GitHub GitLab And Source Code As Evidence

Source code from platforms like GitHub and GitLab offers a detailed, date-stamped history of development. This guide assists UK litigators, in-house counsel, and investigators in understanding and utilising this digital evidence, covering its anatomy, authentication, and common pitfalls.

e-discovery.uk, Computer Forensics Lab. Published 2026-08-30.
https://e-discovery.uk/library/github-gitlab-and-source-code-as-evidence

SOURCECODEEVIDENCE · A GUIDE FOR UK LAWYERS GitHub, GitLab and Source Code as Evidence Commits, Clones and the Development History That Dates Every Line COMPUTER FORENSICS LAB

§ ABOUT THE AUTHOR PREPARED BY COMPUTER FORENSICS LAB E-DISCOVERY TEAM

§ CONTENTS In this guide 01 Executive summary 02 The problem in plain English: the manuscript that dates itself 03 Git anatomy: commits, branches, hashes and what they prove 04 Platform evidence: pull requests, reviews, issues and logs 05 Authenticating history: forged dates and rewritten pasts 06 Exfiltration: clones, forks and the departing developer 07 Copying disputes: similarity analysis and provenance 08 Source architecture: where else the evidence lives 09 Worked examples 10 Common mistakes and technical limitations 11 Questions to ask · Suggested wording 12 Checklist and red flags · When to involve a digital forensic expert 13 Frequently asked questions 14 Glossary · References · Disclaimer · How a specialist laboratory can assist

§ 01 · ORIENTATION Executive summary THE HEADLINE POINT: GITDATESEVERYLINEANDNAMESEVERYAUTHOR, BUTBOTH FIELDS ARE SET TABLE: PLATFORM - SIDERECORDSARETHEANCHORTHATMAKES REPOSITORYHISTORYTRUSTWORTHY

§ 02 · FIRST PRINCIPLES The problem in plain English: the manuscript that dates itself

§ 03 · THEREPOSITORY Git anatomy: commits, branches, hashes and what they prove

§ 04 · THEPLATFORMLAYER Platform evidence: pull requests, reviews, issues and logs

§ 05 · TRUST, VERIFIED Authenticating history: forged dates and rewritten pasts

§ 06 · TAKEN Exfiltration: clones, forks and the departing developer

§ 07 · COPIED Copying disputes: similarity analysis and provenance

§ 08 · THEWIDERMAP Source architecture: where else the evidence lives EVIDENCE CLONES / OLD PLATFORM CI/CD & REGISTRIES & DELETED / REPO DEPLOYS ARTEFACTS RECOVERABLE LOCAL BACKUPS / FORKS DEVICES

§ 09 · IN THE WILD Worked examples EXAMPLE1 · THECOMMITFROMTHEFUTURE ' SPAST EXAMPLE2 · THETOKENTHATOUTLIVEDTHEJOB EXAMPLE3 · THETYPOTHATTRAVELLED

§ 10 · WHEREITGOESWRONG Common mistakes and technical limitations Common mistakes

Technical limitations

§ 11 · INTERROGATORIES & DRAFTING AIDS Questions to ask · Suggested wording Ask your client Ask your opponent Ask your e Discovery / forensic provider SUGGESTED WORDING · SOURCE - CODELIMBFORTHEPRESER VAT I ON LETTER

§ 12 · QUICK CONTROL Checklist and red flags · When to involve a digital forensic expert The source-code checklist Red flags When to involve a digital forensic expert

§ 13 · COMMON QUESTIONS Frequently asked questions Can git really prove when code was written? A developer says someone else committed under their name. Possible? The opponent deleted the repository. Is the history gone? How much code similarity is enough to prove copying? Our code is commercially sensitive. Must we hand it to the opponent to fight a copying case? Does AI-generated code change any of this?

§ 14 · REFERENCE Glossary Sources and authoritative references DISCLAIMER

§ HOW A SPECIALIST LABORATORY CAN ASSIST Working with Computer Forensics Lab Speak to a forensic examiner, not a salesperson. INSTRUCTTHELAB NEWENQUIRIESEMAILE - DISCOVERY

Questions and answers

Can git really prove when code was written?

Git asserts it; corroboration proves it: commit dates are local and set table, so reliable dating stacks the server- side anchors: push times, PR and CI records: with clone triangulation and internal-consistency checks. Where those align, the dating is about as strong as digital chronology gets; where a repository stands alone, Example 1 is the caution.

A developer says someone else committed under their name. Possible?

Trivially: the author field is typed, not authenticated: so attribution climbs the ladder: which platform account pushed, from which IPs and devices, whether commits were signed, and whose working pattern the code matches. Signed commits and protected branches close the gap prospectively; forensic all y, the push record usually answers what the author field cannot.

The opponent deleted the repository. Is the history gone?

Rarely: git distributes complete history to every clone: developer machines, forks, CI checkouts, backups: and platforms retain deleted repositories for recovery windows while audit logs record the deletion itself. §8's map is the recovery plan, and the deletion event, dated against the dispute, is frequently worth more than the repository was.

How much code similarity is enough to prove copying?

No percentage answers it: volume of resemblance has innocent explanations (frameworks, generators, AI assistance); the probative class is the fingerprints: shared typos, dead code, arbitrary quirks: joined to provenance showing your version existed first. Example 3's single misspelled comment outweighed thousands of merely similar lines, which is the right intuition to keep.

Our code is commercially sensitive. Must we hand it to the opponent to fight a copying case?

No: the standard resolution is the §7 protocol: designated experts examine both codebases in controlled environments and report findings, with disclosure of the code itself confined or eliminated: confidential it y preserved on both sides while the comparison proceeds. Courts order and endorse such regimes routinely; refusing all exam in at i on while alleging copying is the position that does not survive.

Does AI-generated code change any of this?

It complicates similarity and authorship, not chronology: generated code produces innocent resemblance across unrelated projects (raising the bar to the fingerprint class) and dilutes what "the developer wrote it" means (assist an t involvement is now a standard exam in at i on question): while the dating and integrity disciplines: hash chains, server anchors, triangulation: work unchanged. Expect provenance-of-authorship questions to grow, and keep the corroboration habits that answer them. cflab. u k · e-disc ove r y. u k