (1) Phone / tablet (1) Server (2) Router / gateway (1) DNS server (1)
Broadcast / multicast (3) Smart TV / streamer (1) No reply (silent) (40)

Device	Type / OS	MAC / vendor	Roles	Services offered	Sent / received	Peers
 DESKTOP-ACCT01 192.168.1.10	Computer Windows	3c:22:fb:a5:4d:ca Apple, Inc.			26.0 MB / 2.0 MB	14
 files.mega-share.io 198.51.100.77 · Internet	Internet server Linux/macOS/Unix		web-server	443/tcp https	947.2 KB / 25.9 MB	1
 video.netflix.com 45.57.10.1 · Internet AS2906 AS-SSI	Internet server Linux/macOS/Unix		web-server	443/tcp https	913.2 KB / 36.5 KB	1
 MacBook-Pro 192.168.1.23	Mac (Apple) Windows	3c:22:fb:bb:1d:6d Apple, Inc.			44.8 KB / 313.0 KB	10
 Pixel-8 192.168.1.31	Phone / tablet Android	3c:22:fb:18:25:30 Apple, Inc.			71.6 KB / 203.2 KB	9
 www.bbc.co.uk 151.101.0.81 · Internet AS54113 FASTLY	Internet server Linux/macOS/Unix		web-server	443/tcp https	190.0 KB / 14.1 KB	3
 api.openweather.org 104.20.30.5 · Internet AS13335 CLOUDFLARENET	Internet server Linux/macOS/Unix		web-server	443/tcp https	128.7 KB / 12.4 KB	3

- **Hardware:** the maker of the network card, from the IEEE registry of MAC address prefixes (shipped with pCapitan). A randomised MAC address is recognised and names no maker.
- **Software:** what the device announces: LLDP and CDP from switches, phones and access points; the DHCP vendor class and host name; mDNS services; UPnP server strings; SSH, web, FTP and mail banners; TLS certificate names; browser User-Agents; and the operating system signature.

The most trustworthy source decides (LLDP and CDP first, the network card's maker last). For internet devices, only what they announce themselves counts, such as certificates and banners. The device drawer shows the result under  **Identity** (*Identified as*), with the evidence behind it.



The screenshot shows the pCapitan interface with a sidebar on the left containing navigation options like Captures, Live capture, Overview, Ask pCapitan, Narrated replay, Network map, World map, Traffic flow, Timeline, Protocols, Conversations, Packets, Devices, DNS, Web (HTTP), and Theme. The main area displays an 'INVENTORY Devices' list with 69 addresses seen. A device named 'DESKTOP-ACCT01' is highlighted, showing its IP (192.168.1.10) and MAC (3c:22:fb:a5:4d:ca). A detailed view of this device is shown on the right, indicating it is an 'Attacker'.

DESKTOP-ACCT01  **Attacker**

192.168.1.10 • 3c:22:fb:a5:4d:ca

DESKTOP-ACCT01 looks like a computer running Windows, network card made by Apple, Inc... Sent 26.0 MB, received 2.0 MB, talked with 14 peers.

Compromised 25.9 MB left the network to files.mega-share.io (198.51.100.77); HTTP form password exposed: jdoe; NTLM (SMB) hash exposed: CORP\jdoe

Identity

Address: Private network
RFC 1918

Identified as: **Computer** · Microsoft
Windows
MSFT 5.0
Evidence (5)
DESKTOP-ACCT01

Names: Windows
Roles: Apple, Inc.
Typical TTL: 128
Seen: 6/9/2024, 11:13:20 PM – 6/9/2024, 11:14:11 PM
DHCP vendor: MSFT 5.0
User-Agents: Mozilla/5.0 (Windows NT 10.0; Win64; x64) Chrome/126 Microsoft BITS/7.8

What it used

HTTPS/TLS QUIC/HTTP3 HTTP 1/3

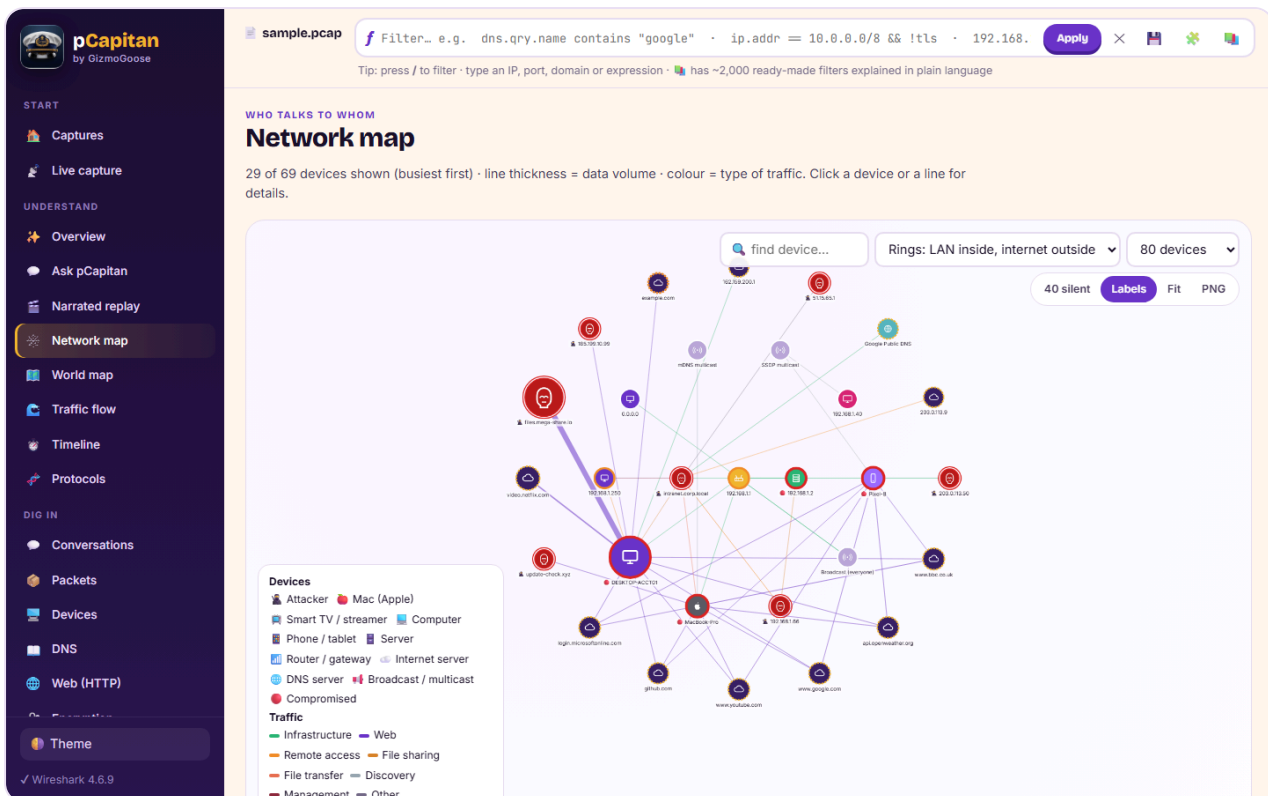
Talked with

Peer	Conv.	Data	Servi
files.mega-share.io	1	26.9 MB	https

Services it offered
None (acts only as a client).

Findings involving it
Succeeded target HTTP form password

The device types include  computer,  Mac,  phone,  server,  router,  firewall or NAT device,  switch,  wireless access point,  IP phone,  camera,  printer,  NAS,  domain controller,  DNS server and  internet server. A device that attacked another device, in a finding with an outcome, is drawn as  **Attacker** on the map.



Address facts. The device profile describes every address from the registries that define it:

- **Special-purpose addresses** come from the IANA special-purpose address registries: private networks (RFC 1918), carrier-grade NAT, loopback, link-local, documentation, benchmarking, multicast and others, with the RFC.
- **Public addresses** show the Regional Internet Registry that allocated them (AFRINIC, APNIC, ARIN, LACNIC or RIPE NCC) and the country.
- **The network owner** is the network (autonomous system) that announces the address, for example *AS15169 GOOGLE*. It appears as  under each internet device in **Devices** and as *Network* in the device profile.

The owner table is built from the public iptoasn.com tables and ships with pCapitan. Update it in **Updates & feeds** →  **Network owners (ASN)**. Lookups are made on your computer.

NAT. When a capture shows address translation,  **Devices** opens with a  **NAT in this capture** card, and Ask answers *What kind of NAT is this?*. pCapitan reads it from one capture in three ways:

- **STUN answers.** Video calls, WebRTC, games and phones ask STUN servers how the internet sees them. The answer gives the public address and port the NAT handed out. From it pCapitan tells, in the terms of RFC 4787:
 - **cone** (endpoint-independent): one inside port keeps one outside port for every destination, so peer-to-peer calls and games connect directly;
 - **symmetric** (endpoint-dependent): a new outside port for each destination, so peer-to-peer needs a relay;
 - whether **ports are kept** (port preservation);

- a **second NAT**: a public address inside 100.64.0.0/10 means carrier-grade NAT at the provider; a private address means a double NAT.

The mapping type needs the same inside port to reach two STUN servers; with only one, pCapitan says so. - **Addresses that hide several devices**. Connections from one address that start with different initial TTLs, web requests naming different operating systems, or several independent TCP timestamp clocks mean several devices share it: a NAT gateway or a proxy. pCapitan gives the lowest number of devices it can prove. - **The NAT device itself**. In a capture taken inside a network, the default gateway (where internet traffic goes) is the device doing NAT, and its network card's maker names the vendor.

Vendor hints come with it: outside ports all in 50001–65535 are pf (pfSense, OPNsense, OpenBSD); ports handed out one after another are typical of home routers and carrier-grade NAT; random ports of Cisco ASA, FortiGate or Linux with port randomisation; an MSS of 1380 is Cisco ASA's default clamp, and 1452 means a PPPoE link.

With two captures taken on both sides of a firewall or router,  **Compare captures** → **Two sides of a firewall or router** measures the same things directly from the matched packets, and adds TCP sequence randomisation (a Cisco ASA default) and MSS changes.

These are hints, not proof. Many devices can be set up to behave either way.

Proxies and firewalls in the path

A proxy hides the real client: the server only sees the proxy, or the firewall in front of it. pCapitan finds the real client in the headers the proxy adds to web requests.

- **X-Forwarded-For**: pCapitan takes the **right-most** valid address. The proxy adds that one; anything to the left of it was written by the client and can be forged.
- If there is no X-Forwarded-For, it reads X-Real-IP, True-Client-IP, CF-Connecting-IP, X-Client-IP, X-Cluster-Client-IP or the standard Forwarded header.
- **X-Forwarded-Host** gives the address the client connected to, and **X-Forwarded-Server** the name the proxy gives itself (right-most values only).

The device the server saw is called the **last hop**. pCapitan cannot always tell whether it is a firewall doing address translation (NAT) or the proxy itself, and says so. A last hop that pCapitan had listed as a computer, an internet server or an unknown device becomes  **Firewall / NAT**.

Where you see it:

-  **What happened?** shows a card  **Hidden behind a proxy** for each real client, with the path *real client* → *proxy* → *last hop* → *server*, the number of requests and the time span, what the client identified itself as, and  **All its requests**.
- The conversation's  **Story** has a section **Behind a proxy**.

- In the device drawer, the hidden client has its own profile ( **Hidden behind a proxy**), and the last hop shows  **Forwards traffic for hidden clients**.
- A finding under *Policy / visibility* names the hidden clients.
- The filter `http.real_client == ADDRESS` shows a hidden client's requests.

Unusual captures

Not every capture is a continuous recording. pCapitan checks what kind it is and adapts.

Kind	How pCapitan recognises it	What changes
Alert export from a firewall, IPS or IDS	At least 70% of TCP conversations have no start (no handshake), and they average 4 packets or fewer, or were collected on 3 or more days with long gaps.	The Overview says the capture keeps only the packets that triggered alerts. Traffic volumes, timing, connection health and slowness figures are not shown. Findings are never marked as still going on. The security findings apply.
Mid-stream recording	At least 70% of TCP conversations started before the capture did.	The Overview says connection set-up and health figures are incomplete. No retransmission finding is raised.

Damaged files. If a capture file is damaged, for example cut off or corrupted while it was written or copied, pCapitan looks for the next valid packet after the damage and carries on from there. The **Overview** then shows a warning: in how many places the file is damaged, how many bytes were skipped, and where the first damage is. Every readable packet around the damage is analysed. This also covers a `.gz` file that was cut off (everything before the break is analysed), damaged pcapng blocks, and packet times that cannot be right (for example from a wrong time resolution in the file), which are shown as unknown instead of stopping the analysis.

Formats. pCapitan reads pcap (micro- and nanosecond, including the "modified" variant), pcapng (several sections and interfaces), Microsoft Network Monitor 2.x and gzip-compressed captures. Link types include Ethernet (with VLAN and PPPoE), Linux cooked capture, raw IP, BSD loopback, PPP and 802.11 Wi-Fi (radiotap, Prism): data on an open Wi-Fi network is analysed like wired traffic. Network Monitor 1.x files must first be saved as pcapng in Wireshark.

If the file is damaged at the very first packet and nothing valid follows, pCapitan cannot open it and says the file is corrupt.

Digging into the details

Conversations

A **conversation** is all the packets exchanged between two programs (same addresses, ports and protocol). The table shows:

- client, server and service;
- **state**: Completed, No reply, Rejected, Reset...;
- duration, data in each direction, round-trip time and problems;
- a one-line plain-language summary.

Click a column header to sort. Problem rows are tinted.

pCapitan by GizmoGoose

sample.pcap Filter: e.g. dns.qry.name contains "google" · ip.addr == 10.0.0.0/8 && !tls · 192.168. Apply

Tip: press / to filter · type an IP, port, domain or expression · has ~2,000 ready-made filters explained in plain language

EVERY CONVERSATION

Conversations

492 conversations (a conversation = all packets between two programs). Click one for its full story, a flowchart of the exchange, and its reassembled content. [CSV](#)

#	Started	Client	Server	Service	Encryption	State	Duration	Data ↑ / ↓ ▼	Pkts	RTT
467	+32.6 s	DESKTOP-ACCT01 :52000	files.mega-share.io tcp/443	HTTPS/TLS files.mega-share.io	Encrypted	Completed	12.6 s	25.9 MB / 947.2 KB	35,922	20 r
42	+12.2 s	DESKTOP-ACCT01 :52974	video.netflix.com tcp/443	HTTPS/TLS video.netflix.com	Encrypted	Completed	1.7 s	36.5 KB / 913.2 KB	1,300	30 r

The conversation story

Click any conversation to open its **story**. It explains, step by step:

1. **What happened:** the service, the site name, and the most likely activity (web page, large download, interactive session, video call, keep-alive / possible C2 ...).
2. **Opening the connection:** the handshake, with timings and what they mean.
3. **Agreeing on encryption:** TLS version, cipher, certificate and client fingerprint, with warnings for weak or self-signed setups.

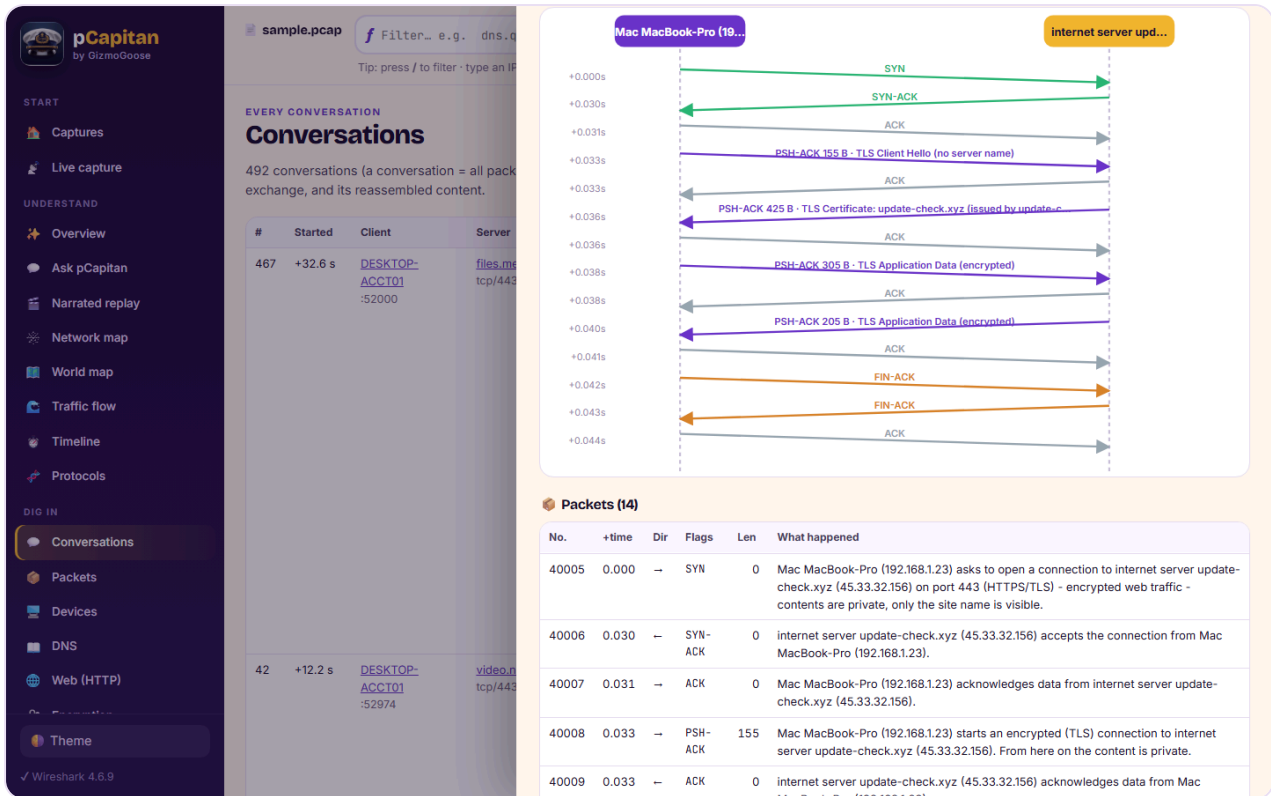
4. **The conversation:** volumes, turns, pauses, web requests and DNS answers.
5. **Ending:** who closed it, and whether cleanly or abruptly.
6. **Network health:** retransmissions and stalls.
7. **Security assessment:** anything risky, including related findings.

The screenshot displays the pCapitan network analysis tool interface. On the left is a dark sidebar with navigation options: START (Captures, Live capture), UNDERSTAND (Overview, Ask pCapitan, Narrated replay, Network map, World map, Traffic flow, Timeline, Protocols), and DIG IN (Conversations, Packets, Devices, DNS, Web (HTTP), Theme). The main panel shows a list of conversations under the heading 'EVERY CONVERSATION'. A specific conversation is selected, showing details for 'CONVERSATION #474 · TCP' between 'Mac MacBook-Pro (192.168.1.23)' and 'internet server update-check.xyz (45.33.32.156):443'. The status is 'Completed', 'Encrypted', and 'Web API call / small page'. Performance metrics show 44 ms, 1.8 KB, and RTT 30 ms. Below this, a 'Who made it slow?' section features a bar chart showing Network at 94%, Server at 6%, and Client at 0%. A detailed breakdown follows: Network (62 ms, 2 round trips, 0 retransmissions), Server (4.0 ms, think time 2.0 ms on average, longest 2.0 ms at packet #40014), and Client (0.0 ms). The 'The story, step by step' section provides a narrative: 'What happened' (client contacted server), 'Opening the connection' (SYN, SYN-ACK, ACK exchange), and 'Agreeing on encryption' (TLS setup with fingerprints and certificates).

Below the story, the **flowchart** draws every packet as an arrow between client (purple) and server (gold):

- green arrows are connection opening;
- purple arrows carry data;
- orange arrows are closing;
- red arrows are resets and retransmissions.

Hover an arrow for its explanation, or click it to dissect the packet.



📄 **Follow stream** reassembles the actual bytes exchanged. ↓ **.pcap** saves just this conversation.

The path to the server. A conversation with an internet address also shows how far away the server is (the number of router hops, read from the packets) and what is on the way. Click **Trace the route** to measure it now: pCapitan runs your computer's own traceroute to that address and lists every hop with its delay, who owns it (your network, carrier-grade NAT, your internet provider, the service's network) and where the delay grows. A hop marked **congestion** adds delay that stays for every hop after it: that is where the slowdown is. A hop marked **slow reply** only answers traceroute slowly (routers give these answers low priority) and is not a problem. Tracing works from the computer running pCapitan, for addresses it can reach.

Packets

Every packet in order. Switch between 🗣️ **Plain language** (a sentence per packet) and 🔧 **Technical** (the classic info column). ↓ **Export these packets** writes the currently filtered packets to a new .pcap you can open in Wireshark.

sample.pcap Filter: e.g. dns.qry.name contains "google" · ip.addr == 10.0.0.0/8 && !tls · 192.168. Apply

Tip: press / to filter · type an IP, port, domain or expression · has ~2,000 ready-made filters explained in plain language

PACKET BY PACKET

Packets

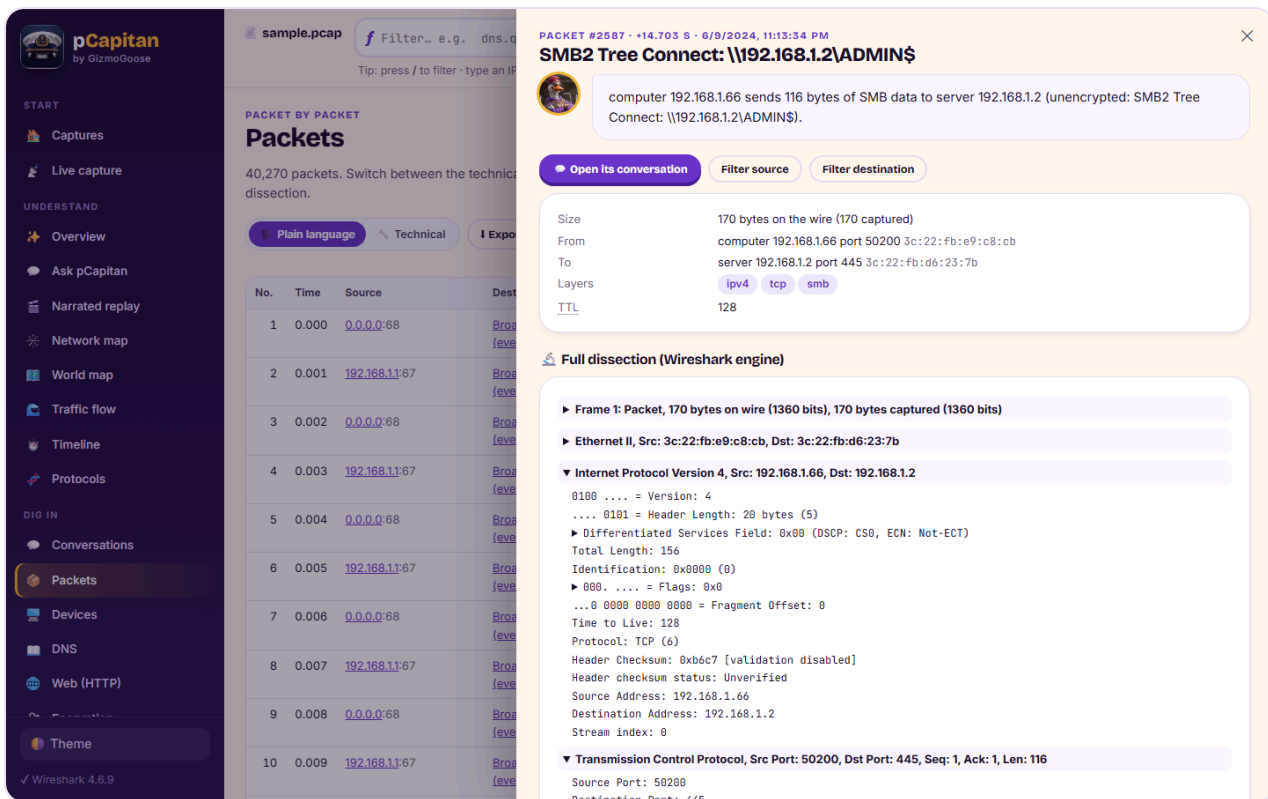
40,270 packets. Switch between the technical view and plain-language sentences. Click any packet for its full dissection.

Plain language Technical **Export these packets (.pcap)**

No.	Time	Source	Destination	Protocol	Encryption	Len	What happened
1	0.000	0.0.0.0:68	Broadcast (everyone):67	DHCP		318	DHCP DISCOVER from DESKTOP-ACCT01 (wants 192.168.1.10).
2	0.001	192.168.1.1:67	Broadcast (everyone):68	DHCP		330	DHCP OFFER: server offers/confirm 192.168.1.10 for DESKTOP-ACCT01.
3	0.002	0.0.0.0:68	Broadcast (everyone):67	DHCP		318	DHCP REQUEST from DESKTOP-ACCT01 (wants 192.168.1.10).
4	0.003	192.168.1.1:67	Broadcast (everyone):68	DHCP		330	DHCP ACK: server offers/confirm 192.168.1.10 for DESKTOP-ACCT01.
5	0.004	0.0.0.0:68	Broadcast (everyone):67	DHCP		318	DHCP DISCOVER from Pixel-8 (wants 192.168.1.31).
6	0.005	192.168.1.1:67	Broadcast (everyone):68	DHCP		330	DHCP OFFER: server offers/confirm 192.168.1.31 for Pixel-8.
7	0.006	0.0.0.0:68	Broadcast (everyone):67	DHCP		318	DHCP REQUEST from Pixel-8 (wants 192.168.1.31).
8	0.007	192.168.1.1:67	Broadcast (everyone):68	DHCP		330	DHCP ACK: server offers/confirm 192.168.1.31 for Pixel-8.
9	0.008	0.0.0.0:68	Broadcast (everyone):67	DHCP		305	DHCP DISCOVER from MacBook-Pro (wants 192.168.1.23).
10	0.009	192.168.1.1:67	Broadcast (everyone):68	DHCP		317	DHCP OFFER: server offers/confirm 192.168.1.23 for MacBook-Pro.

Click a packet for its full dissection:

- the plain-language explanation;
- the main fields;
- the complete Wireshark protocol tree (when Wireshark is installed);
- the payload as text;
- the raw bytes.



Devices

The inventory of every address seen. Each entry shows:

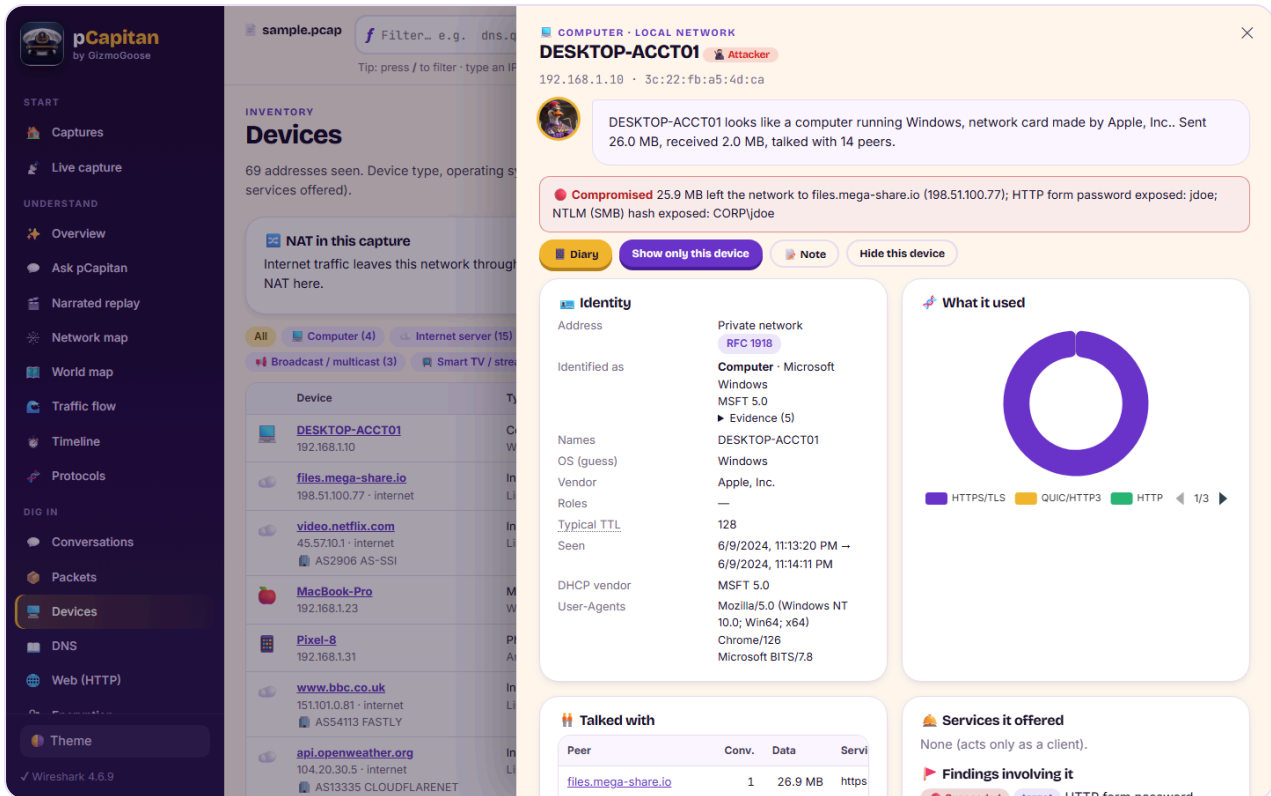
- **device type** and **guessed operating system**, from DHCP, User-Agent, names and the services it offered. pCapitan names a type (computer, phone, printer, router...) only when the traffic proves it; otherwise the device is simply shown as ♦ **Device** with its name or IP address, rather than a guess that could mislead you;
- network-card **vendor** and **roles** (gateway, DNS/DHCP server, domain controller, web, file or database server);
- the services it offered and how much it sent and received.

Filter by type with the chips, or search by name, IP or vendor.

Click a device for its profile:

- identity;
- what it used;
- who it talked to;
- services offered;
- the findings involving it;
- the names it looked up;
- activity per minute.

For internet addresses there are one-click **VirusTotal**, **AbuseIPDB** and **Shodan** lookups. These open your browser, and only the IP address is sent.



Destinations

Destinations lists every internet address the capture reached and what it really is. Many services sit behind other companies: a game catalogue served by Akamai, a news page from a Microsoft cache, a shop behind Cloudflare. For each address pCapitan shows the whole chain:

- the **name that was asked for** and every **DNS alias** (CNAME) on the way;
- the **service and company** behind it (from pCapitan's built-in list of well-known services);
- the **CDN or hosting platform** that actually answered;
- the **network owner** of the address;
- where it sits: inside **your internet provider's network** (a cache node close to you), in the **service's own network**, or **elsewhere on the internet**;
- the devices that used it and the data exchanged.

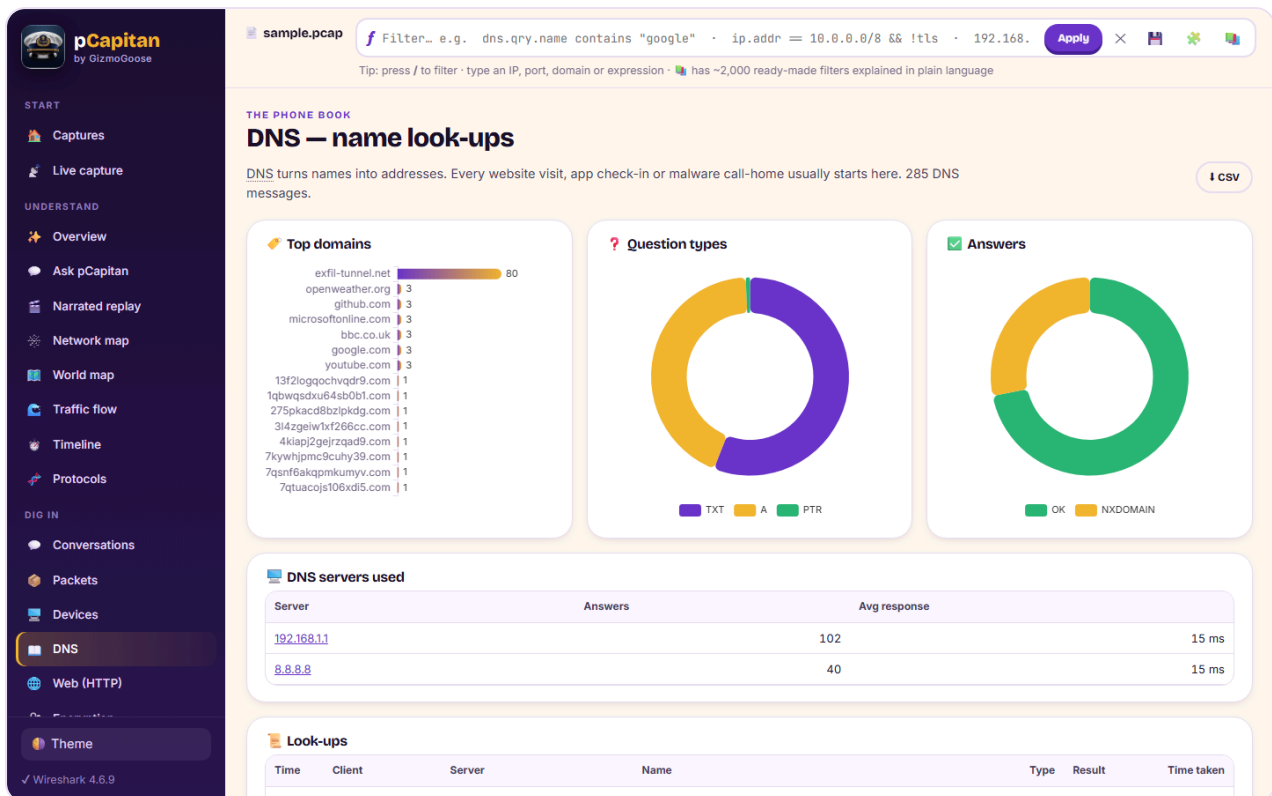
The bar at the top shows how much of your internet traffic stayed inside your provider's network and how much went out to the internet. Your provider is recognised from the capture itself (STUN answers, or content caches that sit on its network).

DNS

DNS is the internet's phone book: almost every connection starts with a look-up. This screen shows:

- top domains, question types and answer codes (*NXDOMAIN* means "no such name");
- the DNS servers used and how fast they answered;
- every look-up.

Many failed look-ups of random-looking names is a classic malware sign, and pCapitan flags it.



Web (HTTP)

Unencrypted web traffic is fully readable: sites, status codes, content types, every **User-Agent** (browser or program) and every request. Encrypted web is on the next screen.

Encryption (TLS and QUIC)

With encrypted traffic the content is private, but much is still visible:

- **sites** (SNI);
- **versions** (TLS 1.0 / 1.1 / SSL are weak);
- **certificates** (self-signed, expired, issuer);
- **client fingerprints** (JA3 / JA4), which identify the software making the connection;
- **QUIC / HTTP/3** handshakes, which pCapitan decrypts to recover their site names.

sample.pcap Filter: e.g. dns.qry.name contains "google" · ip.addr == 10.0.0.0/8 && !tls · 192.168. Apply × 🔍 🌐 📄

Tip: press / to filter · type an IP, port, domain or expression · 📖 has ~2,000 ready-made filters explained in plain language

ENCRYPTED TRAFFIC

Encryption (TLS & QUIC)

Content is private, but the site name, protocol version, certificate and the client's fingerprint are visible. QUIC (HTTP/3) handshakes are decrypted to reveal their site names. Decryption keys CSV

Decryption Vault

One encrypted vault for every key pCapitan may decrypt with: server private keys and certificates (.pfx, .p12, .crt + .key), TLS key-logs, Kerberos keytabs, SSH key-logs, Wi-Fi passwords, TLS pre-shared keys and IPsec ESP keys. Only the **Admin** and the **Captain** can open it; it is encrypted on disk (AES-256), locks itself after 15 minutes without use, and pCapitan matches every capture against what is inside.

42
Encrypted conversations

0
Decrypted

0
Ready with a vault key

42
Need a key-log

Create the Decryption Vault (Admin)

Choose the Admin password (at least 10 characters). Keep it safe: without it or the Captain password, nobody can open the vault, not even GizmoGoose.

Admin password
 Repeat it
Create the vault

▼ Why can't a server's private key decrypt TLS 1.3 or ECDHE? And how to get what can.

With **RSA key exchange** (TLS 1.0-1.2, cipher names without ECDHE / DHE) the client encrypts the session secret with the server's public key, so the server's private key recovers it. With **ECDHE / DHE**, and in every **TLS 1.3** and QUIC connection, both sides create a temporary secret that never crosses the network and is thrown away (forward secrecy): the server's key only proves who it is. Only a key-log file from the client or the server can open those sessions. The table below says which case each conversation is.

Option A: export the session keys (works for everything, TLS 1.3 and QUIC included)

Set SSLKEYLOGFILE before the program starts, reproduce the traffic while capturing, then add the file here.

Windows (Chrome, Edge, Firefox, curl)

```
setx SSLKEYLOGFILE "%USERPROFILE%\Desktop\sslkeys.log"
```

Then close and reopen the browser.

Linux / macOS

```
export SSLKEYLOGFILE=~/.sslkeys.log
```

Decrypting HTTPS with a key-log file

If you have the **SSLKEYLOGFILE** from the browser or app that made the traffic, add it to the **Decryption Vault** on this page and click **Decrypt this capture**. pCapitan, using the Wireshark engine, decrypts the TLS and QUIC sessions. The vault also takes server private keys, .pfx / .p12 files, Kerberos keytabs, SSH key-logs, Wi-Fi passwords and IPsec keys: see *Decrypting traffic: the Decryption Vault*. The HTTP/2 and HTTP/3 requests inside then appear on the Web screen, and any credentials they carried appear in Forensics.

To create a key-log file:

- **Windows:** `setx SSLKEYLOGFILE "%USERPROFILE%\sslkeys.log"` , then restart Chrome or Firefox and capture.
- **macOS / Linux:** `export SSLKEYLOGFILE=~/.sslkeys.log` , then start the browser from that terminal.

The keys never leave your computer.

Filters

One filter, typed once, applies to **every screen**: map, charts, tables, stories, exports and the report. As you type, pCapitan:

- translates the filter into English under the bar;
- shows how many packets match;
- suggests field names (**Tab** or ↓ to pick one).

 **pCapitan**
by GizmoGoose

START

Captures

Live capture

UNDERSTAND

Overview

Ask pCapitan

Narrated replay

Network map

World map

Traffic flow

Timeline

Protocols

DIG IN

Conversations

Packets

Devices

DNS

Web (HTTP)

...

Theme

✓ Wireshark 4.6.9

sample.pcap

f ip.src == 192.168.1.66 && tcp.dstport in {22 445}

Apply

×

📄

🌱

🎨

Show traffic where source IP (CIDR allowed) is 192.168.1.66 AND TCP destination port is one of {22, 445}.

PACKET BY PACKET

Packets

332 packets. Switch between the technical view and plain-language sentences. Click any packet for its full dissection.

Plain language

Technical

Export these packets (.pcap)

No.	Time	Source	Destination	Protocol	Encryption	Len	What happened
2584	14.680	192.168.1.66:50200	192.168.1.2:445	SMB		54	computer 192.168.1.66 asks to open a connection to server 192.168.1.2 on port 445 (SMB) - Windows file sharing / network drives.
2586	14.701	192.168.1.66:50200	192.168.1.2:445	SMB		54	computer 192.168.1.66 acknowledges data from server 192.168.1.2.
2587	14.703	192.168.1.66:50200	192.168.1.2:445	SMB		170	computer 192.168.1.66 sends 116 bytes of SMB data to server 192.168.1.2 (unencrypted: SMB2 Tree Connect: \\192.168.1.2\ADMIN\$).
2589	14.706	192.168.1.66:50200	192.168.1.2:445	SMB		228	computer 192.168.1.66 sends 174 bytes of SMB data to server 192.168.1.2 (unencrypted: SMB2 Create (open file): Windows\Temp\PSXESVC.exe).
2591	14.708	192.168.1.66:50200	192.168.1.2:445	SMB		190	computer 192.168.1.66 sends 136 bytes of SMB data to server 192.168.1.2 (unencrypted: SMB2 Create (open file): svcctl).
2593	14.711	192.168.1.66:50200	192.168.1.2:445	SMB		122	computer 192.168.1.66 sends 68 bytes of SMB data to server 192.168.1.2 (unencrypted).
2595	14.712	192.168.1.66:50200	192.168.1.2:445	SMB		54	computer 192.168.1.66 politely closes its side of the connection with server 192.168.1.2.
2597	14.714	192.168.1.66:50200	192.168.1.2:445	SMB		54	computer 192.168.1.66 acknowledges data from server 192.168.1.2.
2648	14.759	192.168.1.66:40022	intranet.corp.local:22	SSH		54	computer 192.168.1.66 asks to open a connection to server intranet.corp.local (192.168.1.5) on port 22 (SSH) - encrypted remote command-line administration.
2650	14.760	192.168.1.66:40022	intranet.corp.local:22	SSH		54	computer 192.168.1.66 abruptly ends/refuses the connection with server intranet.corp.local (192.168.1.5) (reset).
3009	14.903	192.168.1.66:40445	intranet.corp.local:445	SMB		54	computer 192.168.1.66 asks to open a connection to server intranet.corp.local (192.168.1.5) on port 445 (SMB) - Windows file sharing / network drives.

Quick filters (no syntax needed)

Type	Shows
192.168.1.5	everything to or from that address
10.0.0.0/8	a whole network
443	anything on port 443
google.com	anything related to that name (DNS, site names, host names)
dns , tls , http , quic , arp , smb , kerberos , wifi	that protocol
host 8.8.8.8 , port 53 , src host 10.0.0.2	tcpdump-style shortcuts

The filter language

It works like Wireshark's display filters:

```
field operator value
```

Operators:

- comparison: `==`, `!=`, `>`, `<`, `>=`, `<=`
- text: `contains`, `not contains`, `matches "regex"`, `startswith`, `endswith`
- sets: `in {80 443 8000..8100}`, `not in {...}`

Logic: `&&` / `and`, `||` / `or`, `!` / `not`, plus `xor` (one or the other but not both), `nand` (not both) and `nor` (neither). Use parentheses to group.

Values:

- sizes: `1k`, `5M`, `2G`
- times: `frame.time >= "2024-05-01 10:00"`, `frame.time_relative <= 30`

Examples:

```
ip.addr == 10.0.0.0/8 && !tls
dns.qry.name contains "google" || tls.sni matches "^api\."
flow.bytes > 1M and traffic.outbound
tcp.flags.syn == 1 && tcp.flags.ack == 0 && flow.state in {S0 REJ}
http.user_agent matches "sqlmap|nikto|curl"
payload contains "password"           (searches inside every packet's bytes)
host.country == "RU"
kerberos.etype == 23                   (weak RC4 Kerberos tickets)
wlan.subtype == deauthentication
ids.sid == "2001219"
```

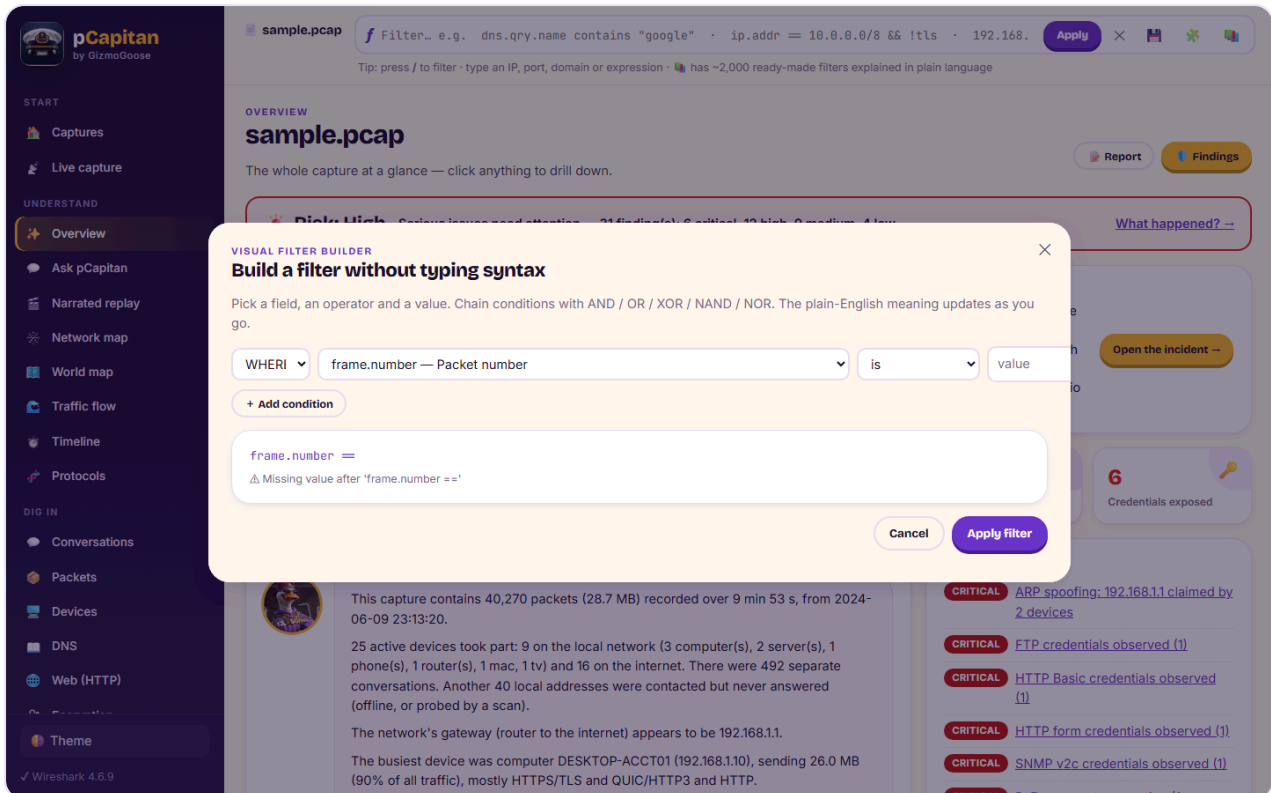
Open  **Filter library** →  **All fields** for the complete list of more than 120 fields, each with an explanation.

The visual filter builder

Click  next to the filter bar:

1. Pick a field, an operator and a value.
2. Chain conditions with AND / OR / XOR / NAND / NOR / AND NOT.

The English meaning updates as you build.



The filter library

 **Filter library** holds **over 2,000 ready-made filters**. Each one is written out in plain English, with a note on when to use it. They are grouped into:

- basics, direction and TCP behaviour;
- performance, DNS, web and encryption;
- Windows & AD and remote access;
- threat hunting, credentials and devices;
- every service and port, in useful combinations.

Click  next to the filter bar to open the library as a **drawer above the screen you are on**. Type words to search (every word must match: `smb admin` finds the SMB admin-share filters), pick a group, and click **Apply**. The filter is applied to the current screen and the drawer stays open, so you can try another. **+** **and** adds a filter to the one already in the filter bar. Press **Esc** or **X** to close it.

pCapitan
by GizmoGoose

START

- Captures
- Live capture

UNDERSTAND

- Overview
- Ask pCapitan
- Narrated replay
- Network map
- World map
- Traffic flow
- Timeline
- Protocols

DIG IN

- Conversations
- Packets**
- Devices
- DNS
- Web (HTTP)
- Theme

✓ Wireshark 4.6.9

sample.pcap

Filter: e.g. dns.qry.name contains "google" · ip.addr == 10.0.0.0/8 && !tls · 192.168. Apply ×

Tip: press / to filter · type an IP, port, domain or expression · has ~2,000 ready-made filters explained in plain language

2,497 matches Full library & combine →

All Basics Direction TCP behaviour Performance DNS Web Encryption Windows & AD Remote access E-mail Threat hunting Credentials IoT

Everything except LAN chatter `!(arp || mdns || llmnr || nbns || ssdp || stp || lldp || cdp || igmp)`

Show traffic where NOT (ARP address look-ups on the LAN, OR local mDNS discovery, OR LLMNR name look-ups, OR NetBIOS name service, OR SSDP/UPnP (smart devices (TVs, routers, speakers) advertising themselves) traffic, OR spanning-tree switch messages, OR LLDP device announcements, OR Cisco discovery messages, OR multicast group management)).

Apply + and

Only IPv4 `ipv4`

Show traffic where IPv4 traffic.

Apply + and

Only IPv6 `ipv6`

Show traffic where IPv6 traffic.

Apply + and

TCP only `tcp`

Show traffic where TCP traffic (connection-based).

Apply + and

UDP only `udp`

Show traffic where UDP traffic (connectionless).

Apply + and

TCP or UDP but never both `tcp xor udp`

Apply + and

PACKET BY PACKET

Packets

40,270 packets. Switch between the technical view and plain-language sentences. Click any packet for its full dissection.

Plain language Technical Export these packets (.pcap)

No.	Time	Source	Destination	Protocol	Encryption	Len	What happened
1	0.000	0.0.0.0:68	Broadcast (everyone):67	DHCP		318	DHCP DISCOVER from DESKTOP-ACCT01 (wants 192.168.1.10).

The full **Filter library** page (Tools group) lists the same filters. There you can also click **+ Combine** on several filters and join them with AND / OR / XOR / NAND / NOR / AND NOT.

pCapitan
by GizmoGoose

START

- Captures
- Live capture

UNDERSTAND

- Overview
- Ask pCapitan
- Narrated replay
- Network map
- World map
- Traffic flow
- Timeline
- Protocols

DIG IN

- Conversations
- Packets**
- Devices
- DNS
- Web (HTTP)
- Theme

✓ Wireshark 4.6.9

sample.pcap

Filter: e.g. dns.qry.name contains "google" · ip.addr == 10.0.0.0/8 && !tls · 192.168. Apply ×

Tip: press / to filter · type an IP, port, domain or expression · has ~2,000 ready-made filters explained in plain language

All Basics Direction 8 TCP behaviour 55 Performance 14 DNS 46 Web 81 Encryption 23 Windows & AD 16 Remote access 7 E-mail 4 Threat hunting 25 Credentials 5 IoT & Industrial 6 Devices 11 Size & volume 21 Service: Messaging 59 Service: Mobile 28 Service: Infrastructure 130 Service: Industrial 75 Service: Routing 56 Service: P2P 13

2,497 match(es)

Everything except LAN chatter Basics

`!(arp || mdns || llmnr || nbns || ssdp || stp || lldp || cdp || igmp)`

Show traffic where NOT (ARP address look-ups on the LAN, OR local mDNS discovery, OR LLMNR name look-ups, OR NetBIOS name service, OR SSDP/UPnP (smart devices (TVs, routers, speakers) advertising themselves) traffic, OR spanning-tree switch messages, OR LLDP device announcements, OR Cisco discovery messages, OR multicast group management)).

Hide background noise so the real conversations stand out.

Apply + Combine

Only IPv4 Basics

`ipv4`

Show traffic where IPv4 traffic.

Restrict to classic IPv4 traffic.

Apply + Combine

Only IPv6 Basics

`ipv6`

Show traffic where IPv6 traffic.

Is IPv6 in use on this network?

Apply + Combine

Combine filters

Pick filters with "+" + Combine, choose how to join them.

Nothing picked yet.

AND OR XOR NAND NOR

AND NOT

Apply combination

Language cheat-sheet

= < > <= >= compare · contains · matches

"regex" · in {80 443 8080..8180}

&&/and · ||/or · !/not · xor · nand · nor · ()

Shortcuts: 192.168.1.5 · 10.0.0.0/8 · 443 · google.com · host x · port n · src host x

Sizes/times: 1k 5M 2G · frame.time >= "2024-05-01 10:00"

Content: payload contains "text"

All 129 fields

Your own filters

Type a filter and click  (save) next to the filter bar. Give it a name and a note. It appears in the library under **My filters**, where you can apply, combine or delete it.

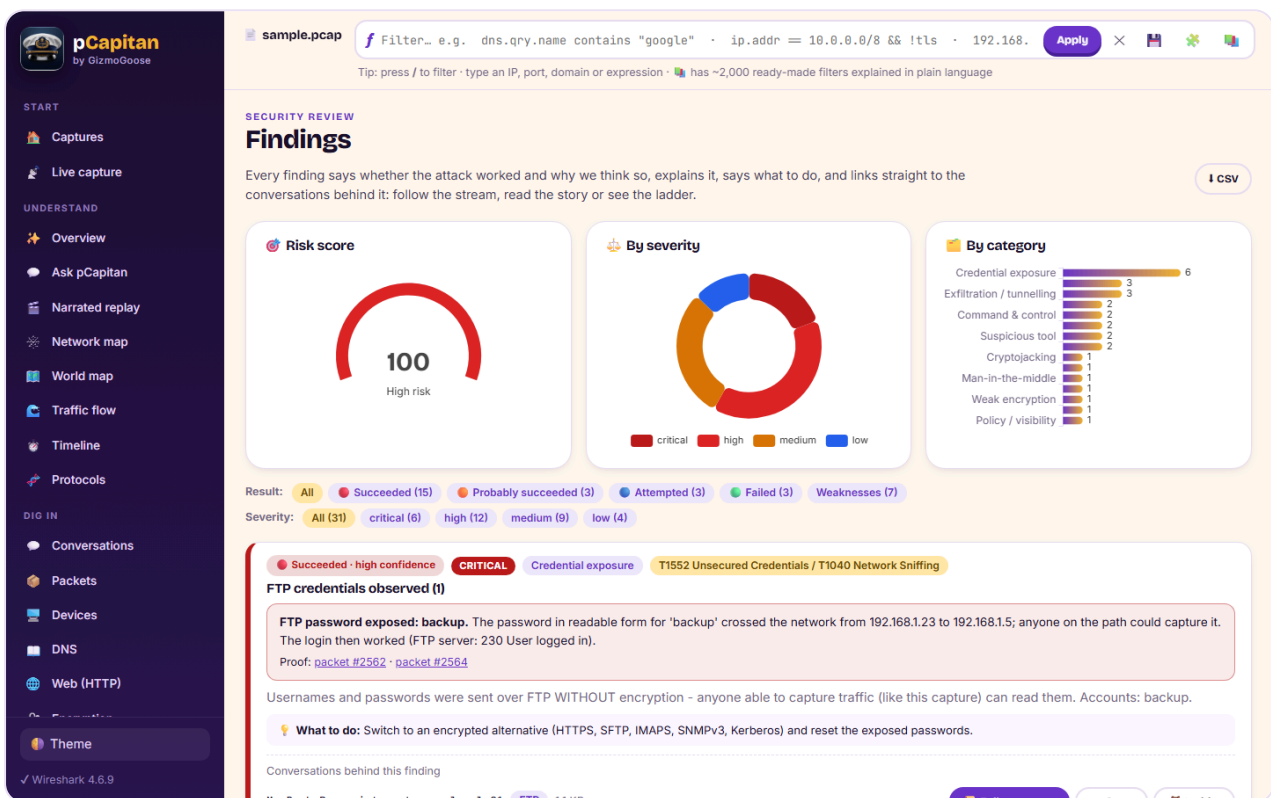
Security and forensics

Findings

pCapitan runs **more than 80 detectors** on every capture. Each finding comes with:

- **severity**: critical, high, medium, low or info;
- a **plain-language explanation** of what was seen and why it matters;
- **what to do**;
- its **MITRE ATT&CK** technique;
- the **evidence**.

Click  **Investigate** to filter the whole app to that evidence.



What it looks for:

- **Reconnaissance**: port scans, ping and ARP sweeps, web content discovery, user enumeration.
- **Credential attacks**: brute force, password spraying, SMB password guessing (from the server's refusals), Kerberos user enumeration and spraying, Kerberoasting, AS-REP roasting, accounts without Kerberos pre-authentication, DCSync, repeated HTTP log-in failures.
- **Exploits**: EternalBlue (MS17-010) and the DoublePulsar implant, Zerologon, Heartbleed, Log4Shell (and whether the callback happened), remote WMI, RPC endpoint enumeration.

- **Command & control:** beaconing (regular "phone home" connections), repeated check-ins to a bare address, plain HTTP on encrypted ports, encrypted channels on unusual ports, fake browser identities, DGA domains, persistent low-volume sessions, IRC, Tor.
- **Exfiltration and tunnelling:** large uploads, DNS tunnels, ICMP tunnels, encrypted DNS (DoH) and bypassing the company DNS.
- **Man-in-the-middle:** ARP spoofing, rogue DHCP, rogue IPv6 routers, LLMNR/NBT-NS poisoning (Responder), evil-twin Wi-Fi.
- **Lateral movement:** admin-share access, PsExec, remote services and scheduled tasks, DCSync, executables copied over SMB.
- **Malware indicators:** executables over HTTP, PowerShell download cradles, reverse shells, Log4Shell strings, cryptomining, known-bad TLS fingerprints, threat-feed matches, IDS signature hits and antivirus hash matches.
- **Exposure and hygiene:** clear-text credentials, insecure protocols (Telnet, FTP, SNMP v1/v2, SMBv1), weak, expired or self-signed TLS, open Wi-Fi, amplification-attack participation.
- **Network health:** retransmissions, overwhelmed receivers, slow DNS, many "unreachable" errors.

A finding is a **lead, not a verdict**. pCapitan explains when something can also be legitimate, such as a backup job that looks like exfiltration, or an updater that looks like beaconing.

Forensics

-  **Credentials:** logins visible in the traffic (FTP, HTTP Basic and forms, POP3/IMAP/SMTP, SNMP communities, NTLM user names, RDP user names). Secrets are masked until you click **Reveal**.
-  **Extracted files:**  **Extract files** carves files out of HTTP, SMB, TFTP, FTP, e-mail and DICOM. It hashes them (SHA-256), identifies their type, and offers a hash-only VirusTotal lookup. Files can be malicious, so open them with care.
-  **File-share activity:** SMB shares and files opened. Admin shares (ADMIN\$, C\$) and control pipes (svcctl, atsvc) are highlighted.
-  **IOC matching:** paste IPs, networks, domains, URLs, JA3/JA4 fingerprints or file hashes (defanged `evil[.]com` is fine), and see every match.
-  **Payload hunt:** search for text or a regex inside every packet's raw bytes. The result becomes a filter.
-  **DHCP**,  **ARP** and  **Content indicators**.

sample.pcap Filter: e.g. dns.qry.name contains "google" · ip.addr == 10.0.0.0/8 && !tls · 192.168. Apply

Tip: press / to filter · type an IP, port, domain or expression · has ~2,000 ready-made filters explained in plain language

FORENSICS WORKBENCH

Forensics

Evidence extracted from the traffic: logins, files, file-share access, indicators of compromise and raw content search.

Credentials | Extracted files | File-share activity | IOC matching | Payload hunt | DHCP | ARP | Content indicators

Credentials seen in the traffic

These were readable by anyone able to capture the network — exactly like this file. Change them.

Reveal secrets | CSV

Time	Protocol	Client	Server	Username	Secret	Context
13.850	HTTP Basic	MacBook-Pro	intranet.corp.local	admin	\$*****!	http://intranet.corp.local/admin
13.927	HTTP form	DESKTOP-ACCT01	example.com	jdoe	P*****d	POST http://example.com/login.php
14.637	FTP	MacBook-Pro	intranet.corp.local	backup	b*****3	clear-text login
14.679	SNMP v2c	intranet.corp.local	192.168.1.250	community	p*****c	SNMP community strings act as passwords and are sent in clear text
14.738	NTLM (SMB)	DESKTOP-ACCT01	intranet.corp.local	CORP\jdoe	NTLM challenge-response (hash, 24 bytes)	workstation DESKTOP-ACCT01
51.539	RDP	203.0.113.9	intranet.corp.local	administrator		RDP connection cookie (username only)

Wireshark 4.6.9

IDS signatures (Snort / Suricata rules)

Load an IDS ruleset in **Updates & feeds**, for example the free *Emerging Threats* open rules, your own rules, or a vendor's Snort-format rules. Then open a capture, go to **IDS signatures** and click **Scan now**. Every payload is matched against every rule in a single pass. Matches appear here and on the Findings page.

Kerberos / Active Directory

Click **Extract Kerberos**, which uses Wireshark. You then see:

- message types and encryption types (RC4 is weak);
- errors;
- the service tickets requested;
- every message.

The AD detectors (Kerberoasting, AS-REP roasting, spraying, enumeration) then run automatically.

Wi-Fi (802.11)

For **monitor-mode** captures, which contain raw Wi-Fi frames, pCapitan lists:

- access points with their security (open, WEP, WPA, WPA2/3) and signal;
- probe requests;
- deauthentication frames;
- active clients.

It flags deauthentication floods, evil-twin access points and open networks. Ordinary captures don't contain Wi-Fi frames, because the operating system has already converted them to Ethernet.

The screenshot displays the pCapitan interface for analyzing a Wi-Fi capture file named 'sample-wifi.pcap'. The interface is divided into a sidebar on the left and a main content area on the right.

Sidebar (Left):

- START**
 - Captures
 - Live capture
- UNDERSTAND**
 - Overview
 - Ask pCapitan
 - Narrated replay
 - Network map
 - World map
 - Traffic flow
 - Timeline
 - Protocols
- DIG IN**
 - Conversations
 - Packets
 - Devices
 - DNS
 - Web (HTTP)
 - Theme

Main Content Area (Right):

WIRELESS
Wi-Fi (802.11)
280 wireless frames decoded. Access points, client devices, probe requests and attack frames.

Access points seen

Network (SSID)	BSSID	Vendor	Security	Signal	Beacons
CorpWiFi	08:11:22:aa:bb:01	CIMSYS Inc	WPA2/3	-45 dBm	31
CorpWiFi	08:11:22:aa:bb:02	CIMSYS Inc	WPA2/3	-60 dBm	30
GuestOpen	08:11:22:cc:dd:83	CIMSYS Inc	open	-55 dBm	30
CorpWiFi	de:ad:be:ef:13:37	Randomized/locally administered	WPA2/3	-40 dBm	25

Frame types

A donut chart showing the distribution of frame types: deauthentication (purple), beacon (yellow), and as (green). The legend indicates 1/4 of the frames are deauthentication.

Probe requests (devices looking for networks)

Network searched	Devices	Frames
CorpWiFi	1	1

Deauth / disassoc frames

From	To	Count
08:11:22:aa:bb:01	a4:83:e7:11:22:33	1
08:11:22:aa:bb:01	ff:ff:ff:ff:ff:ff	1

Most active client devices

Device MAC	Data frames
Nothing here.	

Event timeline

A chronological incident narrative. It covers:

- the first look-up of each name;
- sessions opened;
- credentials and file access;
- DHCP leases;
- new devices;
- alerts.

Filter it by type. It is ideal for writing up "what happened, when".

Performance

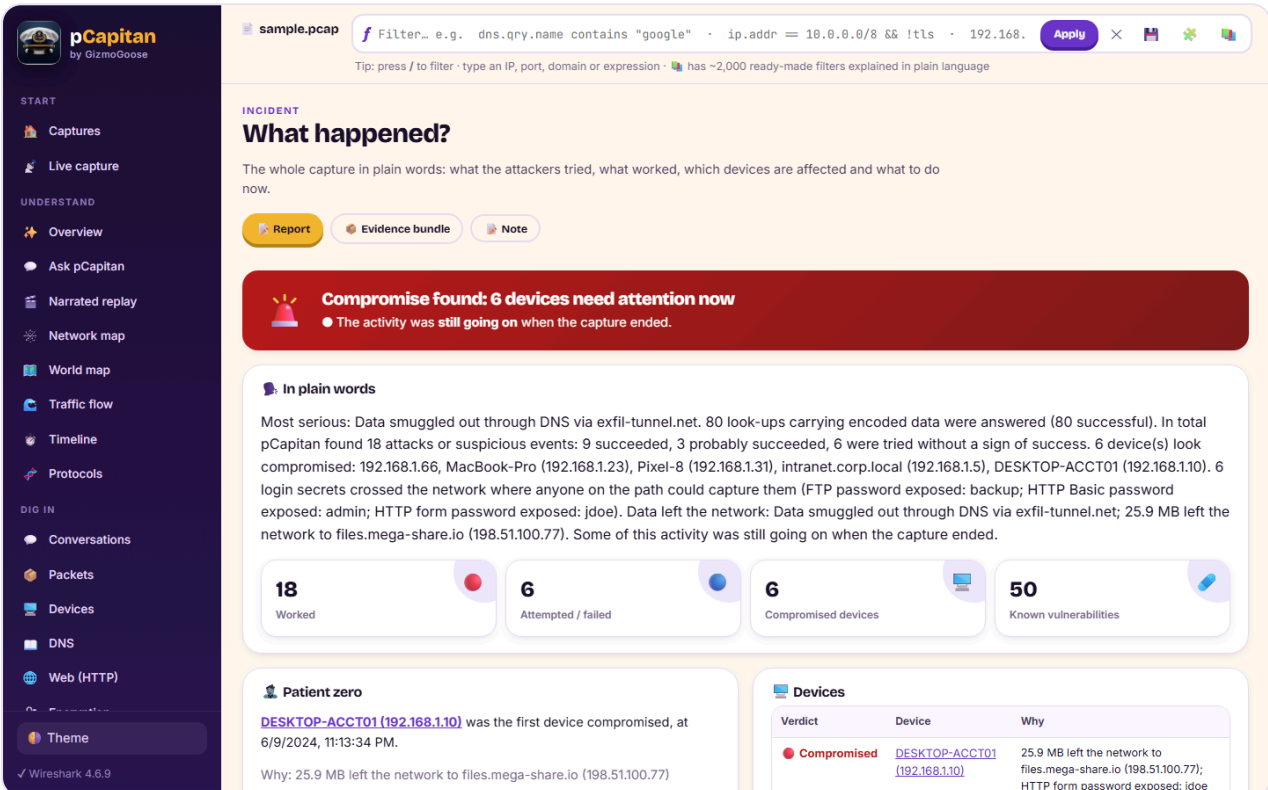
Packet loss, handshake delays, slow servers, slow DNS and slow web responses, each with a verdict in plain language.

Incident response

pCapitan does more than list suspicious traffic. For every attack it checks **whether it worked**, puts the attacks together into **what happened**, matches device software against **known vulnerabilities**, and gives you the **block rules, indicators and evidence** to respond.

What happened?

 **What happened?** (Security group in the sidebar) tells the whole story of a capture in plain words. The **Overview** shows the same summary under the risk banner; click it to open the page.



The screenshot shows the pCapitan interface. The left sidebar contains navigation options: START (Captures, Live capture), UNDERSTAND (Overview, Ask pCapitan, Narrated replay, Network map, World map, Traffic flow, Timeline, Protocols), DIG IN (Conversations, Packets, Devices, DNS, Web (HTTP)), and Theme. The main content area is titled 'INCIDENT What happened?' and includes a filter bar at the top. Below the title, it states: 'The whole capture in plain words: what the attackers tried, what worked, which devices are affected and what to do now.' There are buttons for 'Report', 'Evidence bundle', and 'Note'. A red banner indicates 'Compromise found: 6 devices need attention now' with the note 'The activity was still going on when the capture ended.' The 'In plain words' section provides a detailed summary of the incident. Below this, four statistics are shown: 18 Worked, 6 Attempted / failed, 6 Compromised devices, and 50 Known vulnerabilities. The 'Patient zero' section identifies the first compromised device as DESKTOP-ACCT01 (192.168.1.10) at 6/9/2024, 11:13:34 PM. The 'Devices' table lists the verdict, device name, and the reason for compromise.

Verdict	Device	Why
Compromised	DESKTOP-ACCT01 (192.168.1.10)	25.9 MB left the network to files.mega-share.io (198.51.100.77); HTTP form password exposed: jdoe

The page shows, from top to bottom:

- **Status:** *Compromise found, Attacks worked, Attempted, none worked or No attacks found*, and whether the activity was **still going on** when the capture ended (is it over?).
- **In plain words:** one paragraph starting with the most serious thing that happened, with counts of attacks that worked, attempts, compromised devices and known vulnerabilities.
-  **Patient zero:** the first device that was compromised, when, and why pCapitan thinks so.
-  **Blast radius:** the local devices patient zero talked to *afterwards*. Check these too.
-  **Devices:** a verdict for every device involved:  **Compromised**,  **At risk** or  **No sign of compromise**, with the reasons.
-  **Incidents, step by step:** related findings joined into an attack chain in attack order (reconnaissance, interception, credential attacks, exploitation and execution, lateral movement,

command & control, data theft or impact). Each step has its time, outcome and a **Details** → link.

Did the attack work?

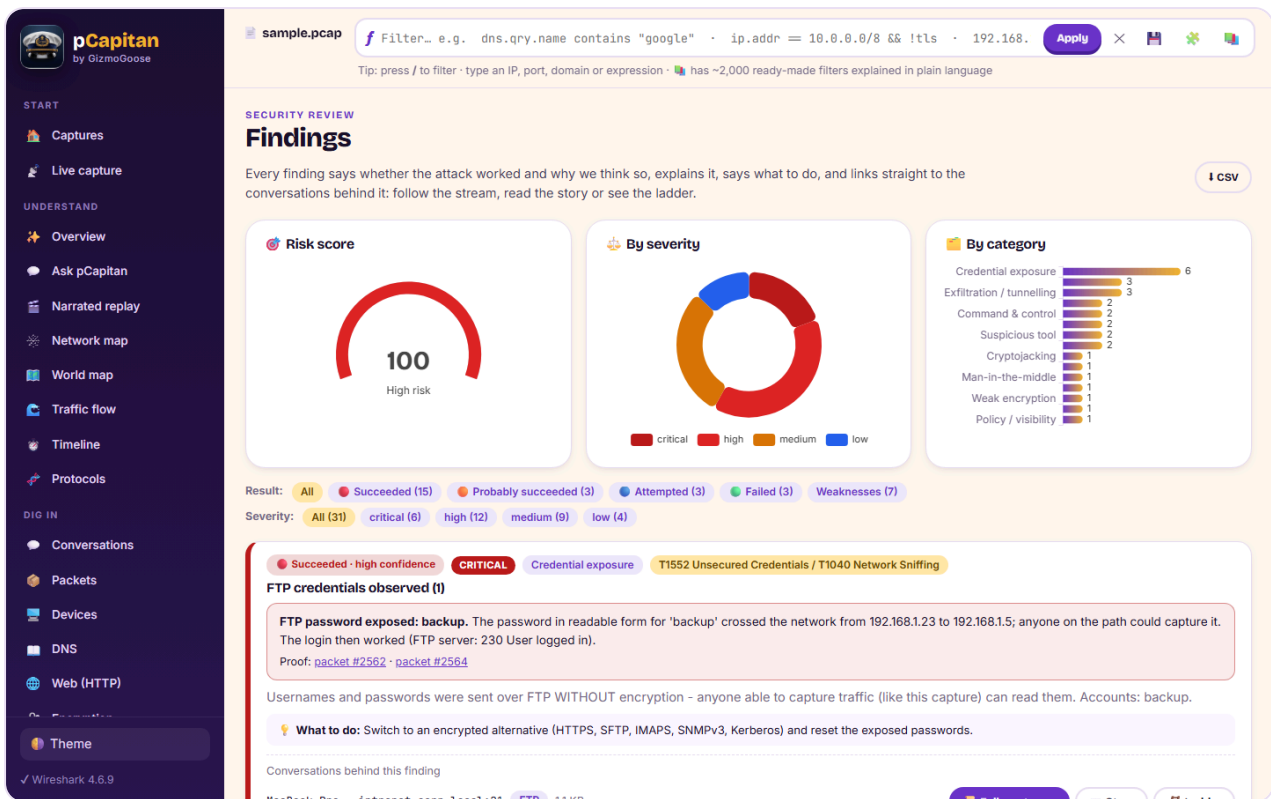
Every finding about an attack now carries an **outcome**, decided from what the traffic shows afterwards:

Outcome	Meaning	Example
● Succeeded	The traffic shows it worked.	A brute force followed by a successful log-in; an FTP password followed by <code>230 Login successful</code> and a download.
● Probably succeeded	Strong signs, but not proof.	An exploit string followed by a new outbound connection from the target.
● Attempted	It was tried; nothing shows it worked.	A port scan that found no open ports.
● Failed	The traffic shows it did not work.	Every log-in rejected; the tool's requests refused.
● Unclear	This capture cannot tell.	The answer was encrypted, or the capture ended too early.

Findings that describe a **weakness** rather than an attack (Telnet in use, a weak TLS version, an expired certificate) have no outcome. Findings that worked are listed first.

Each finding card also shows:

- the **reason** for the outcome, in one sentence;
- the **attacker** and the **victim**;
- the **proof** packets (click one to dissect it);
- the **conversations** involved, each with 📄 **Follow stream** (the bytes exchanged), 📖 **Story** and 📊 **Ladder**;
- 🔍 **Filter to the evidence** and 📝 **Note**.



The device drawer (click any device) shows the same: its verdict, the software it runs, its known vulnerabilities, the findings where it was the attacker or the victim, and its conversations.

An outcome is pCapitan's reading of the traffic, not a certainty. A *Failed* brute force can still have succeeded over a channel the capture did not see. Confirm on the device before you close an incident.

Vulnerabilities

Vulnerabilities works like the application identification of a next-generation firewall. pCapitan reads the software and versions that devices announce in the traffic:

- web server banners (Apache, nginx, IIS, lighttpd, Jetty, Tomcat) and PHP versions;
- SSH banners (OpenSSH, Dropbear, libssh);
- FTP and mail greetings (vsftpd, ProFTPD, Pure-FTPd, FileZilla Server, Exim);
- OpenSSL versions, SMB version 1 (EternalBlue), and operating systems that are out of support.

It matches them against an **offline vulnerability database** built from **NIST NVD**, with **CISA Known Exploited Vulnerabilities (KEV)** and **EPSS** exploit-likelihood scores. Vulnerabilities known to be exploited in the wild are listed first.

EXPOSURE

Vulnerabilities

Like a next-generation firewall: pCapitan reads the software versions devices announce in the traffic and matches them against known vulnerabilities (NIST NVD), flagging the ones criminals actively exploit (CISA KEV) and how likely exploitation is (EPSS).

[Update the database](#)

Vulnerability database of 2026-10-21: 2,179 CVEs for 16 products, 23 known to be exploited. Versions are only visible in unencrypted banners; Linux distributions often fix flaws without changing the version, so those matches are marked "possible".

Known vulnerabilities (50)

Device	Software	CVE	Severity	Exploited	Confidence	What it means
185.199.10.99	nginx 1.18.0	CVE-2023-44487	7.5 HIGH	in the wild	likely	The HTTP/2 protocol allows a denial of service (server resource consumption) because request cancellation can reset many streams quickly, as exploited in the wild in August through October 2023.
intranet.corp.local (192.168.1.5)	nginx 1.18.0	CVE-2023-44487	7.5 HIGH	in the wild	likely	The HTTP/2 protocol allows a denial of service (server resource consumption) because request cancellation can reset many streams quickly, as exploited in the wild in August through October 2023.
example.com (93.184.216.34)	nginx 1.18.0	CVE-2023-44487	7.5 HIGH	in the wild	likely	The HTTP/2 protocol allows a denial of service (server resource consumption) because request cancellation can reset many streams quickly, as exploited in the wild in August through October 2023.
intranet.corp.local (192.168.1.5)	OpenSSH 8.9p1	CVE-2023-28531	9.8 CRITICAL		possible	ssh-add in OpenSSH before 9.3 adds smartcard keys to ssh-agent without the intended per-hop destination constraints. The earliest affected version is 8.9.
intranet.corp.local (192.168.1.5)	OpenSSH 8.9p1	CVE-2023-38408	9.8 CRITICAL		possible	The PKCS#11 feature in ssh-agent in OpenSSH before 9.3p2 has an insufficiently trustworthy search path, leading to remote code execution if an agent is forwarded to an attacker-controlled system. (Code in /usr/lib is not necessarily safe for loading into ssh-agent.) NOTE: this is
intranet.corp.local (192.168.1.5)	OpenSSH 8.9p1	CVE-2024-6387	8.1 HIGH		possible	A security regression (CVE-2006-5051) was discovered in OpenSSH's server (sshd). There is a race condition which can lead sshd to handle some signals in an unsafe manner. An unauthenticated, remote attacker may be able to trigger it

The **Confidence** column says how sure the match is:

- **likely**: the exact version is in a vulnerable range.
- **possible**: a Linux distribution build (for example `OpenSSH_8.9p1 Ubuntu-3ubuntu0.6`). Distributions fix security problems without changing the version number, so check the device's package version.

When an IDS rule that names a CVE fires against a device, the attempt is shown next to the vulnerability (*attacked here*) and listed under **Exploits attempted**: someone tried to exploit it.

A CVE on CISA's list carries the badge **actively exploited**, or **actively exploited · ransomware** when CISA says ransomware campaigns use it. The badge appears wherever pCapitan names such a CVE: in **Exploits attempted**, on IDS alerts and in the reasons for a device's verdict. In the vulnerability table, the same CVEs are marked *in the wild*.

The database ships with pCapitan and can be refreshed in **Updates & feeds** → **Vulnerability database** → **Refresh from NVD**. A refresh downloads from NVD and CISA (a few minutes; set the `NVD_API_KEY` environment variable for a faster download). After a refresh, every capture is matched again the next time you open it.

Respond & notes

Respond & notes turns the findings into action.

The screenshot shows the pCapitan interface with a sidebar on the left containing navigation options like Captures, Live capture, Overview, and more. The main area is titled 'RESPOND' and 'Contain, document, hand over'. It includes a filter bar at the top, a 'Block rules' section with various firewall options and a generated PowerShell script, an 'Indicators of compromise (5)' table, and an 'Allow rules' section for locking down servers.

Filter: e.g. dns.qry.name contains "google" · ip.addr == 10.0.0.0/8 && !tls · 192.168. [Apply]

Tip: press / to filter · type an IP, port, domain or expression · has ~2,000 ready-made filters explained in plain language

RESPOND

Contain, document, hand over

Block what attacked you, keep your findings with the capture, and hand over a complete evidence bundle.

[Download evidence bundle](#)

Block rules

For the attackers' addresses and domains below. Review before applying.

Windows Firewall (PowerShell)

Linux iptables

Cisco ASA / FTD (FlexConfig-style)

FortiGate CLI

hosts file (domains)

DNS RPZ zone (domains)

Plain list

```
# Generated by pCapitan on 2026-10-21 19:07 from confirmed and attempted
# Run in an elevated PowerShell.
New-NetFirewallRule -DisplayName "pCapitan block 185.199.10.99 (out)" -D
New-NetFirewallRule -DisplayName "pCapitan block 185.199.10.99 (in)" -D
New-NetFirewallRule -DisplayName "pCapitan block 45.33.32.156 (out)" -D
New-NetFirewallRule -DisplayName "pCapitan block 45.33.32.156 (in)" -D
New-NetFirewallRule -DisplayName "pCapitan block 51.15.65.1 (out)" -Dir
New-NetFirewallRule -DisplayName "pCapitan block 51.15.65.1 (in)" -Dire
# Domains cannot be blocked by Windows Firewall: use the hosts or RPZ e
```

Copy

Download

Indicators of compromise (5)

Addresses, domains, URLs and file hashes from attacks that worked or were attempted.

Type	Value	Result	Seen in
ip	45.33.32.156	Succeeded	Beaconing: 192.168.1.23 contacts update-check.xyz every ~30s
domain	exfil-tunnel.net	Succeeded	Possible DNS tunneling via exfil-tunnel.net
ip	185.199.10.99	Succeeded	Executable/script downloaded over plain HTTP (1)
ip	51.15.65.1	Probably succeeded	Connections to port 3333 (crypto-mining pool)

CSV

STIX 2.1

Allow rules: lock a server down to what it really uses

Built from the conversations that worked in this capture: each server's services and who used them, then deny the rest. Capture long enough (a full day or week) so nothing legitimate is missed.

All local servers

FortiGate CLI

Cisco ASA

Palo Alto (set commands)

MikroTik RouterOS

Linux nftables

Linux iptables

Windows Firewall (PowerShell)

CSV (review in Excel)

- 🚫 **Block rules** for the external attackers' addresses and domains, ready to paste into **Windows Firewall (PowerShell)**, **iptables**, **Cisco ASA / FTD**, **FortiGate**, a **hosts** file, a **DNS RPZ** zone, or a plain list. 📄 **Copy** or ⬇️ **Download**. Review them before you apply them.
- ✅ **Allow rules:** lock a server down to what it really uses (see below).
- 🔗 **Indicators of compromise:** external IPs, domains, URLs and file hashes, with the outcome of the attack each came from. Export as **CSV** or **STIX 2.1** for a SIEM, a threat-intelligence platform or a colleague. Local addresses and shared hosting domains are never included.
- 📝 **Analyst notes:** click 📝 **Note** on a finding, conversation, device or packet, choose a tag (*confirmed, false positive, benign, to do or note*) and write what you found. Notes are kept with the capture and appear in the report.
- 📦 **Evidence bundle:** one .zip with the report, the indicators (CSV and STIX), your notes, the full assessment, the packets of the incident as a .pcap, and **CHAIN-OF-CUSTODY.txt** listing the SHA-256 of the original capture and of every file in the bundle. The report in the bundle carries your branding (see *Reports and exports*).

✅ Allow rules

Block rules stop what attacked you. **Allow rules** go the other way: they allow what each local server really used in this capture, and deny the rest.

pCapitan looks at the conversations that worked (TCP and UDP, with an answer from the server) and lists, for each local server, its services (protocol and port) and the clients that used them. Choose **All local servers** or one server. With one server, pCapitan also lists what that server itself connected out to (→ **out**).

Choose a format: **FortiGate CLI**, **Cisco ASA**, **Palo Alto** (set commands), **MikroTik RouterOS** (IPv4 and IPv6 menus), **Linux nftables**, **Linux iptables**, **Windows Firewall** (PowerShell) or **CSV** (to review in Excel).  **Copy** or **↓ Download**. The rules end with a deny for the rest of each server's traffic.

- **Attackers and indicators are never allowed.** Addresses that attacked, and the indicators of compromise, are left out of the clients. The table shows them as *left out*, and the rules' header names them.
- Clients are written as single addresses, or as a /24 network when four or more share one and no left-out address is in it. When a service had more than 25 clients, or any internet client, the client is written as *any*; the left-out addresses then get their own **deny** rules placed before it, so *any* never lets them back in.
- The **Windows Firewall** rules are for inbound traffic only and run on the server itself. Block rules for left-out addresses come first (Windows applies block rules before allow rules). Windows already blocks other inbound traffic by default; the command to restore that default is included as a comment.

Only what the capture saw is allowed. Capture long enough (a full day or week), and add anything the capture did not cover (monthly jobs, backups, failover) before you apply the rules.

The incident report

 **Report & export** → **Open report** now starts with what matters:

1. Executive summary: status, what happened, and **Do now**
2. Attacks that worked
3. Affected devices, patient zero and blast radius
4. Known vulnerabilities
5. Data that left the network
6. Exposed passwords
7. Timeline
8. Action plan: Now, Today, This week
9. Indicators to block
10. Attempts and weaknesses
11. Analyst notes

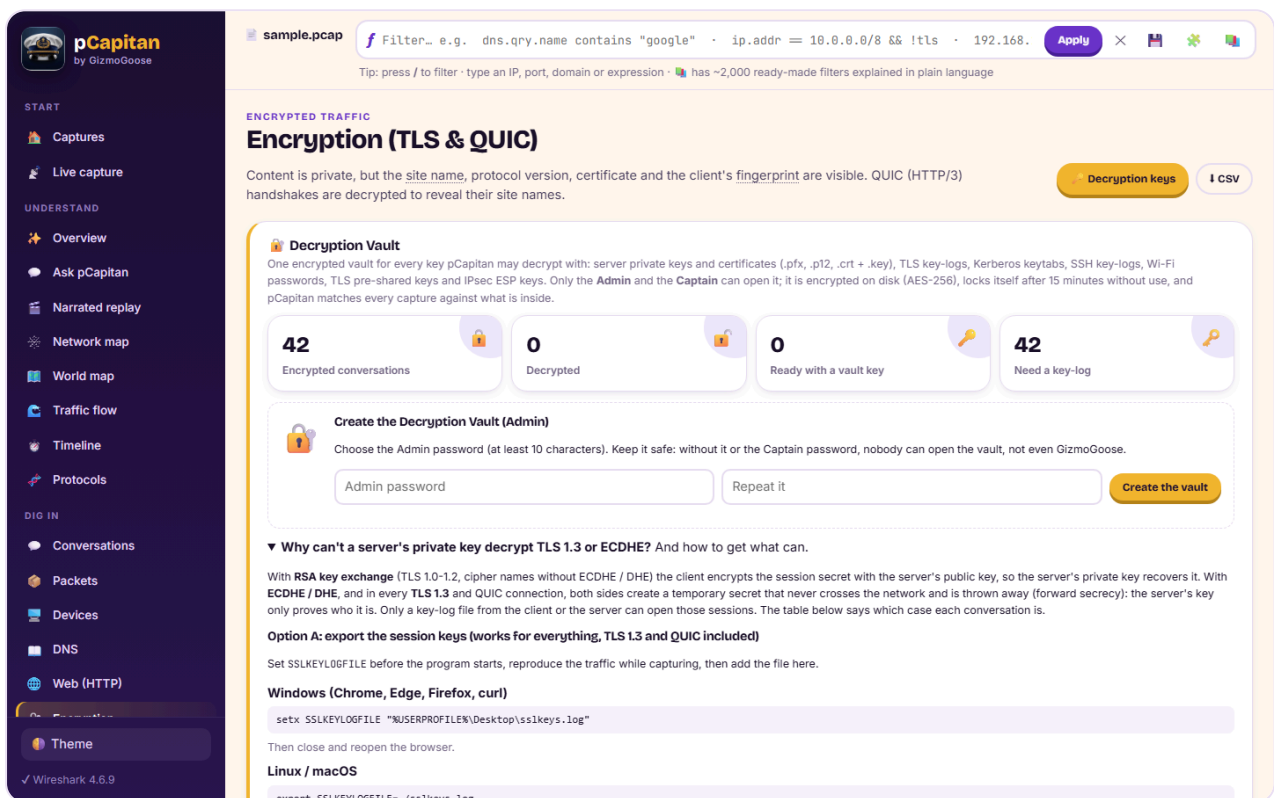
The traffic statistics follow as appendices.

Decrypting traffic: the Decryption Vault

Most traffic today is encrypted. pCapitan can read it when you have the right key. Every key, certificate, key-log and password goes into one place, the  **Decryption Vault**. pCapitan matches each capture against everything in the vault, tells you which conversations each key opens and what is still missing, and decrypts with one click.

Where to find it

Open  **Encryption** (Dig in group in the sidebar) and click  **Decryption keys** at the top, or scroll down to the  **Decryption Vault** card. From any encrypted conversation,  **Add a key to decrypt** it takes you to the same place.



What the vault is

The vault is one encrypted file on this computer. It is never sent anywhere.

- A random 256-bit **vault key** encrypts the contents with **AES-256-GCM**.
- The vault key is **sealed** separately under each role's password: the password is stretched with **scrypt** ($N = 2^{15}$, $r = 8$, $p = 1$, a random 16-byte salt per role) into a key that seals the vault key, again with AES-256-GCM.
- Keys are never written to disk readable. When the Wireshark engine needs them, pCapitan writes temporary files that are deleted as soon as the engine finishes.

- Nobody can open the vault without the Admin or the Captain password, not even GizmoGoose. There is no recovery: if both passwords are lost, the contents are lost.

Admin and Captain

The vault has two roles. Each has its own password of at least **10 characters**.

Action	👑 Admin	👤 Captain
Create the vault	✓	—
Unlock and lock the vault	✓	✓
Add keys, and decrypt with them	✓	✓
Remove keys	✓	—
Set or change the Captain password	✓	—
Remove the Captain	✓	—
Change the Admin password	✓	—

1. The Admin creates the vault: **Create the vault**, with the password typed twice. The vault opens as Admin.
2. Under 👑 **Admin: passwords**, the Admin can **Set the Captain password**, change it, **Remove the Captain**, or **Change the Admin password**. Changing a password does not re-encrypt the keys.
3. Next time, either person types their password in 🗝️ **Unlock**. pCapitan works out the role from the password.

Every key in the list shows **added by** Admin or Captain.

Locking

- The vault **locks itself after 15 minutes** without use. Each action restarts the 15 minutes. The bar shows **locks itself in ... min**.
- 🗝️ **Lock now** locks it at once.
- The unlocked vault exists only in memory, so it is locked whenever pCapitan stops.
- After **5 wrong passwords** in a row, pCapitan makes you wait **30 seconds**, then 30 seconds longer after each further wrong password, up to **5 minutes**. A correct password clears the count.

While the vault is locked, conversations already decrypted stay marked 🗝️ **Decrypted**, but pCapitan cannot match keys, decrypt again or show the decrypted content.

What can go in

Use **Choose files** for key files (you can choose several at once), and the boxes beside it for passwords and pre-shared keys. pCapitan recognises each file by its content, not by its name.

What you add	Where	What it decrypts
 .pfx / .p12 (key and certificate)	Choose files, with its password in File password	TLS sessions to that server that used RSA key exchange
 Private key (.key, .pem, .der), with or without a password	Choose files	The same
 Certificate (.crt, .cer, .pem)	Choose files	Nothing on its own: it holds only the public key. pCapitan shows which server uses it
 PEM bundle (key and certificates in one file)	Choose files	As a private key; pCapitan warns if the certificate does not belong to the key
 TLS key-log (SSLKEYLOGFILE)	Choose files	Every TLS version and cipher, TLS 1.3 and QUIC included
 TLS pre-shared key (hex)	TLS pre-shared key (IoT, MQTT)	TLS-PSK sessions
 Kerberos keytab	Choose files	Kerberos tickets and what they protect
 SSH key-log (Wireshark 4.x format)	Choose files	SSH sessions
 Wi-Fi password and network name	Wi-Fi (WPA / WPA2-Personal)	WPA/WPA2-Personal traffic, into a new capture
 IPsec ESP keys	 IPsec ESP tunnels in this capture	ESP tunnels, into a new capture

For a **.crt** and its **.key**, add both files together. A private key that is not RSA (for example an EC key) is accepted, but pCapitan tells you at once that it cannot decrypt TLS: servers with such certificates always use (EC)DHE key exchange.

TLS, SSH and Kerberos decryption uses the Wireshark engine. If Wireshark is not installed, pCapitan asks you to install it (it is free). Wi-Fi and IPsec are decrypted by pCapitan itself.

How keys are matched to the capture

Under **In the vault, matched against this capture**, every key says what it does in the capture you have open:

- A **private key or certificate** is matched to a server by its public key: pCapitan compares it with every server certificate in the capture.
- A **TLS key-log** is matched session by session: a key-log line belongs to a session when its client random is in that session's ClientHello.
- A **Wi-Fi password** matches when its network name (SSID) is in the capture.
- An **ESP key** matches when its source, destination and SPI are in the capture.
- A **keytab**, **SSH key-log** or **TLS pre-shared key** matches when the capture has Kerberos, SSH or TLS traffic.
- Every keytab and every SSH key-log in the vault is used together. The engine takes one TLS pre-shared key at a time: the most recently added one.
- The **File password** is used only by the files that need one, so you can add a .pfx and a plain .key in one go. Key files can be up to 256 MB.

State	Meaning
✅ Decrypted	The key matched, and conversations are now readable.
🔑 Ready	The key matched. Click 🔑 Decrypt this capture.
⚠️ Blocked by forward secrecy	The private key is the right one for the server, but every session used ECDHE / DHE or TLS 1.3. A key-log is needed.
📄 Certificate only	The certificate belongs to a server here, but cannot decrypt. Add its private key or a key-log.
❌ No match	Nothing in this capture belongs to this key.

A certificate is sent only in the first handshake of a connection. If the capture started after it, or the sessions were resumed, a correct key can show ❌ **No match**.

Every encrypted conversation and what opens it

The table under the vault lists every TLS conversation, its version, cipher and **Key exchange** (*RSA: a private key works or Forward secrecy: key-log needed*), and in **What opens it** one of these:

Status	Meaning	What to do
decrypted	Already decrypted, with what was inside (for example HTTP).	🔑 Follow to read it.
ready	RSA key exchange, and a private key in the vault belongs to this server.	Click 🔑 Decrypt this capture.
needs-key	RSA key exchange, but no matching key yet.	Add the server's private key, or a key-log.
try-keylog	Forward secrecy or TLS 1.3, and the vault has a key-log. pCapitan will try it.	Click 🔑 Decrypt this capture.
needs-keylog	Forward secrecy or TLS 1.3 (or an unknown key exchange), and no key-log in the vault.	Get a key-log from the client or server (below).

The counters at the top show **Encrypted conversations**, **Decrypted**, **Ready with a vault key** (ready and try-keylog) and **Need a key-log**.

RSA key exchange and forward secrecy

With **RSA key exchange** (TLS 1.2 and older, cipher names without ECDHE or DHE), the client encrypts the session secret with the server's public key. The server's private key recovers it, so a **.pfx** or **.key** decrypts these sessions.

With **ECDHE / DHE**, and in every **TLS 1.3** and **QUIC** connection, both sides create a temporary secret that never crosses the network and is then thrown away. This is **forward secrecy**. The server's key only proves who the server is; it cannot recover the secret. This is by design, not a fault in pCapitan.

For these sessions you need a **key-log file**: the program that made the connection writes its session secrets to it.

- **Windows (Chrome, Edge, Firefox, curl):** `setx SSLKEYLOGFILE "%USERPROFILE%\Desktop\sslkeys.log"` , then close and reopen the browser.
- **Linux / macOS:** `export SSLKEYLOGFILE=~/.sslkeys.log` , then start the program from that terminal.
- **Node.js:** `node --tls-keylog=sslkeys.log app.js` . **Python 3.8+:** set `SSLKEYLOGFILE` before starting.

Set it before the program starts, capture while you reproduce the traffic, then add the file to the vault. The card **Why can't a server's private key decrypt TLS 1.3 or ECDHE?** has these recipes, and opens by itself when a key is blocked by forward secrecy.

The card also shows how to make a **test** server use RSA key exchange so its private key works. This turns off forward secrecy. Do it only on a lab server, and put it back afterwards.

The Encryption column

The **Conversations** and **Packets** lists have an **Encryption** column, and the conversation drawer shows the same badge:

-  **Encrypted:** the content is encrypted (TLS, SSH, QUIC, RDP, VPNs and similar).
-  **Decrypted:** a key from the vault has opened it.
- empty: the content is not encrypted. In **Packets**, packets without a payload have no badge.

Adding a key does not change the column by itself. Click  **Decrypt this capture:** pCapitan decrypts in the background, and the matching conversations change from  to  . Web requests and logins found inside are added to **Web (HTTP)** and **Forensics**, marked *[decrypted]*, and the security checks run again on what was revealed.

Each  **Decrypt this capture** starts again from the keys in the vault now. If you removed a key, the conversations it opened go back to  **Encrypted**.

Wi-Fi and IPsec: a new decrypted capture

Wi-Fi and IPsec decryption does not change the capture you have open. pCapitan writes a **new capture** with the decrypted traffic, so every view, check and report works on what was inside. When it is done, an **Open the decrypted capture** button appears for each new capture. They are named after the original, with **(Wi-Fi decrypted:)** or **(IPsec decrypted)**. Only the Wi-Fi networks that appear in the capture are tried.

Wi-Fi (WPA / WPA2-Personal)

Enter the **Network name (SSID)** and the **Password** (8 to 63 characters) or the 64-character hexadecimal **PSK**.

- The capture must be taken in **monitor mode** (802.11 frames). A capture of a wired link, or Wi-Fi in normal mode, has nothing to decrypt.
- It must contain the **4-way handshake** of each device. Capture while the device joins the network, or make it re-join.
- pCapitan checks the password against the handshake, so a wrong password or network name is reported, not guessed.
- Broadcast traffic is decrypted with the group key from the handshake.
- Only **CCMP** (AES) is supported. **WPA3-SAE**, **WPA2-Enterprise** and **TKIP** are not decrypted.

IPsec ESP

When the capture has ESP traffic,  **IPsec ESP tunnels in this capture** lists each tunnel by source, destination, **SPI** and packet count. For each tunnel, choose the algorithms, paste the keys from the router or VPN gateway (for example `show crypto ipsec sa`, or strongSwan `swanctl --list-sas --raw`) and click **Add**. Each direction of a tunnel has its own SPI and keys.

Encryption	Key length
AES-CBC	16, 24 or 32 bytes
AES-GCM (key + 4-byte salt)	20, 28 or 36 bytes
3DES-CBC	24 bytes
NULL (no encryption)	none

Integrity: **HMAC-SHA1-96**, **HMAC-SHA256-128**, **HMAC-SHA384-192**, **HMAC-SHA512-256**, **HMAC-MD5-96**, or **none** (only with AES-GCM, which has its own check, or NULL). Keys are hexadecimal; `0x`, spaces and colons are allowed.

pCapitan checks every packet's integrity code, so a wrong key is reported instead of producing rubbish. Tunnel mode and transport mode are supported, and so is **NAT traversal** (ESP inside UDP port 4500).

Following decrypted streams

After decryption, the conversation drawer's button becomes  **Follow stream (decrypted)**, and the vault table has  **Follow** for each decrypted conversation. It shows the plain content, client and server in turn, as  **Decrypted content**. Characters that cannot be printed are shown as dots; content that is mostly binary is shown as a hex dump. It works for TLS, and for decrypted SSH, QUIC and Kerberos data too.

For a conversation that is still encrypted,  **Follow stream** shows  **Encrypted content (add a key to read it)**.

The decrypted content is produced again from the vault each time, so the vault must be unlocked to follow a decrypted stream.

Good practice

Keys are sensitive. A server's private key or a key-log lets anyone who has it read the traffic, and a Wi-Fi password opens the whole network.

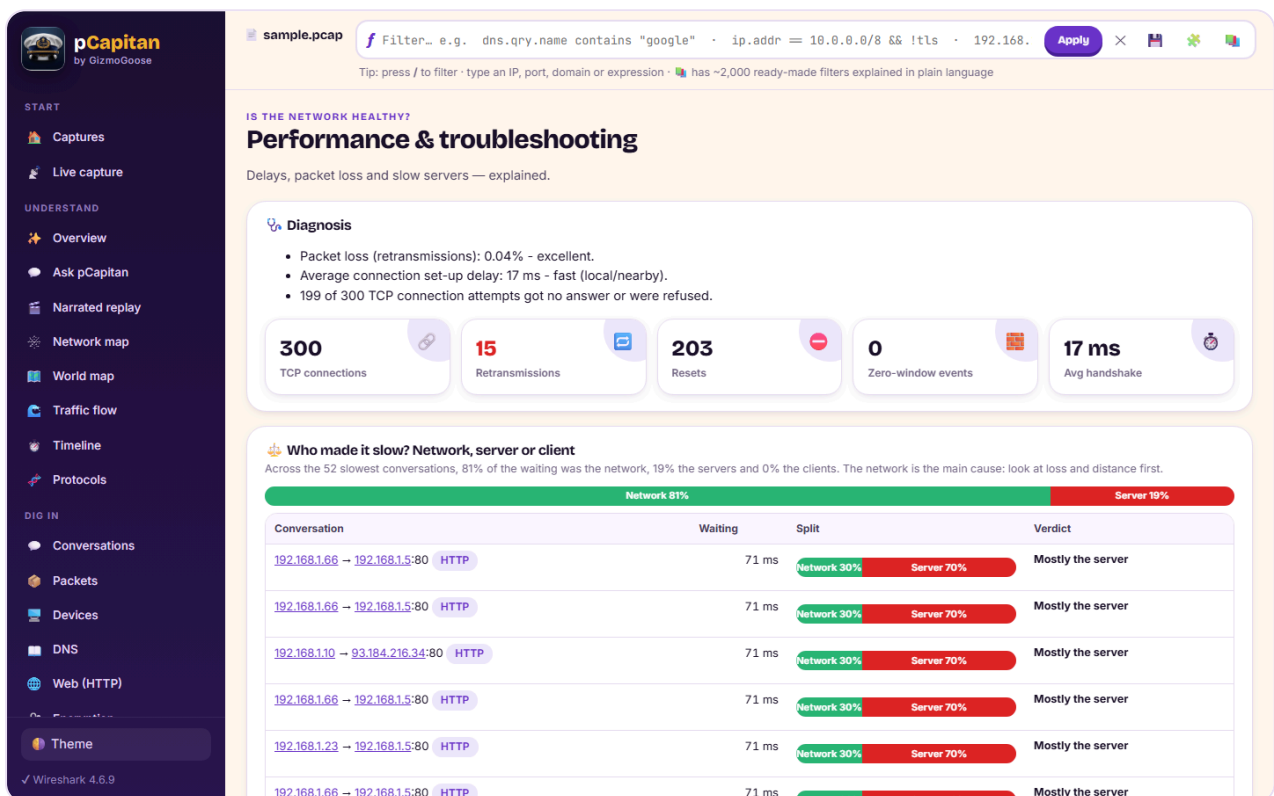
- Give the Captain password only to the person who needs it, and remove the Captain when they no longer do.
-  **Lock now** when you step away; do not rely on the 15 minutes.
- When the case is closed, the Admin removes its keys (X on each key).
- Prefer a key-log from a test machine to a production server's private key.

Why is it slow? Blame, compare, calls and logins

When users say "the network is slow", pCapitan tells you **who is to blame**: the network, the server or the client. It can **compare two captures** to show what changed, or what a firewall did to the traffic between its two sides. It also explains **VoIP calls** and their audio, and **network logins and routing sessions** (RADIUS, 802.1X, PPPoE and BGP).

Who made it slow?

 **Performance** (Dig in group in the sidebar) has a card  **Who made it slow? Network, server or client**. It times every request and answer in the TCP conversations that lasted longest and splits the waiting into three parts.



The card shows:

- **one sentence** for the whole capture, for example "Across the 25 slowest conversations, 70% of the waiting was the network...";
- **the bar** for the whole capture;
- **a table** of the conversations that waited longest, each with its **Waiting** time, its own bar and a **Verdict**. Click a row to open the conversation.

The same card appears in every TCP conversation (click a conversation anywhere in the app), above  **The story, step by step**. You can also ask  **Ask pCapitan** "why is it slow?".

How the time is split

Part	What is counted	What it means
Network	One round trip for every request and answer, plus the time TCP waited to re-send lost packets, plus the wait for unanswered connection requests (SYN).	Distance, congestion or packet loss.
Server	The server's think time : from the moment the request arrived until the server started to answer. Also the time the server's receive buffer was full (zero window).	A slow application, database or server CPU.
Client	The time the client's receive buffer was full (zero window), and repeated Nagle / delayed-ACK stalls on small writes.	A busy or slow computer receiving the data, or an application setting.

How each part is measured:

- **Round trip**: from the handshake. The SYN to SYN-ACK time is the leg to the server and back; the SYN-ACK to ACK time is the leg to the client and back. When one leg is under 1 ms, pCapitan says where the capture was taken: **on the client**, **on the server** or **between the client and the server**.
- **Think time**: when the server first acknowledges the request with an empty packet, think time runs from that acknowledgement to the first byte of the answer. Otherwise it is the gap after the request minus one round trip. Answers that come more than 30 seconds later are not counted: they are pushed messages or long polls.
- **Retransmissions**: the time from the first copy of a segment to its re-sent copy (at most 30 seconds each).
- **Nagle / delayed-ACK**: small client writes (under 536 bytes) that waited about 40 ms or about 200 ms for the server's acknowledgement. They are counted only when they happen three times or more.

Reading the bar

The bar has three parts, **Network**, **Server** and **Client**, each a percentage of the waiting. A conversation with no waiting shows **No waiting**. Hover a part for its exact value. The **Verdict** gives the answer in one line:

Verdict	When
Fast	Less than 50 ms of waiting in total.
The network is innocent: the server is slow	Server over 70%, round trip under 25 ms, loss under 0.5%.
Packet loss on the network	Network over 60% and more than 2% of the packets re-sent.
Distance: every round trip is long	Network over 60%, round trip over 180 ms, loss under 0.5%: a chatty application over a long link.
The client could not keep up	Client over 40% with a full receive buffer on the client.
Nagle and delayed-ACK stalls	Small writes keep stalling. Turn on <code>TCP_NODELAY</code> in the application.
Firewall drop or overloaded server	The client had to send its connection request two or more times.
Mostly the network / server / client	None of the above: the largest part.

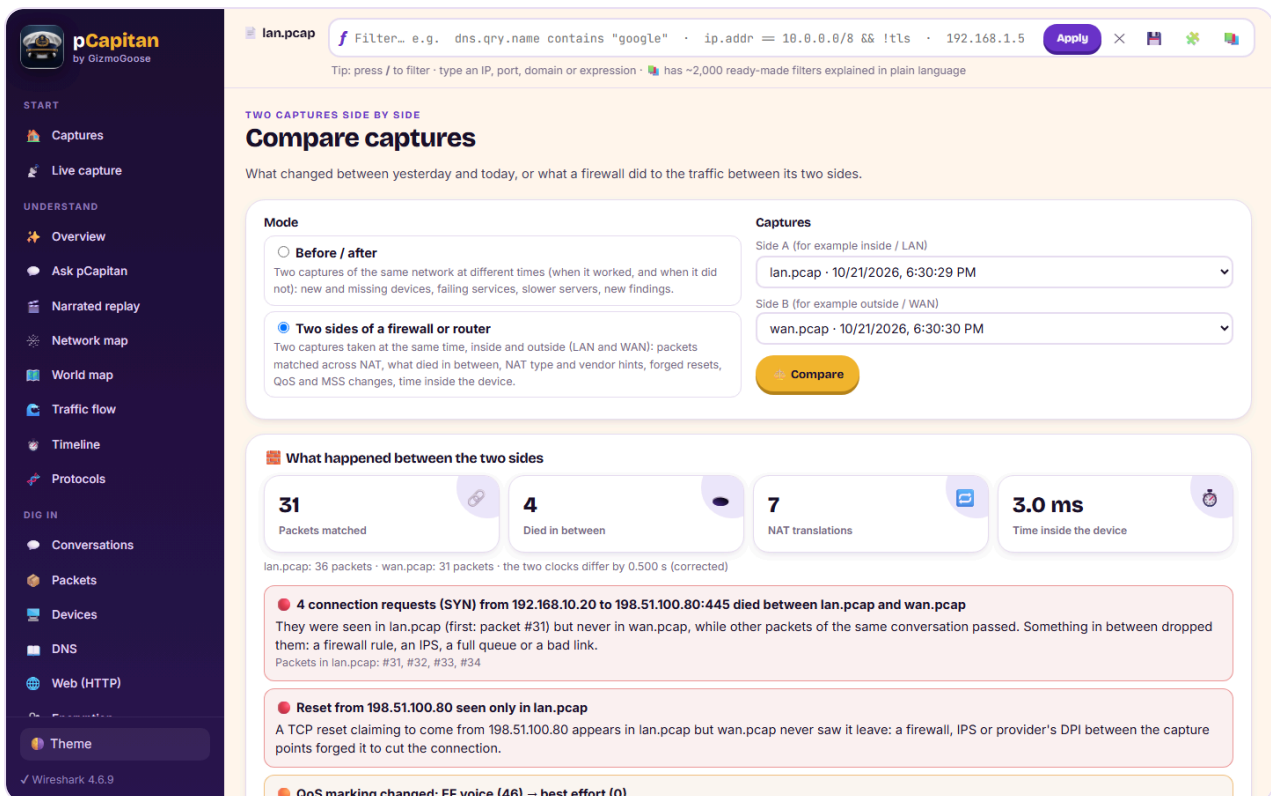
The conversation card adds the details: the number of round trips and retransmissions, the average and longest think time with a link to the packet, and how often each side's buffer was full.

Interactive and message-based services (SSH, Telnet, RDP, VNC, SIP, BGP, MQTT, VPNs and others) are left out of the table, and their conversation card says **Not a request-and-answer conversation**. Their pauses are people typing or keep-alives, not waiting.

The round trip comes from the handshake. If the capture started after a connection was opened, there is no handshake: pCapitan then estimates the round trip from the quickest answers in the conversation and says *estimated*, so the network share is approximate.

Compare captures

 **Compare captures** (Tools group) puts two captures side by side. Open both captures first, choose a **Mode**, choose the two captures and click  **Compare**. Both captures must be fully analysed.



Each result is a card: ● a problem, ● a warning, ● information, ● good news. Problems come first. Buttons at the bottom open either capture.

Before / after

Use this for two captures of the same network at different times: yesterday when it worked, and today when it does not. ● **What changed** can show:

- **new local devices** and **local devices missing**;
- a **new DNS server** answering names (a changed DHCP setting, a VPN, or someone redirecting look-ups);
- a service that worked before and now fails: **refused (reset)**, **not answered** or **reset before answering**;
- **services answering now** that did not before, on local devices;
- servers that **take longer to connect** (handshake at least 50% and 20 ms slower) and **websites** that answer at least 50% and 100 ms slower;
- **packet loss up** (at least twice as much, and over 1%);
- names that **now fail to resolve**;
- **new findings** and **findings gone** (critical, high and medium only);
- a **traffic mix** that changed by 10 points or more.

If nothing differs, the result is **No meaningful difference**.

Two sides of a firewall or router

Use this for two captures taken **at the same time** on both sides of a device: inside and outside (LAN and WAN) of a firewall, a NAT router or a load balancer. Choose the inside as **Side A** and the outside as **Side B**.

Matching packets across NAT. The device changes addresses and ports, so pCapitan matches each packet by what the device leaves alone: the content and TCP flags. Empty TCP segments are matched by their flags, window and timestamp option, because some firewalls change the sequence numbers. ICMP is matched without its identifier.

Clock calibration. The two capture computers rarely have the same time. A SYN and its SYN-ACK cross the device in opposite directions, so their two time differences give the clock difference and the time inside the device. Without handshakes, pCapitan separates the two directions from the time differences themselves. The line under the counters says how much the clocks differ; this is corrected.

The counters at the top:

Counter	Meaning
Packets matched	Packets seen in both captures.
Died in between	Packets of a conversation that does cross, seen leaving one side but never arriving on the other (requests from A to B, and replies from B to A). Something in between dropped them: a firewall rule, an IPS, a full queue or a bad link.
NAT translations	Conversations whose address or port changed on the way.
Time inside the device	How long packets spent between the two capture points.

The cards then explain:

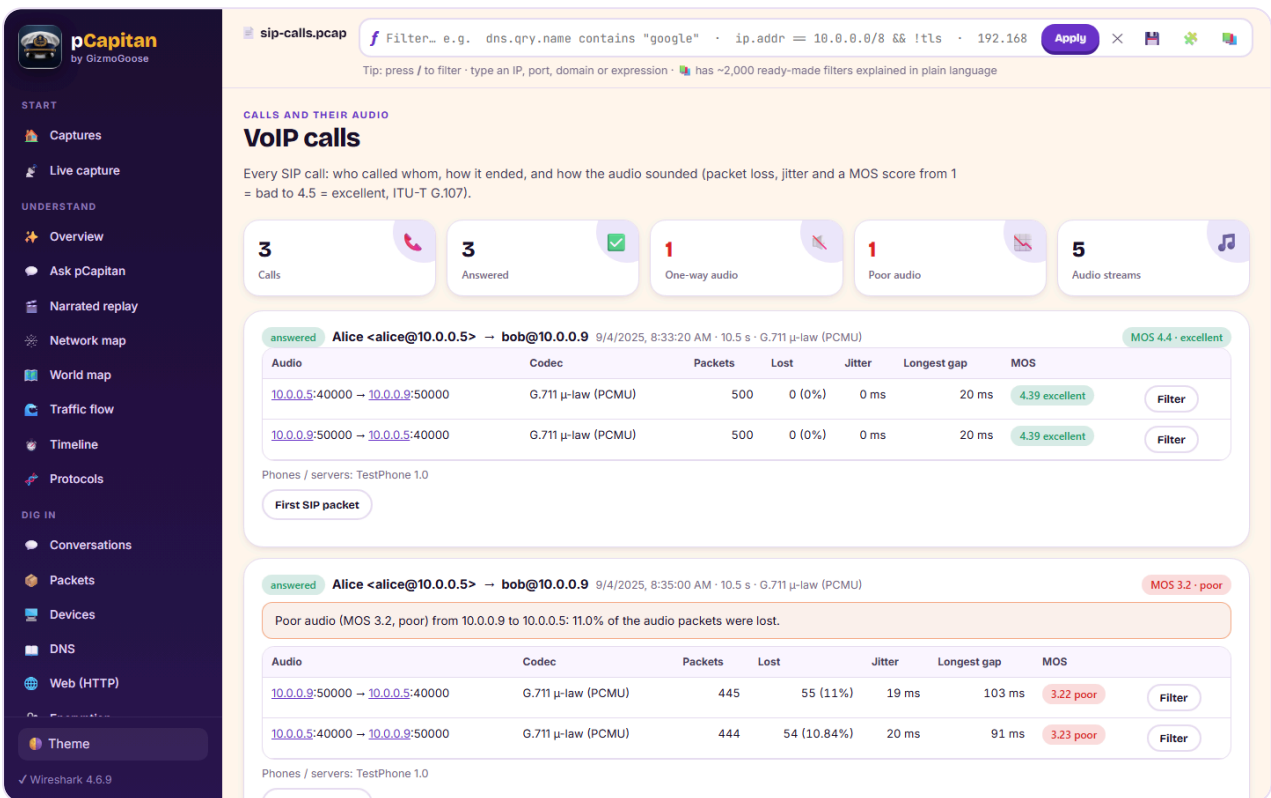
- **...died between A and B** (or **between B and A** for lost replies), with the packet numbers; when they are mostly large packets with *Don't Fragment* set while small ones pass, an **MTU black hole**;
- **Address translation (NAT)**: inside → outside for each conversation, and the **NAT type**: *cone* (one outside port for every destination; peer-to-peer and VoIP work) or *symmetric* (a new outside port for each destination), how ports are allocated, and which kinds of firewall behave that way;
- **TCP sequence numbers are rewritten in between** (Cisco ASA does this by default);
- **Reset from ... seen only in ...: a forged reset**. In a conversation that crosses the device, one side shows a TCP reset claiming to come from the other side, which that side never sent: a firewall, IPS or provider's DPI cut the connection. Resets in conversations that do not cross the device are not reported;
- **QoS marking changed**, for example EF voice (46) → best effort (0): voice and video lose their priority;
- **MSS clamped** from one value to another, and the number of **router hops** between the capture points;

- **Time inside the device.** When packets with content wait over 25 ms but empty acknowledgements under 2 ms, the device's content inspection (DPI, IPS or antivirus) is slow.

If the two captures share no traffic, the result is **No packet appears in both captures**: they were taken at different times or on different links.

VoIP calls

 **VoIP calls** (Dig in group) lists every SIP call and says how its audio sounded.



VoIP calls

Every SIP call: who called whom, how it ended, and how the audio sounded (packet loss, jitter and a MOS score from 1 = bad to 4.5 = excellent, ITU-T G.107).

3 Calls, 3 Answered, 1 One-way audio, 1 Poor audio, 5 Audio streams

answered Alice <alice@10.0.0.5> → bob@10.0.0.9 9/4/2025, 8:33:20 AM · 10.5 s · G.711 μ-law (PCMU) **MOS 4.4 · excellent**

Audio	Codec	Packets	Lost	Jitter	Longest gap	MOS
10.0.0.9:40000 → 10.0.0.5:50000	G.711 μ-law (PCMU)	500	0 (0%)	0 ms	20 ms	4.39 excellent
10.0.0.9:50000 → 10.0.0.5:40000	G.711 μ-law (PCMU)	500	0 (0%)	0 ms	20 ms	4.39 excellent

Phones / servers: TestPhone 1.0

First SIP packet

answered Alice <alice@10.0.0.5> → bob@10.0.0.9 9/4/2025, 8:35:00 AM · 10.5 s · G.711 μ-law (PCMU) **MOS 3.2 · poor**

Poor audio (MOS 3.2, poor) from 10.0.0.9 to 10.0.0.5: 11.0% of the audio packets were lost.

Audio	Codec	Packets	Lost	Jitter	Longest gap	MOS
10.0.0.9:50000 → 10.0.0.5:40000	G.711 μ-law (PCMU)	445	55 (11%)	19 ms	103 ms	3.22 poor
10.0.0.5:40000 → 10.0.0.9:50000	G.711 μ-law (PCMU)	444	54 (10.84%)	20 ms	91 ms	3.23 poor

Phones / servers: TestPhone 1.0

The counters at the top show **Calls**, **Answered**, **One-way audio**, **Poor audio** and **Audio streams**. Each call card shows:

- the **outcome**: *answered, busy, cancelled, declined, no answer (timeout), number not found, codec not acceptable* and others, with the SIP code;
- **caller** → **callee**, the start time, the duration and the **codecs**;
- the worst **MOS** of the call and its rating;
- the **problems** in plain words;
- every **RTP stream** of the call: **Codec**, **Packets**, **Lost**, **Jitter**, **Longest gap** and **MOS**, with **Filter** to see its packets;
- the phones and servers (user agents) and a link to the **First SIP packet**.

Streams are tied to a call by the addresses and ports in the call set-up (SDP), within the call's own time.

How the audio is measured

- **Packet loss** comes from the RTP sequence numbers: how many were expected and how many arrived.
- **Jitter** is the RFC 3550 interarrival jitter: how unevenly the packets arrive, in milliseconds.
- **MOS** is calculated with the ITU-T G.107 **E-model**, from the codec, the packet loss and an estimated delay (40 ms plus twice the jitter). One capture cannot measure the real one-way delay. The scale runs from 1 (bad) to 4.5 (excellent). A clean G.711 call scores about 4.4; a clean G.729 call about 4.1, because the codec itself costs quality.

MOS	Rating
4.3 and above	 excellent
4.0 to 4.3	 good
3.6 to 4.0	 fair
3.1 to 3.6	 poor
below 3.1	 bad

A call counts as **Poor audio** when its worst stream is below 3.6. The problem line says what made it poor: 1% or more of the packets lost, jitter of 30 ms or more, or a gap of 200 ms or more with no audio. If none of these applies, the codec itself is the cause.

Codecs known by number are G.711 μ -law and A-law, G.722, G.729, GSM and G.723.1. Opus, iLBC, Speex, AMR, AMR-WB and others are read from the SDP.

One-way audio

On an answered call with audio in one direction only, pCapitan gives the likely cause:

- a phone behind NAT advertised its **private address** in the SDP, so the other side sends its audio nowhere. Enable SIP ALG or NAT traversal (STUN, ICE), or use a session border controller;
- otherwise, a firewall or NAT lets the audio through in one direction only. Check the RTP port range on the firewall.

No audio at all was seen for an answered call usually means the audio took another path than the capture point. SIP over TLS (port 5061) is encrypted, so those calls cannot be read.

Logins & routing

 **Logins & routing** (Dig in group) explains network logins and routing sessions.

The screenshot shows the pCapitan interface with a sidebar on the left containing navigation options like Captures, Live capture, Overview, and more. The main panel displays 'Logins & routing' data for a file named 'radius-pppoe-bgp.pcap'. It includes a filter bar at the top and three main sections: RADIUS logins, PPPoE internet session, and BGP session eBGP.

RADIUS logins

Each user's attempts through the network access servers (Wi-Fi controllers, VPN gateways, switches).

User	Let in	Refused	Why refused	Method	Through	Answer time
alice		0	1	Password incorrect	wic-1	200 ms
bob	1		0		wic-1	10 ms

► Every exchange (2)

PPPoE internet session

Client: 00:11:22:33:44:55
 Provider (access concentrator): BRAS-7
 Login: CHAP refused Bad password
 Address received: —
 MTU (MRU): —
 Ended: the provider ended the session (PADT): generic error: authentication failed
 Errors: generic error: authentication failed
 PADT (looking for a provider) → PADO (provider offer) → PADR (session request) → PADS (session confirmed) (session 42) → PADT (session terminated) (session 42)

BGP session eBGP

Peers: 192.0.2.1:50000 ↔ 192.0.2.2:179
 Neighbours: 192.0.2.1 (AS 65001, router ID 1.1.1.1, hold time 90 s)
 192.0.2.2 (AS 65002, router ID 2.2.2.2, hold time 90 s)
 State: went down: hold timer expired (no KEEPALIVE received in time: a link, CPU or MTU problem)

🔑 RADIUS logins

One row per user, refused users first:

- **Let in** (Access-Accept) and **Refused** (Access-Reject);
- **Why refused**: the server's reply message. When the server gives none, pCapitan says so: a wrong password, an unknown user or a policy;
- **Method** (the EAP method, for example PEAP or EAP-TLS) and **Through**: the network access server (NAS), such as a Wi-Fi controller, VPN gateway or switch;
- **Answer time**: the average time the RADIUS server took to answer.

Every exchange lists each request and answer with the station, the address given to the user, accounting **Start / Stop**, and why a session **ended** (idle timeout, lost carrier, admin reset and others).

📡 802.1X logins (EAP)

Devices logging in to a switch port or enterprise Wi-Fi: the device's MAC address, the **Identity** it gave, the **Method**, and the **Result**: *accepted, refused, then accepted, refused or unfinished*.

🌐 PPPoE internet sessions

One card per client:

- the **Provider** (access concentrator) and service name;
- the **Login**: *PAP accepted / refused or CHAP accepted / refused*, with the provider's message;
- the **Address received** from the provider;

- the **MTU (MRU)**. When it is below 1500, pCapitan reminds you to clamp the TCP MSS to the MRU minus 40 (1452 → 1412);
- who **Ended** the session: the client or the provider, with a PADT or an LCP Terminate-Request, or *still up at the end of the capture*;
- the steps of discovery (PADI, PADO, PADR, PADS, PADT) and any provider errors.

BGP sessions

One card per session:

- **eBGP** (the two routers are in different AS numbers) or **iBGP** (the same AS), read from the OPEN messages;
- **Neighbours**: each router's address, AS number, router ID and hold time;
- **State**: *established, opening, or went down* with the reason;
- **Routing updates**: how many, how many brought new routes and how many withdrew routes, and the number of keepalives.

When a router closes the session with a **NOTIFICATION**, the card says who closed it, when and why, for example **hold timer expired** (no keepalive in time: a link, CPU or MTU problem), **bad peer AS** (the neighbour's AS number is configured wrong), **maximum number of prefixes reached** or **administrative shutdown**.

RADIUS and BGP are read over IPv4 and IPv6. BGP messages split across several TCP segments are put back together, and 4-byte AS numbers (capability 65) are shown as the real AS number.

Which page for which problem

Symptom	Page
"The application is slow."	 Performance →  Who made it slow?
"It worked yesterday."	 Compare captures → Before / after
"It works inside but not through the firewall."	 Compare captures → Two sides of a firewall or router
Connections reset for no reason.	 Compare captures → Two sides (forged resets)
Large downloads hang, small pages load.	 Compare captures → Two sides (MTU black hole, MSS clamping)
Calls drop, or one side cannot hear.	 VoIP calls
Choppy or robotic voice.	 VoIP calls (MOS, loss, jitter)
Users cannot join the Wi-Fi or VPN.	 Logins & routing → RADIUS / 802.1X
The internet line keeps dropping.	 Logins & routing → PPPoE
Routes disappear, a BGP neighbour flaps.	 Logins & routing → BGP

Keeping detection current: Updates & feeds

The screenshot shows the pCapitan application interface. On the left is a dark sidebar with navigation options: START (Captures, Live capture), UNDERSTAND (Overview, Ask pCapitan, Narrated replay, Network map, World map, Traffic flow, Timeline, Protocols), and DIG IN (Conversations, Packets, Devices, DNS, Web (HTTP), Theme). The main area is titled 'KEEP PCAPITAN AND ITS DETECTION CURRENT' and 'Updates & feeds'. It contains several sections: 'pCapitan updates' with a 'Check now' button and a checkbox for 'Check for updates automatically'; 'Vulnerability database' with a 'Refresh from NVD' button; 'Cisco Talos rules (snort.org)' with an 'Oinkcode' input field and a 'Download Talos rules' button; 'abuse.ch Auth-Key (ThreatFox, URLhaus, MalwareBazaar)' with an input field and a 'Save' button; 'One-click threat feeds'; and 'Location database (GeoIP)'.

- **↑ pCapitan updates:** pCapitan asks gizmogoose.com once a day whether a newer version is out, and shows it here with a **Download** button. The answer is signed by GizmoGoose. Nothing is downloaded or installed by itself. Untick *Check for updates automatically* to stop the daily check, or click **Check now**.
- **⚡ One-click threat feeds:** community blacklists, fingerprint lists and IDS rulesets. Click **Import** to download one (see the list below).
- **🌿 abuse.ch Auth-Key:** three of the feeds (ThreatFox, URLhaus, MalwareBazaar) need a free personal key from abuse.ch (see below).
- **🔔 IDS / IPS rulesets:** fetch a `.rules` URL or upload a file. Snort and Suricata format is supported; it is the same format Cisco Firepower uses. Tick or untick a ruleset to enable or disable it.
- **🦋 Cisco Talos rules (snort.org):** **Talos Community rules** are free and one click away in the threat feeds. For the full Talos ruleset (the rules behind Cisco Firepower and Secure Firewall), sign up at snort.org and paste your **Oinkcode** in the Talos card and click **Download Talos rules:** *Registered* rules are free and 30 days behind, *Subscriber* rules are current. The Oinkcode is stored only on this computer. pCapitan does not use Cisco's licensed firewall update files (VDB, SRU, LSP, GeoDB); they are for Cisco devices only.
- **💊 Vulnerability database:** the offline CVE database used by the **Vulnerabilities** page (NIST NVD, CISA Known Exploited Vulnerabilities, EPSS). pCapitan ships with a recent copy; click

Refresh from NVD to download the latest.

-  **YARA rules:** rules for the files carved out of the traffic (see below).
-  **Threat-intelligence packs:** JSON lists of bad IPs, domains, JA3/JA4 fingerprints, ports and user agents. Upload your own, or point pCapitan at a URL you host. That URL becomes your team's **update channel**.
-  **Antivirus file hashes:** MD5, SHA-1 or SHA-256 lists, or ClamAV `.hsb` / `.hdb` files. Carved files are checked against them.
-  **Location database:** builds the country table from the Regional Internet Registries (no sign-up), or uses a MaxMind GeoLite2 `.mmdb`.
-  **Network owners (ASN):** which company's network each internet address belongs to (see below).

After changing feeds or rules, open a capture and click ↺ (re-analyse) on the home screen so the detectors use the new data.

Pack format (JSON):

```
{"bad_ips": {"203.0.113.9": "our C2 list"},  
"bad_domains": {"evil.example": "phishing kit"},  
"ja3": {"72a589da586844d7f0818ce684948eea": "Metasploit"}}
```

What ships with pCapitan and what you download

pCapitan ships with:

- the **Feodo Tracker**, **URLhaus** and **SSLBL** blacklists and the **JA4+ mapping**, already loaded;
- the **vulnerability database**, with CISA's list of Known Exploited Vulnerabilities;
- the **location** table and the **network owners (ASN)** table;
- the network card **manufacturer** table (IEEE).

Everything else is downloaded by you, from **Updates & feeds**, onto this computer: the other threat feeds, the Talos rules, the YARA rules and your own packs. A download goes straight from its publisher to this computer.

Threat feeds

The  **One-click threat feeds** card lists:

- **Feodo Tracker** botnet C2 IPs (abuse.ch; mostly empty since Operation Endgame)
- **ThreatFox** C2 IP:port, domains, URLs and hashes, with malware names (abuse.ch, needs the Auth-Key)
- **MalwareBazaar** recent malware SHA-256 hashes (abuse.ch, needs the Auth-Key)
- **C2IntelFeeds:** C2 servers seen in the last 30 days
- **CINS Army:** active scanners and brute-force attackers
- **Tor exit nodes** (Tor Project, with a mirror)
- **OpenPhish:** active phishing sites

- **Public DNS-over-HTTPS servers** (encrypted DNS that bypasses your DNS server)
- **JA4+ mapping** (FoxIO): which application each JA4 fingerprint belongs to
- **PT Research AttackDetection** Suricata rules (new CVEs, Active Directory attacks)
- **OISF TrafficID**: application identification rules
- **URLhaus** malware URLs (abuse.ch, needs the Auth-Key)
- **SSLBL** malicious JA3 fingerprints (abuse.ch)
- **Spamhaus DROP**: hijacked and criminal netblocks (EDROP is merged into it)
- **Spamhaus DROP for IPv6**
- **FireHOL level-1** blocklist
- **Emerging Threats open** Suricata ruleset
- **Talos Snort Community** rules (Cisco Talos)

Click **Import** next to a feed. It shows ✓ when done, or ✗ with the reason. Blacklists become threat-intelligence packs, rulesets appear under 🚨 **IDS / IPS rulesets**, and file hashes (MalwareBazaar, and the hashes in ThreatFox) appear under 🦟 **Antivirus file hashes**.

URL feeds name single malicious pages on big shared platforms (GitHub, Google Drive, Dropbox and others). pCapitan never blacklists those platforms' whole domains, so ordinary visits to them are not flagged.

The abuse.ch Auth-Key. abuse.ch now gives its ThreatFox, URLhaus and MalwareBazaar datasets only with a free personal key. Create one at auth.abuse.ch, paste it in the 🦟 **abuse.ch Auth-Key** card and click **Save**. The card then shows *Key saved* with its last four characters. The key stays on this computer and is sent only to abuse.ch. To remove it, clear the box and click **Save**. Without a key, importing one of these three feeds stops with a message asking for the key. The URLhaus list that ships with pCapitan works without a key; you need the key only to refresh it.

CISA Known Exploited Vulnerabilities

CISA keeps a list of the vulnerabilities that criminals are known to exploit. pCapitan has the whole list, in the vulnerability database.

Wherever pCapitan shows a CVE number that is on CISA's list, it adds the badge 🔥 **actively exploited**, and 🔥 **actively exploited · ransomware** when CISA says ransomware campaigns use it. You see it on IDS alerts (the **IDS signatures** page), on 🎯 **Exploits attempted** on the **Vulnerabilities** page, and in the reasons given for a device's verdict (**What happened?** and the device drawer). Hover over the badge for an explanation.

The list is updated with 🦟 **Vulnerability database** → **Refresh from NVD**.

🧬 YARA rules

YARA rules recognise malware and attack tools by patterns in their files. pCapitan scans the files it carves out of the traffic (**Forensics** → **Files**) with the rules in the 🧬 **YARA rules for carved files** card.

- **↓ YARA-Forge core** or **↓ YARA-Forge extended**: YARA-Forge is a free, curated collection of public rules. One click downloads it onto this computer. Each rule keeps its author's licence, so pCapitan does not ship these rules. *Extended* contains *core*, so pCapitan keeps one of the two: downloading one replaces the other.
- **↑ Add my .yar file**: add your own rules (`.yar` or `.yara`). pCapitan checks that the file compiles before it keeps it.

The card lists each rule file, its number of rules and when it was updated; **remove** deletes one. A rule file that does not compile is skipped and named in red under *Skipped (do not compile)*. The other files still work.

FORENSICS WORKBENCH

Forensics

Evidence extracted from the traffic: logins, files, file-share access, indicators of compromise and raw content search.

[Credentials](#)
[Extracted files](#)
[File-share activity](#)
[IOC matching](#)
[Payload hunt](#)
[DHCP](#)
[ARP](#)
[Content indicators](#)

Credentials seen in the traffic

These were readable by anyone able to capture the network — exactly like this file. Change them.

[Reveal secrets](#)
[CSV](#)

Time	Protocol	Client	Server	Username	Secret	Context
13.850	HTTP Basic	MacBook-Pro	intranet.corp.local	admin	S*****!	http://intranet.corp.local/admin
13.927	HTTP form	DESKTOP-ACCT01	example.com	jdoe	P*****d	POST http://example.com/login.php
14.637	FTP	MacBook-Pro	intranet.corp.local	backup	b*****3	clear-text login
14.679	SNMP v2c	intranet.corp.local	192.168.1.250	community	p****c	SNMP community strings act as passwords and are sent in clear text
14.738	NTLM (SMB)	DESKTOP-ACCT01	intranet.corp.local	CORP\jdoe	NTLM challenge-response (hash, 24 bytes)	workstation DESKTOP-ACCT01
51.539	RDP	203.0.113.9	intranet.corp.local	administrator		RDP connection cookie (username only)

Files are scanned when you click **Extract files on Forensics** → **Files**. After adding or changing rules, click **Scan again with YARA rules** there. The **YARA** column shows the rules each file matched (hover for the description and author), or *no match*. Files over 64 MB are scanned on their first 64 MB.

Every match becomes a **Malware indicator** finding: the file, its type, the protocol it came over, the rule, the rule set and the rule's author. Rules that give a score set the severity (80 or more is *critical*, 60 or more *high*, lower *medium*); rules without a score give *high*.

Network owners (ASN)

pCapitan ships with a table of the networks announced on the internet (from iptasn.com, public domain). For each internet address it shows the network that owns it, for example `AS13335 CLOUDFLARENET` : a cloud provider, a hosting company or a home ISP. You see it on the **Devices** page

and in the device drawer ( **Network**). Local addresses have no owner.  **Ask pCapitan** also answers questions such as *who owns 203.0.113.9?* or *which cloud providers did we talk to?*.

The card shows how many networks and address ranges the table holds. Click **Update now** to download the current table. pCapitan uses whichever is newer: your download or the copy that ships with the version you have.

Live capture

 **Live capture** records from a network adapter and analyses it **while it is happening**. The overview, map, packet ticker and findings refresh every few seconds.

 **pCapitan**
by GizmoGoose

START

Captures

Live capture

UNDERSTAND

Overview

Ask pCapitan

Narrated replay

Network map

World map

Traffic flow

Timeline

Protocols

DIG IN

Conversations

Packets

Devices

DNS

Web (HTTP)

Theme

Wireshark 4.6.9

Filter... e.g. dns.qry.name contains "google" · ip.addr == 10.0.0.0/8 && !tls · 192.168.1.5 · 443

Tip: press / to filter · type an IP, port, domain or expression · has ~2,000 ready-made filters explained in plain language

LISTEN TO THE NETWORK

Live capture

Capture packets from a network interface and watch pCapitan analyse them in real time. The capture is saved, so you can keep investigating afterwards.

 **Wi-Fi**
192.168.1.10

★ In use Change ▶

Capturing with dumptcap

Capture filter (BPF, optional) — reduces what is recorded

e.g. not port 22 · host 192.168.1.10 · tcp port 443

Everything

No SSH (avoid self-capture)

Only web

Only DNS

No broadcast

One host...

Stop after (seconds)

Stop after (MB)

Stop after (packets)

Name

Live capture

Start capturing

 **How it works**

- dumptcap** (from Wireshark) records packets from the chosen interface into a capture file.
- pCapitan reads the file *while it grows*, decoding every packet immediately.
- Every ~3 seconds, conversations, devices and all security detectors are refreshed — the overview, map, findings and packet list update live.
- Press **Stop** to finish; the full capture stays available for analysis, reporting and export.

Capture filters use BPF syntax (like tcpdump) and decide what is *recorded*. The **f** display filter at the top decides what is *shown*, at any time.

 Only capture networks you own or are authorised to monitor.

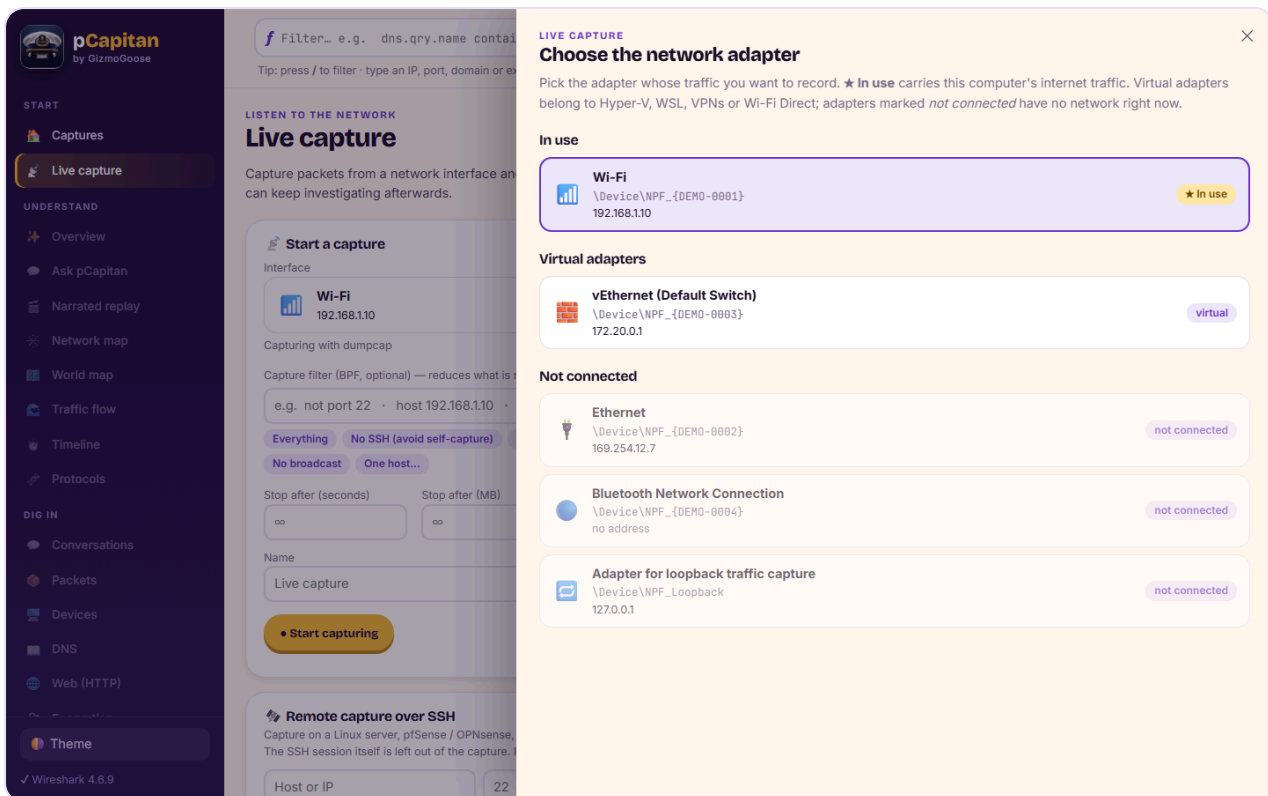
 **Remote capture over SSH**

Capture on a Linux server, pfSense / OPNsense, OpenWrt or Ubiquiti EdgeOS router without installing anything there; pCapitan runs tcpdump over SSH and analyses the traffic live, here. The SSH session itself is left out of the capture. Passwords and keys are used for this connection only and never stored.

Host or IP

22

User (root or a sudo user)



1. Click the adapter card (**Change** ►) to open the **adapter drawer**. The adapter this computer uses for the internet is marked ★ **In use** and chosen for you; virtual adapters (Hyper-V, WSL, VPN) and adapters that are not connected are listed separately, each with its addresses. Click a card to choose it.
2. Optionally, add a **capture filter** in BPF syntax, for example `not port 22`, `host 192.168.1.10` or `tcp port 443`. Presets are one click away.
3. Optionally, set limits: stop after N seconds, N MB or N packets.
4. Click ● **Start capturing**. Click ■ **Stop capture** when done. The capture stays available like any other file.

Live capture needs **Wireshark 4.2 or later** (the latest is recommended) + **Npcap** (Windows) or **dumpcap with capture permission** (Linux: `sudo usermod -aG wireshark $USER`; macOS: install Wireshark's `ChmodBPF`). Only capture networks you own or are authorised to monitor.

Wireshark's capture program (dumpcap) sends the packets to pCapitan, which writes the capture file itself, so live capture works the same from the installer and from the Microsoft Store.

Live alerts

During a live capture the detectors run again every few seconds. When a new serious finding appears (*critical* or *high*, or an attack that worked), an alert such as 🚨 **New critical finding** pops up at once on whatever page is open, with the finding's outcome. Click **Open** to read the finding, or **What happened?** to see the whole story. An alert closes by itself after 30 seconds, or click X.

- At most three alerts pop up at a time; a message tells you how many more are waiting on **What happened?**.

- The **What happened?** entry in the sidebar counts the new findings.
- A finding whose count grows (*3 attempts*, then *40 attempts*) is not announced again.
- Findings that were already there when you opened the capture are not announced.

Remote capture over SSH

Capture on a Linux server, a pfSense or OPNsense firewall, an OpenWrt router or a Ubiquiti EdgeOS router, and analyse the traffic live on this computer. pCapitan runs `tcpdump` on the remote machine over SSH. Nothing is installed there, and you do not need Wireshark on this computer for it.

1. In the  **Remote capture over SSH** card, type the host, the port (22 if empty) and the user: `root`, or a user that may use `sudo`.
2. Type the password, or open **Use a private key instead** and paste the key (OpenSSH or PEM; RSA, ECDSA or Ed25519) and its passphrase, if it has one.
3. Click **Connect**.
4. **The first time** you connect to a machine, pCapitan shows its host key fingerprint. Check it on the server (the card shows the `ssh-keygen -lf` command to run there) and click **It matches: trust and connect**.
5. Choose the interface, optionally a capture filter and limits (*stop after* seconds, MB or packets), and click **Start remote capture**.

The capture then works like a local live capture, with **Stop capture**, live findings and live alerts. Its title shows  **host over SSH**.

What the remote machine needs:

- **tcpdump** installed (for example `apt install tcpdump`). If it is missing, pCapitan says so after connecting.
- To log in as **root**, or as a user that may run `sudo`. As another user, pCapitan runs `tcpdump` with `sudo`, using the login password. If you log in with a key, the user needs passwordless `sudo`.
- MikroTik RouterOS has no `tcpdump`. pCapitan says so and suggests its packet sniffer (*Tools > Packet Sniffer*): save a `.pcap` there and open it in pCapitan.

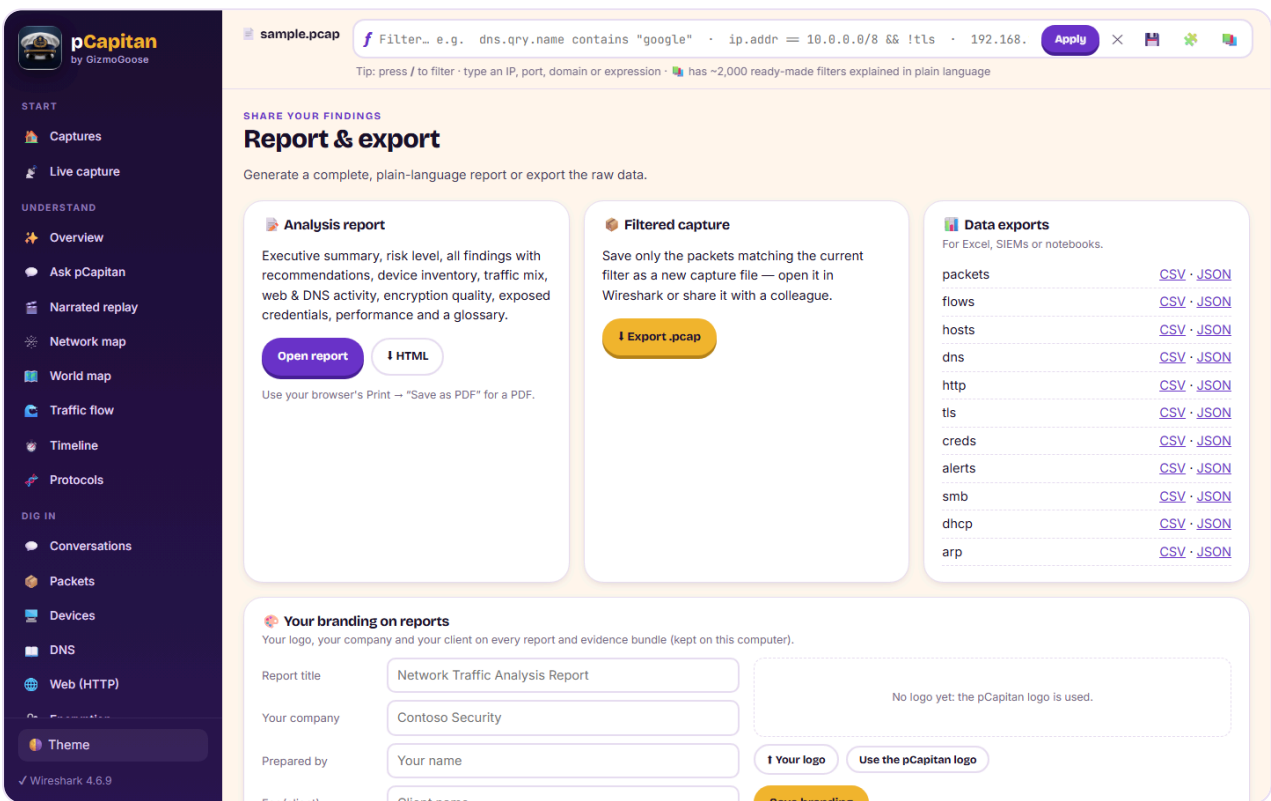
Security:

- **Host key:** pCapitan remembers each machine's host key once you have trusted it, in `ssh_known_hosts.json` in the data folder. If the key changes later, pCapitan refuses to connect, because this can be a man-in-the-middle attack. If the server was reinstalled, click **The server was reinstalled: forget the old key**, then check and confirm the new fingerprint.
- **Passwords and keys** are used for the connection only and never stored.
- **The SSH session itself** is left out of the capture.
- A connection that is not used to start a capture expires after 10 minutes; connect again.

Reports and exports

 **Report & export** creates, for the current filter:

- **Analysis report:** a printable incident report. It starts with the status, what happened and what to do now, then the attacks that worked, affected devices, known vulnerabilities, data that left, a timeline, an action plan and the indicators to block (see *Incident response*). The appendices hold every finding with recommendations, device inventory, traffic mix, web and DNS activity, encryption quality, exposed credentials, performance and a glossary. **Open report**, then print and choose *Save as PDF*.
- **Filtered capture:** only the matching packets, as a `.pcap` for Wireshark or a colleague.
- **Data exports:** CSV or JSON of packets, conversations, devices, DNS, HTTP, TLS, credentials, findings, SMB, DHCP and ARP, for Excel, a SIEM or a notebook.
-  **Vendor-safe copy:** the matching packets with your private details replaced, for a vendor's support team (see below).



The screenshot shows the pCapitan interface. The sidebar on the left contains navigation links: START (Captures, Live capture), UNDERSTAND (Overview, Ask pCapitan, Narrated replay, Network map, World map, Traffic flow, Timeline, Protocols), and DIG IN (Conversations, Packets, Devices, DNS, Web (HTTP), Theme). The main content area is titled 'Report & export' and includes a filter bar at the top. Below the filter bar, there are three main sections: 'Analysis report' (with 'Open report' and 'HTML' buttons), 'Filtered capture' (with 'Export .pcap' button), and 'Data exports' (listing various data types with CSV and JSON links). At the bottom, the 'Your branding on reports' section allows users to customize their reports with fields for Report title, Your company, Prepared by, and Client name, and buttons for 'Your logo' and 'Use the pCapitan logo'.

Your branding on reports

Put your own name on what you hand over. Fill in the  **Your branding on reports** card and click **Save branding**:

- **Report title**, for example *Network Traffic Analysis Report*;
- **Your company** and **Prepared by**;

- **For (client);**
- **Accent colour**, for the report's lines and headings;
- **Footer:** contact details, a disclaimer or a case number;
- **Mark as CONFIDENTIAL:** a red CONFIDENTIAL banner at the top of the report;
- **Don't name pCapitan in the header.**

↑ **Your logo** replaces the pCapitan logo. Use a PNG, a JPEG or a plain SVG, under 1 MB. An SVG that contains scripts or links to other files is refused: save it as a plain SVG or a PNG. **Use the pCapitan logo** removes yours.

The preview at the bottom of the page shows the result at once. The branding is kept on this computer and applies to every report, evidence bundle and batch report made afterwards.

•• Vendor-safe copy

Vendors' support teams (Cisco TAC, Fortinet and others) often ask for a capture. The •• **Vendor-safe copy** card makes a copy of the filtered packets without your private details:

- **Internal addresses** are replaced. Private IPv4 addresses become `10.128.0.0/9` addresses; each /24 network stays together and keeps its host numbers, so `10.10.5.22` and `10.10.5.40` stay neighbours. Automatic addresses (169.254.x.x) get a stable stand-in in the same range. Every IPv6 address becomes an `fd70:6361::` address, because organisations number their own devices from public IPv6 ranges too. Public IPv4 addresses are replaced only when you tick **Also replace public (internet) IPv4 addresses**. Addresses are also replaced inside ARP, ICMP errors, DNS answers, DHCP and text.
- **MAC addresses** are replaced; the manufacturer part is kept.
- **Passwords, cookies and tokens** are masked with `*`: HTTP Authorization headers and cookies, FTP, POP and IMAP log-ins, and `password=` or `token=` style parameters.
- **Internal names** are masked with `x`. pCapitan suggests names from the capture (domain and host names of local devices, user names, Kerberos realms); add your company, domain and user names.
- **Kept:** timestamps, ports, packet sizes and TCP sequence numbers. Checksums are recalculated, so the vendor can still follow every conversation in Wireshark.

Options:

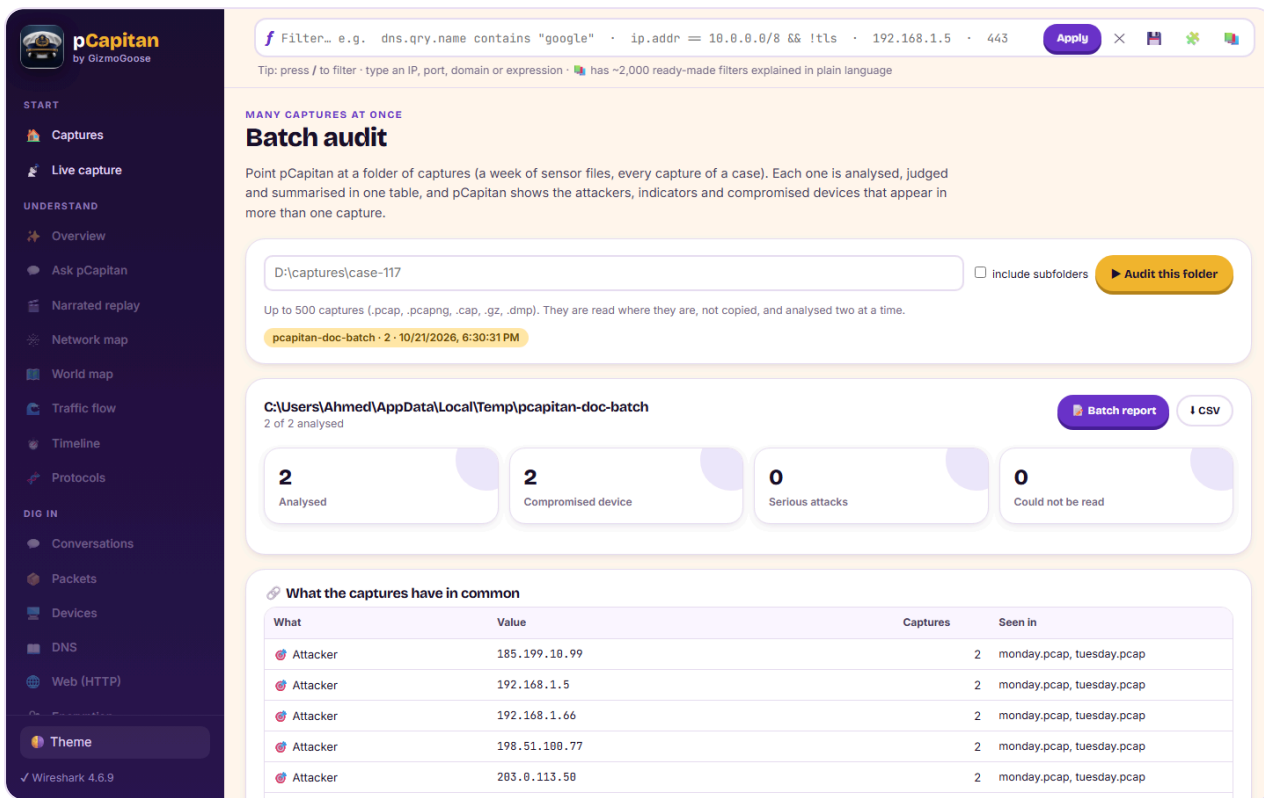
- **Also replace public (internet) addresses:** they become `198.18.0.0/15` addresses. Without this, public IPv4 addresses stay.
- **Headers only:** every byte of content is set to zero. This is the most private; the vendor sees timing, sizes and flags.
- **Include the address map:** adds a list of each real address and its replacement, for you to keep. Do not send it.

Click ↓ **Download vendor-safe copy (.zip)**. The zip holds the capture and **WHAT-WAS-CHANGED.txt**, which counts each kind of change. Packets pCapitan could not read are zeroed after the link header.

Before you send it, open the copy and search for your company, domain and user names.

Batch audit

 **Batch audit** (Tools group in the sidebar) analyses a whole folder of captures, for example a week of sensor files or every capture of a case, and puts the results in one table.



The screenshot shows the pCapitan Batch audit interface. The sidebar on the left contains navigation options: START (Captures, Live capture), UNDERSTAND (Overview, Ask pCapitan, Narrated replay, Network map, World map, Traffic flow, Timeline, Protocols), and DIG IN (Conversations, Packets, Devices, DNS, Web (HTTP), Theme). The main panel has a filter bar at the top with a search filter and an 'Apply' button. Below the filter bar, the 'Batch audit' section shows a folder path 'D:\captures\case-117' and a button 'Audit this folder'. A progress bar indicates 'Up to 500 captures (.pcap, .pcapng, .cap, .gz, .dmp). They are read where they are, not copied, and analysed two at a time.' Below the progress bar, a table shows the results of the batch audit: 2 Analysed, 2 Compromised device, 0 Serious attacks, and 0 Could not be read. At the bottom, a section titled 'What the captures have in common' displays a table with columns 'What', 'Value', 'Captures', and 'Seen in'.

What	Value	Captures	Seen in
Attacker	185.199.18.99	2	monday.pcap, tuesday.pcap
Attacker	192.168.1.5	2	monday.pcap, tuesday.pcap
Attacker	192.168.1.66	2	monday.pcap, tuesday.pcap
Attacker	198.51.100.77	2	monday.pcap, tuesday.pcap
Attacker	203.0.113.50	2	monday.pcap, tuesday.pcap

1. Type the folder that holds the captures. Tick **include subfolders** to include them too.
2. Click ► **Audit this folder**.

- A batch takes up to **500** captures (`.pcap` , `.pcapng` , `.cap` , `.pcap.gz` , `.pcapng.gz`). A folder with more is refused.
- The captures are **read where they are**, not copied, and analysed **two at a time**.
- A progress bar shows how many are done.

 **Every capture** lists, for each capture, its verdict (*compromised*, *attacked*, *minor notes* or *clean*), packets, devices, findings by severity, compromised devices and its top findings. Click a capture's name to open its **What happened?** page. A file that cannot be read shows **not read**; hover over it for the reason. If pCapitan is closed while a batch is running, the batch is shown as *interrupted*: what was analysed is kept, and the rest is marked *not analysed*. The counters at the top show how many were analysed, how many have a compromised device, how many have serious attacks and how many could not be read.

 **What the captures have in common** appears when the batch is done: the attackers, compromised devices and indicators (addresses, domains, URLs, hashes) that appear in more than one

capture, and in which ones.

When the batch is done:

- **↓ CSV:** the table, one line per capture, for Excel.
-  **Batch report:** a printable summary of the whole batch, with your branding.

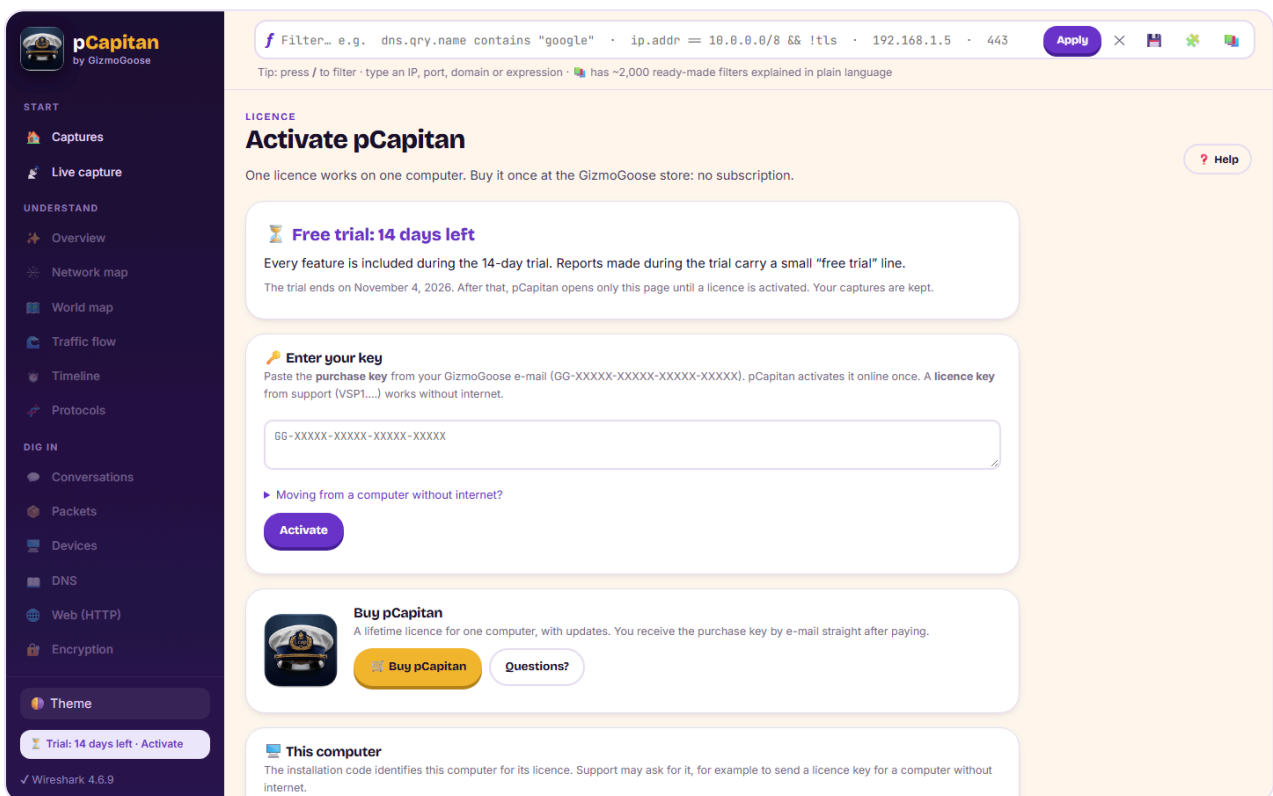
Earlier batches are kept and listed above the results; click one to open it again.

Licence, trial and activation

pCapitan is sold by GizmoGoose at gizmogoose.com/tools/pcapitan. You pay once and keep it: there is no subscription. One licence works on one computer at a time, and you can move it to a new computer yourself.

The 14-day free trial

- The trial starts the first time pCapitan opens, and it includes **every feature**.
- 🕒 **Trial: N days left** in the bottom-left corner shows how long is left. Click it to open the **License** page.
- Reports made during the trial end with a small line: *Made with the pCapitan free trial*. Nothing else is marked or limited.
- When the trial ends, pCapitan opens only the License page until a licence is activated. Your captures, filters and feeds are kept, and everything is back as soon as you activate.



If the computer's clock is set back by more than a day, pCapitan cannot check the trial and asks you to set the correct date and time.

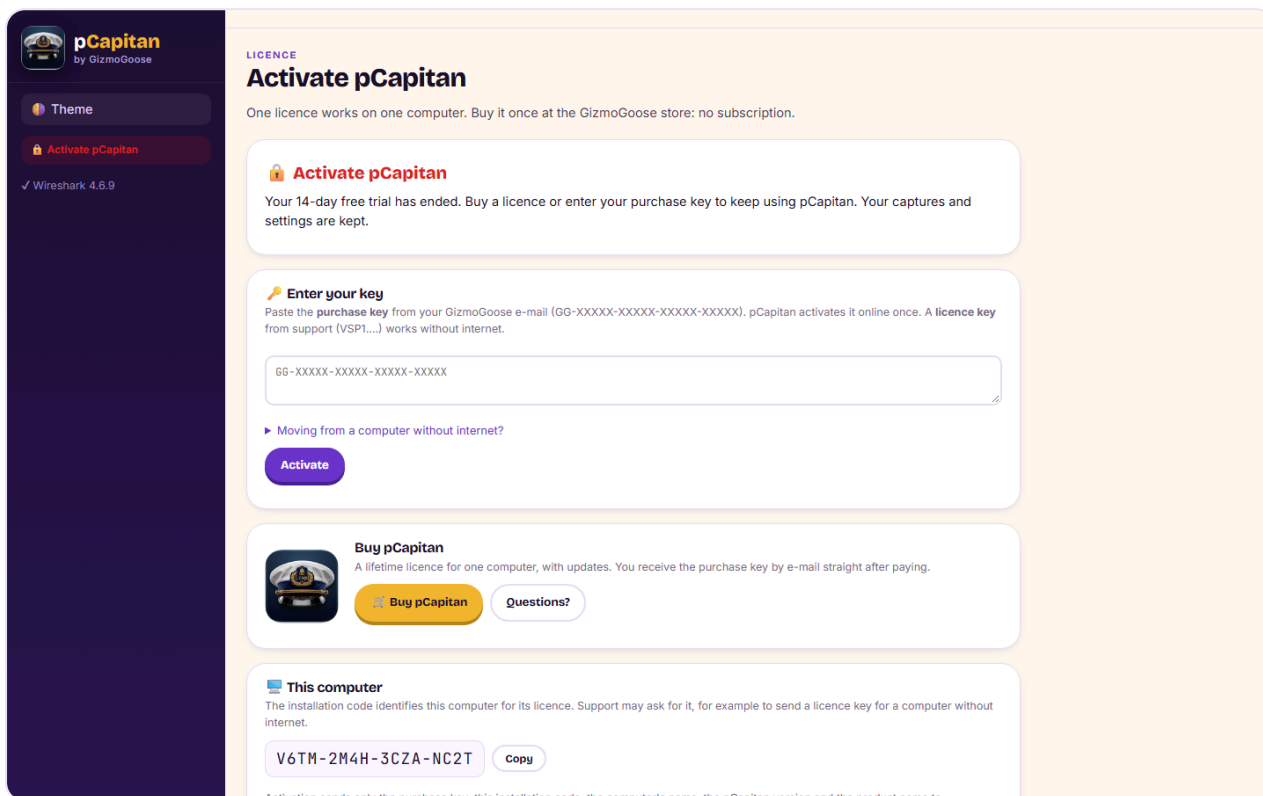
Buying pCapitan

1. On the License page, click  **Buy pCapitan**, or go to gizmogoose.com/tools/pcapitan.
2. Pay at the GizmoGoose checkout (Qi Card, ZainCash and more).

3. Right after paying, you receive an e-mail with your **purchase key**, which looks like GG-XXXXX-XXXXX-XXXXX-XXXXX . You can also see it at any time on your customer desk at gizmogoose.com/account (sign in with the same e-mail address).

Activating

1. Open  **License** in the sidebar. When the trial has ended, this page opens by itself.
2. Paste the purchase key into **Enter your key** and click **Activate**.
3. pCapitan contacts gizmogoose.com once, and the store sends back a licence for this computer. The page then shows  **pCapitan is licensed** and who it is licensed to.



An internet connection is needed for a moment. A computer that is never online can be activated with a **licence key** (it starts with VSP1.) from GizmoGoose support:

- send the **installation code** shown under *This computer* to **support@gizmogoose.com**;
- paste the licence key you receive into the same box and click **Activate**. A licence key works only on the computer whose installation code it was made for, and it never needs the 45-day check described below, so it suits analysis computers that are kept off the internet.

If activation fails

Message	What to do
<i>The store does not know this purchase key</i>	Check the key against your e-mail. Copy and paste it rather than typing it.
<i>This purchase key is for another GizmoGoose app</i>	The key is for a different product, such as VectorSight Pro. Use the pCapitan purchase key.
<i>This purchase is active on another computer</i>	Move it from the old computer first (see below), or enter the old computer's release code.
<i>The GizmoGoose store could not be reached</i>	Check the internet connection, a proxy or the firewall, then try again.
<i>This purchase has been moved too many times this year</i>	Contact support. A purchase can move to a new computer a few times a year.

Moving to another computer

With internet on the old computer:

1. On the old computer, open **License** and click **Move to another computer**. The store frees the purchase, and pCapitan on the old computer goes back to the License page.
2. On the new computer, activate with the same purchase key.

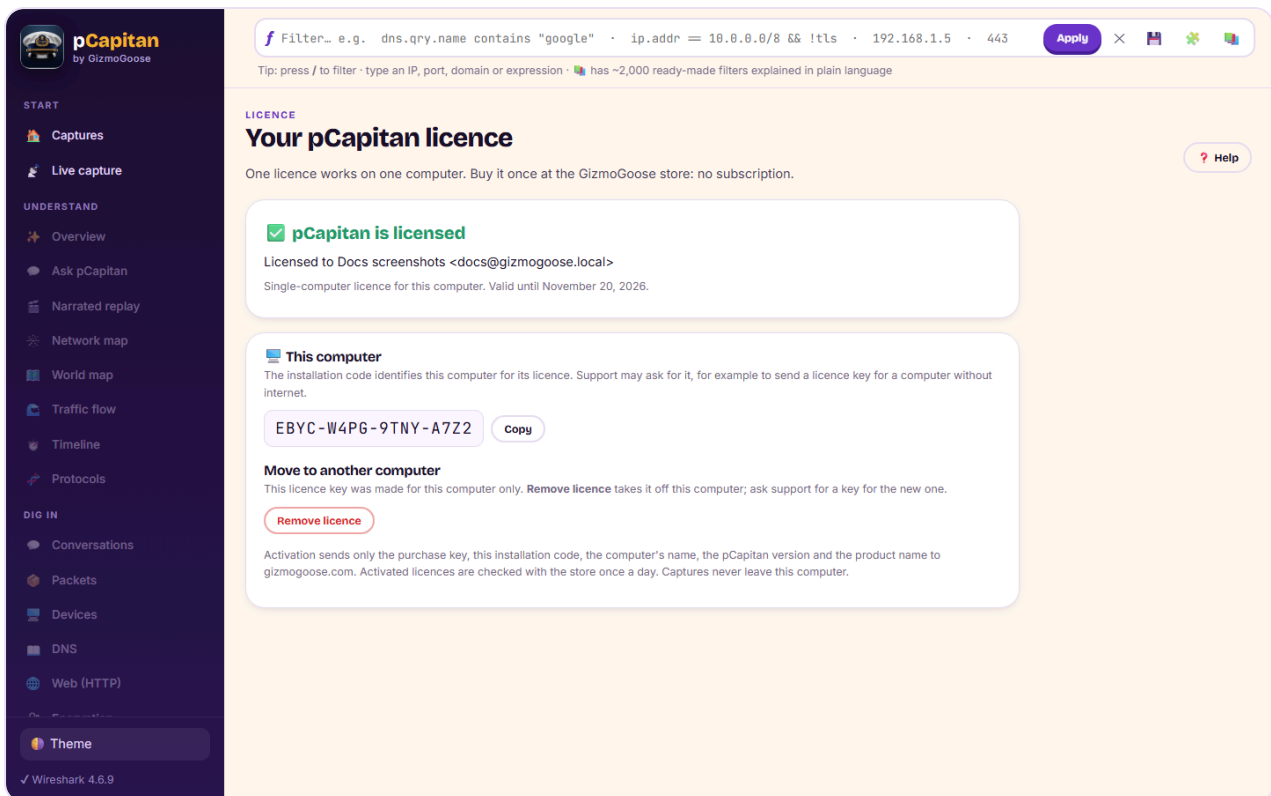
When the old computer is offline:

1. On the old computer, click **Remove licence**. pCapitan shows your purchase key and a **release code** (xxxx-xxxx-xxxx). Write both down.
2. On the new computer, paste the purchase key, open **Moving from a computer without internet?**, type the release code and click **Activate**.

You can also enter the release code at gizmogoose.com/account/release. If the old computer is lost or broken, free it on your customer desk at gizmogoose.com/account, or ask support.

The licence check

- A licence activated on this computer is checked with gizmogoose.com **once a day** while pCapitan runs. The check downloads the store's signed list of licences that were moved away or refunded. Nothing about your computer or your captures is sent.
- A licence that was moved to another computer stops working on the old one at its next check.
- pCapitan must reach the store at least once every **45 days**. After 30 days without a check, the corner shows a reminder. If the 45 days pass, pCapitan asks for a check the next time it starts. Connect to the internet and click ↻ **Check now** on the License page.



Updates

pCapitan asks gizmogoose.com once a day whether a newer version is out. The answer is signed by GizmoGoose, so it cannot be faked. When there is one, **Updates & feeds** shows it with a **Download** button, and a dot appears next to it in the sidebar. pCapitan never downloads or installs anything by itself. You can switch the daily check off on that page.

Updates are free for a year from your purchase. Install the new version over the old one: your licence, captures and settings stay.

Troubleshooting and FAQ

"Wireshark not found" in the corner. Install Wireshark from [wireshark.org](https://www.wireshark.org). On Windows, tick **Npcap** during setup. Then restart pCapitan. Analysis works without it; live capture, the dissection tree, file carving, Kerberos and decryption need it.

Live capture shows no interfaces, or errors out.

- **Windows:** reinstall Npcap and choose "Support raw 802.11 traffic" only if you need Wi-Fi monitor mode.
- **Linux:** add yourself to the `wireshark` group and log in again.
- **macOS:** install `ChmodBPF` from the Wireshark DMG.

A big capture takes minutes. That is expected. pCapitan decodes every packet across every layer, tracks it into a conversation and indexes it, at roughly 40,000 packets per second per file. A few GB holds tens of millions of packets. Use **Analyse it in place** to skip the upload copy. Once a capture is analysed, every view and filter is instant.

The window is blank (Windows). pCapitan uses Microsoft Edge WebView2, which is built into Windows 10/11. If it was removed, install the *WebView2 Runtime* from Microsoft, or start pCapitan with `pCapitan.exe --browser` to use your browser instead.

A finding seems wrong. Findings are leads. Open the evidence and click  **Investigate**. Many patterns, such as uploads or regular check-ins, have legitimate causes, and the explanation says which.

Licence questions. See the chapter *Licence, trial and activation*: buying, activating, moving to another computer and the daily licence check.

How do I start fresh? Delete captures from the home screen. To reset everything, close pCapitan and delete the data folder:

- Windows: `%LOCALAPPDATA%\pCapitan`
- macOS: `~/Library/Application Support/pCapitan`
- Linux: `~/.local/share/pCapitan`

The bundled feeds are restored on the next start. Your licence is in the `license` folder there: keep it, or move the licence first (License → Move to another computer).

Command-line options.

- `pCapitan --browser` opens in your browser.
- `pCapitan --server --port 8765` runs the engine only.
- `pCapitan file.pcap` opens a file.
- `pCapitan --version` prints the version.

Keyboard shortcuts

Key	Action
/	jump to the filter bar
Enter	apply the filter
Tab / ↓ ↑	pick a suggested field
Esc	close a panel or dialog

Privacy and responsible use

- Captures, analyses and TLS keys stay on your computer. pCapitan goes online only:
 - when *you* refresh feeds, rules or the location database;
 - once a day, to ask gizmogoose.com whether a newer pCapitan is out (you can switch this off under **Updates & feeds**);
 - to activate or move a licence, and once a day for the licence check of an activated licence. These send only the purchase key, the installation code, the computer's name and the pCapitan version.
- OSINT buttons open your browser and send only the IP address or file hash.
- Network captures can contain other people's private data. **Only capture or analyse traffic you own or are authorised to examine**, and follow the privacy laws that apply to you.

Support

GizmoGoose: **support@gizmogoose.com** · <https://gizmogoose.com/support/> Include the log file, found in the `logs` folder inside the data folder.