

Where Digital Evidence Lives — and How to Keep It Admissible

Fri, Sep 4, 6:00 PM EDT · <https://scottslab.io/posts/digital-forensics-sources-chain-of-custody>



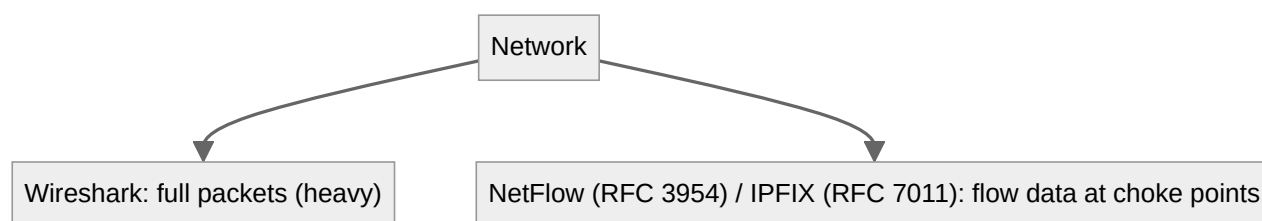
TL;DR

Digital evidence hides in more places than the disk: full packet captures (Wireshark) or flow data (NetFlow, or its standardized successor IPFIX, RFC 7011) at network choke points, software forensics for IP disputes and malware attribution, email headers checked against SPF/DKIM/DMARC, and mobile devices. NIST's mobile forensics guide (SP 800-101) ranks acquisition methods on a five-level pyramid, from reading the screen to reading individual memory gates under a microscope. None of it matters without chain of custody: a specific, RFC-documented record of who found it, who handled it, who held it, and how it moved. A defense attorney only needs to find one gap.

The disk image is where forensics starts, not where it ends. Evidence lives across a bunch of sources, each with its own trade-off, and each with its own way of getting thrown out if you're careless.

Network: depth vs. volume

On the wire, you're choosing between depth and volume. Wireshark captures every packet: powerful, and brutally storage-intensive. Flow-based tools capture high-level metadata instead: source and destination IPs, ports, timestamps, and volume, gathered at network choke points. Cisco's NetFlow (version 9 is documented in RFC 3954) became the basis for **IPFIX**, RFC 7011, the IETF's standardized flow-export protocol. So "NetFlow-style" data today is often literally IPFIX under a different vendor's name. Full capture tells you *what was said*. Flow data tells you *who talked to whom, when, and how much*. For most investigations that's what you actually have, because nobody stores every packet forever.



Network: depth vs. volume

Beyond the wire

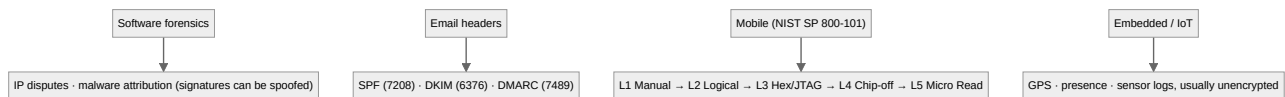
Software forensics analyzes the code itself. It's used for intellectual-property disputes and malware attribution, but tread carefully: public threat-actor signatures can be spoofed or misattributed, and treating one as ground truth is how you accuse the wrong party.

Email header analysis walks the routing path and checks three protocols against each other to tell a forged sender from a real one: **SPF** (RFC 7208), which lists which servers are authorized to send for a domain; **DKIM** (RFC 6376), which cryptographically signs the message so a relay can't tamper with it undetected; and **DMARC** (RFC 7489), which tells a receiving server what to do when SPF or DKIM fails. Tools like MXToolbox parse and check all three for you, but knowing what each one actually verifies (authorized sender, message integrity, and enforcement policy, respectively) is what lets you read a "DMARC pass" correctly instead of treating it as a blanket "this email is legitimate."

Mobile devices are mostly encrypted now, so without the passcode the forensic success rate is genuinely low. That's a real change from a decade ago. NIST SP 800-101 Rev. 1, "Guidelines on Mobile Device Forensics," frames the acquisition options as a five-level pyramid, and the levels aren't interchangeable: **Level 1, Manual**: recording what's on the screen by hand, no recovery of deleted data. **Level 2, Logical**: pulling files and directories over a standard connection, the most common approach and the least technical. **Level 3, Hex Dump/JTAG**: a physical acquisition of memory in place, requiring advanced skill. **Level 4, Chip-Off**: physically desoldering the memory chip. **Level 5, Micro Read**: viewing the physical state of individual gates under a microscope. Cost,

invasiveness, and required skill all climb with the level, and going up a level is often irreversible on the device itself. Chip-off destroys the phone whether or not it recovers anything useful.

Embedded and IoT devices (cars, smart-home gear, sensors) increasingly hold the decisive evidence: GPS history, presence data, even temperature logs that place a person or event in time, often with none of the encryption headaches a modern phone has.

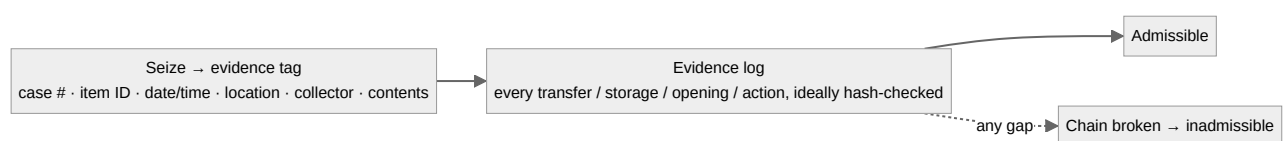


Beyond the wire

Chain of custody, as an actual document

None of these sources are worth anything in a proceeding without chain of custody: the unbroken, documented history of the evidence from seizure onward. RFC 3227 spells out exactly what has to be on record, and it's a short, specific list: **where, when, and by whom the evidence was discovered and collected; where, when, and by whom it was handled or examined; who had custody, for what period, and how it was stored; and, every time custody changed, when and how the transfer happened, including shipping or tracking numbers where relevant.**

In practice that turns into two artifacts. An **evidence tag or bag**, labeled with a case number, a unique item ID, the date and time, the location, the collector's name, and a description of the contents. And an **evidence log**, a running record of every single transfer, every time it was stored, every time it was opened, and every action taken on it. Many practitioners also record the item's hash at each handoff, so a break in integrity is detectable at the exact link in the chain where it happened, not just "somewhere."



Chain of custody, as an actual document

The point of all of it is being able to account for the item at every moment. A defense attorney only has to show one gap, one window where the evidence was unaccounted for, to argue it could have been altered. The cleverest packet capture or the cleanest chip-off image in the world is worthless if you can't prove where it's been since the moment you took it.