Sample report. The team, people and activity are invented to show what a real report contains.

SIGDEK EVIDENCE REPORT

Larkspur Labs

1 July to 30 September 2026 · AI coding agent activity

> Between 1 July and 30 September 2026, 12 people at Larkspur Labs ran AI coding agents in 1,571 sessions across 4 repositories. Rules blocked 2 of 182 sensitive actions. One silent stretch was found.

Scope

This report covers the actions of AI coding agents that SigDek's hooks recorded on enrolled machines during the period.

Item	Detail
Organization	Larkspur Labs
Period	1 July to 30 September 2026 (America/Los_Angeles)
Generated	2 October 2026
Tools covered	Claude Code and Cursor
Enrolled	12 people, 12 machines and 4 repositories
Not covered	Work a person did by hand, without an agent; agent sessions in a repository without SigDek's hooks, or started in a subfolder of one with them, on a machine without the plugin; machines not signed in to the team, such as cloud agents and CI runners

1. Inventory

SOC 2 CC6.1

Two agent tools were in use, by 12 people on 12 machines. Four used both.

Tool	People	Sessions	Versions seen
Claude Code	8	788	Not reported by the tool
Cursor	8	783	2.4.1, 2.5.3

People

Person	Agents	Machines	Sessions
daniel@larkspur.example	Claude Code	1	137
ethan@larkspur.example	Claude Code	1	138
grace@larkspur.example	Claude Code	1	126
hannah@larkspur.example	Claude Code	1	136
jonah@larkspur.example	Cursor	1	132

Sample report. The team, people and activity are invented to show what a real report contains.

Person	Agents	Machines	Sessions
leah@larkspur.example	Cursor	1	128
marcus@larkspur.example	Cursor	1	117
maya@larkspur.example	Claude Code, Cursor	1	126
omar@larkspur.example	Claude Code, Cursor	1	125
priya@larkspur.example	Cursor	1	131
sofia@larkspur.example	Claude Code, Cursor	1	137
tom@larkspur.example	Claude Code, Cursor	1	138

Machines

Machine	OS	Owner	Recorder	Records in period
daniel-mbp `dev_fb4357e77f6ac90466eb9334`	macOS	daniel@larkspur.example	0.1.0	3,015
ethan-mbp `dev_a8d6ac901905ceefb3471f57`	macOS	ethan@larkspur.example	0.1.0	2,860
grace-pc `dev_848c9ba48f21166e0203a339`	Windows	grace@larkspur.example	0.1.0	2,793
hannah-pc `dev_d5f95b5efd957f0626f7e363`	Windows	hannah@larkspur.example	0.1.0	2,837
jonah-mbp `dev_b99613f805e9f97a8ed8555f`	macOS	jonah@larkspur.example	0.1.0	3,002
leah-linux `dev_6a11281c54ae6ef1e86e39bc`	Linux	leah@larkspur.example	0.1.0	2,687
marcus-mbp `dev_7b443f9bd534b79d1a7578d8`	macOS	marcus@larkspur.example	0.1.0	2,680
maya-mbp `dev_021f6439f8cb76fdc182276f`	macOS	maya@larkspur.example	0.1.0	2,756
omar-mbp `dev_1fa9a9c67af1e7d9a24e4061`	macOS	omar@larkspur.example	0.1.0	2,846
priya-linux `dev_e372060a34fb62006c8ef12a`	Linux	priya@larkspur.example	0.1.0	2,786
sofia-mbp `dev_4b74e93b8f997d4a443213db`	macOS	sofia@larkspur.example	0.1.0	3,206
tom-mbp `dev_8ca14e10eada6ad6ae9bf3c3`	macOS	tom@larkspur.example	0.1.0	2,859

Sample report. The team, people and activity are invented to show what a real report contains.

Repositories

Repository	Records
larkspur-labs/api	9,595
larkspur-labs/data-pipeline	8,419
larkspur-labs/infrastructure	8,300
larkspur-labs/web-app	8,013

MCP servers

Agents reached 4 outside systems through MCP servers.

MCP server	Calls	People
Not named by Cursor's tool hooks	552	6
github	339	6
linear	193	4
postgres-staging	29	2

2. Recorder coverage

SOC 2 CC7.2

Recorder coverage shows whether each connected machine's recorder sent records for each working day. It measures the recorder, not the person: the recorder writes records only during agent sessions, so a working day without agent use has no records, the same as a day the recorder was off.

All 12 enrolled people reported during the period, with one silent stretch on one machine. One member has no machine connected and is not enrolled: aisha@larkspur.example.

Machine	Person	From	To	Working days
grace-pc `dev_848c9ba48f21166e0203a339`	grace@larkspur.example	10 Jul	14 Jul	3

A silent stretch is two or more working days in a row (Monday to Friday, America/Los_Angeles) on which a connected machine sent no records. Each machine is counted from the day after it connected until the day before it was revoked. Days marked as leave for the machine's owner are left out: they don't count as silent and don't break a stretch.

By person

Person	Machines	Working days counted	Days with records	Status
aisha@larkspur.example	0	0	0	Not enrolled: no machine connected
daniel@larkspur.example	1	66	66	Enrolled
ethan@larkspur.example	1	66	66	Enrolled
grace@larkspur.example	1	66	63	Enrolled
hannah@larkspur.example	1	66	66	Enrolled

Sample report. The team, people and activity are invented to show what a real report contains.

Person	Machines	Working days counted	Days with records	Status
jonah@larkspur.example	1	66	66	Enrolled
leah@larkspur.example	1	66	66	Enrolled
marcus@larkspur.example	1	62	62	Enrolled
maya@larkspur.example	1	66	66	Enrolled
omar@larkspur.example	1	66	66	Enrolled
priya@larkspur.example	1	66	66	Enrolled
sofia@larkspur.example	1	66	66	Enrolled
tom@larkspur.example	1	66	66	Enrolled

By machine

Machine	Person	Counted	Working days	Days with records	Silent days
daniel-mbp dev_fb4357e77f6ac90466eb9334	daniel@larkspur.example	1 Jul to 30 Sep	66	66	0
ethan-mbp dev_a8d6ac901905ceefb3471f57	ethan@larkspur.example	1 Jul to 30 Sep	66	66	0
grace-pc dev_848c9ba48f21166e0203a339	grace@larkspur.example	1 Jul to 30 Sep	66	63	3
hannah-pc dev_d5f95b5efd957f0626f7e363	hannah@larkspur.example	1 Jul to 30 Sep	66	66	0
jonah-mbp dev_b99613f805e9f97a8ed8555f	jonah@larkspur.example	1 Jul to 30 Sep	66	66	0
leah-linux dev_6a11281c54ae6ef1e86e39bc	leah@larkspur.example	1 Jul to 30 Sep	66	66	0
marcus-mbp dev_7b443f9bd534b79d1a7578d8	marcus@larkspur.example	1 Jul to 30 Sep	62	62	0
maya-mbp dev_021f6439f8cb76fdc182276f	maya@larkspur.example	1 Jul to 30 Sep	66	66	0
omar-mbp dev_1fa9a9c67af1e7d9a24e4061	omar@larkspur.example	1 Jul to 30 Sep	66	66	0
priya-linux dev_e372060a34fb62006c8ef12a	priya@larkspur.example	1 Jul to 30 Sep	66	66	0
sofia-mbp dev_4b74e93b8f997d4a443213db	sofia@larkspur.example	1 Jul to 30 Sep	66	66	0
tom-mbp dev_8ca14e10eada6ad6ae9bf3c3	tom@larkspur.example	1 Jul to 30 Sep	66	66	0

Sample report. The team, people and activity are invented to show what a real report contains.

Marked as leave

Person	From	To	Working days
marcus@larkspur.example	6 Jul	9 Jul	4

3. Activity

SOC 2 CC7.2

Agents took 27,238 recorded actions, 50% of them file edits.

By person

Person	Sessions	Commands	File edits	Web requests	MCP calls
sofia@larkspur.example	137	560	1,304	122	103
daniel@larkspur.example	137	530	1,160	121	114
jonah@larkspur.example	132	513	1,169	128	130
ethan@larkspur.example	138	502	1,126	101	107
tom@larkspur.example	138	605	1,113	103	0
omar@larkspur.example	125	613	1,138	102	0
hannah@larkspur.example	136	516	1,085	104	103
grace@larkspur.example	126	492	1,127	101	126
priya@larkspur.example	131	492	1,112	108	95
maya@larkspur.example	126	482	1,092	110	112
leah@larkspur.example	128	442	1,064	101	104
marcus@larkspur.example	117	488	1,028	110	119
Total	**1,571**	**6,235**	**13,518**	**1,311**	**1,113**

By repository

Repository	Sessions	Commands	File edits	Web requests	MCP calls
larkspur-labs/api	425	1,744	3,776	353	345
larkspur-labs/data-pipeline	389	1,479	3,344	315	253
larkspur-labs/infrastructure	390	1,528	3,213	343	290
larkspur-labs/web-app	367	1,484	3,185	300	225
Total	**1,571**	**6,235**	**13,518**	**1,311**	**1,113**

Commands, file edits, web requests and MCP calls count every attempt, whatever its outcome; section 4 shows which were blocked or refused. A session that touched several repositories counts once in each, so the sessions column can add up to more than the total.

Sample report. The team, people and activity are invented to show what a real report contains.

4. Sensitive actions

SOC 2 CC6.1, CC7.2, CC8.1

Agents attempted 182 actions that matched a sensitive-action rule: 180 ran and 2 were blocked by a rule.

Rule	Matches	Ran	Blocked by rule	Refused by the agent	Failed
Secrets file accessed	41	41	0	0	0
Recursive delete	39	39	0	0	0
Package or release published	34	34	0	0	0
Remote script piped to a shell	30	30	0	0	0
Force push	27	27	0	0	0
Infrastructure change	9	9	0	0	0
No terraform destroy	2	0	2	0	0
Total	**182**	**180**	**2**	**0**	**0**

Every matching action is listed in the CSV export. Up to five examples of each rule, blocks and refusals first:

When America/Los_Angeles	Person	Tool	Repository	Outcome
Secrets file accessed (41 matches)				
25 Sep 14:13 `Read .env`	marcus@larkspur.example	Cursor	larkspur-labs/infrastructure	Ran
29 Sep 09:50 `Read config/secrets.yaml`	hannah@larkspur.example	Claude Code	larkspur-labs/api	Ran
29 Sep 10:16 `Read .env.production`	hannah@larkspur.example	Claude Code	larkspur-labs/api	Ran
30 Sep 09:53 `Read .env.production`	jonah@larkspur.example	Cursor	larkspur-labs/infrastructure	Ran
30 Sep 09:57 `Read .env`	tom@larkspur.example	Claude Code	larkspur-labs/infrastructure	Ran
Recursive delete (39 matches)				
23 Sep 09:41 `rm -rf .next/cache`	ethan@larkspur.example	Claude Code	larkspur-labs/web-app	Ran
24 Sep 11:25 `rm -rf dist node_modules`	daniel@larkspur.example	Claude Code	larkspur-labs/api	Ran
29 Sep 11:10 `rm -rf build coverage`	grace@larkspur.example	Claude Code	larkspur-labs/web-app	Ran
29 Sep 12:08 `rm -rf build coverage`	sofia@larkspur.example	Cursor	larkspur-labs/api	Ran

Sample report. The team, people and activity are invented to show what a real report contains.

When America/Los_Angeles	Person	Tool	Repository	Outcome
30 Sep 10:00	leah@larkspur.example	Cursor	larkspur-labs/web-app	Ran
`rm -rf .next/cache`				
Package or release published (34 matches)				
11 Sep 13:15	marcus@larkspur.example	Cursor	larkspur-labs/web-app	Ran
`pnpm publish --no-git-checks`				
17 Sep 13:25	omar@larkspur.example	Cursor	larkspur-labs/web-app	Ran
`npm publish --access public`				
22 Sep 09:33	omar@larkspur.example	Cursor	larkspur-labs/data-pipeline	Ran
`pnpm publish --no-git-checks`				
23 Sep 09:33	grace@larkspur.example	Claude Code	larkspur-labs/api	Ran
`npm publish --access public`				
28 Sep 11:20	grace@larkspur.example	Claude Code	larkspur-labs/api	Ran
`npm publish --access public`				
Remote script piped to a shell (30 matches)				
14 Sep 11:46	priya@larkspur.example	Cursor	larkspur-labs/data-pipeline	Ran
`curl -fsSL https://get.devtools.example | sh`				
21 Sep 10:03	jonah@larkspur.example	Cursor	larkspur-labs/api	Ran
`curl -sSL https://install.cli-tools.example/cli.sh | bash`				
21 Sep 11:23	tom@larkspur.example	Cursor	larkspur-labs/api	Ran
`curl -fsSL https://get.devtools.example | sh`				
25 Sep 09:40	grace@larkspur.example	Claude Code	larkspur-labs/infrastructure	Ran
`curl -fsSL https://get.devtools.example | sh`				
25 Sep 09:40	hannah@larkspur.example	Claude Code	larkspur-labs/infrastructure	Ran
`curl -fsSL https://get.devtools.example | sh`				
Force push (27 matches)				
15 Sep 09:44	sofia@larkspur.example	Claude Code	larkspur-labs/infrastructure	Ran
`git push -f origin fix/flaky-test`				
18 Sep 11:40	grace@larkspur.example	Claude Code	larkspur-labs/api	Ran
`git push --force-with-lease origin feature/billing-retry`				
21 Sep 09:35	hannah@larkspur.example	Claude Code	larkspur-labs/data-pipeline	Ran
`git push origin main --force`				
22 Sep 09:25	ethan@larkspur.example	Claude Code	larkspur-labs/infrastructure	Ran
`git push origin main --force`				
29 Sep 09:32	omar@larkspur.example	Cursor	larkspur-labs/infrastructure	Ran
`git push -f origin fix/flaky-test`				

Sample report. The team, people and activity are invented to show what a real report contains.

When America/Los_Angeles	Person	Tool	Repository	Outcome
Infrastructure change (9 matches)				
18 Aug 11:34 `kubectl --context production rollout undo deploy/api`	priya@larkspur.example	Cursor	larkspur-labs/infrastructure	Ran
25 Aug 13:39 `terraform -chdir=envs/staging apply`	omar@larkspur.example	Cursor	larkspur-labs/infrastructure	Ran
2 Sep 10:01 `helm upgrade api ./charts/api -f values-prod.yaml`	jonah@larkspur.example	Cursor	larkspur-labs/infrastructure	Ran
17 Sep 09:33 `helm upgrade api ./charts/api -f values-prod.yaml`	daniel@larkspur.example	Claude Code	larkspur-labs/infrastructure	Ran
30 Sep 10:00 `kubectl -n staging delete pod worker-6d8f9`	jonah@larkspur.example	Cursor	larkspur-labs/infrastructure	Ran
No terraform destroy (2 matches)				
31 Aug 09:46 `terraform destroy -target=module.cache`	leah@larkspur.example	Cursor	larkspur-labs/infrastructure	Blocked by rule
3 Sep 10:08 `terraform destroy -target=module.cache`	priya@larkspur.example	Cursor	larkspur-labs/infrastructure	Blocked by rule

5. Controls

SOC 2 CC6.1, CC8.1

One block rule and 12 flag rules applied to every signed-in machine during the period. The rules changed once; the table shows when each was in force.

Rules in force during the period

Rule	Effect	In force
No terraform destroy	Block	From 31 Jul
Destructive database statement	Flag	Whole period
Force push	Flag	Whole period
Git history rewritten	Flag	Whole period
Infrastructure change	Flag	Whole period
Package or release published	Flag	Whole period
Privileged command	Flag	Whole period
Recursive delete	Flag	Whole period
Remote script piped to a shell	Flag	Whole period
Secrets file accessed	Flag	Whole period
Secrets file read from the shell	Flag	Whole period

Sample report. The team, people and activity are invented to show what a real report contains.

Rule	Effect	In force
SigDek's own setup changed	Flag	Whole period
SigDek's own setup changed from the shell	Flag	Whole period

Control	What the record shows
Block rules in force	1: No terraform destroy
Flag rules in force	12
Rule changes	1, listed below
Actions blocked by a rule	2, each counted in section 4
Sessions run with permission checks off	1, by daniel@larkspur.example
Settings changed during a session	9, across 5 people

Rule changes

When	Change	Rule	Effect	By
31 July 2026	Added	No terraform destroy	Block	maya@larkspur.example

The last two rows of the table cover Claude Code only. Cursor's hook data carries no permission mode, so the same check cannot be made there.

6. Integrity

SOC 2 CC7.2

SigDek recomputed the hash of all 34,327 records in the period, compared every stored field with the record as it arrived, and checked that each record links to the one before it on its machine, including the records just before and just after the period. No record was changed or removed after SigDek received it, and none is missing from any machine's chain.

Check	Result
Records in period	34,327: 27,238 actions, 3,938 prompts, 3,142 session starts and ends, 9 settings changes
Hashes recomputed when this report was generated	All matched
Stored fields compared with the record as it arrived	All matched
Hash chain	Unbroken on every machine, across the start and end of the period
Problems noted when records arrived	None
Recorder versions (latest each machine reported)	0.1.0
Prompt text stored	No. Hash and length only
Secrets stripped before upload	126 values

Sample report. The team, people and activity are invented to show what a real report contains.

Seven limits apply to every report of this kind:

- The recorder runs on the developer's machine, and anyone in control of the machine can switch it off. Recorder coverage lists a machine that stops recording for two or more working days in a row. A single working day without records counts as a silent day in the machine table but isn't listed, and looks the same as a day without agent use; switching off for one session or one repository doesn't show.
- Agent sessions in repositories without SigDek's hooks, and sessions started in a subfolder of a repository set up with `sigdek init`, are recorded only on machines with the plugin.
- Records are tamper-evident once SigDek has received them. Before upload, someone with access to the machine can rewrite its local log. After upload, a change made on the machine, or a gap or fork in what it sends, is caught. Someone able to rewrite SigDek's own database consistently could hide a change from later reports, but not from a report you kept.
- Files a person attaches to a Claude Code prompt with `@` are not recorded.
- Cursor's inline Tab completions are not recorded.
- Agents on machines no one has signed in with `sigdek login`, such as Cursor's cloud agents, CI runners and hosted sandboxes, record only on that machine and send nothing to SigDek. Only the built-in rules apply there, and they block nothing.
- The record shows what an agent did. It does not judge whether the resulting code is correct or secure.

Chain heads at the end of the period

Each machine's last record in the period, with its hash, which depends on every record before it on that machine. The next record must link to this hash, and every later report checks that these records are still stored with these hashes. Keep this report: someone able to rewrite SigDek's database could make the stored chain consistent again, but not change this copy, so comparing this table with a later report for the same period shows whether any of these records changed.

Machine	Person	Last record	Hash of last record
daniel-mbp `dev_fb4357e77f6ac90466eb9334`	daniel@larkspur.example	3,015	`37d97a32cd08a9c988820b1667ecb79f` `02a6ac3d13e736cd0548438c7fa2bd98`
ethan-mbp `dev_a8d6ac901905ceefb3471f57`	ethan@larkspur.example	2,860	`83588070a65256cc64da9175c2aba783` `23424493cf67a14d32d349dea04525e5`
grace-pc `dev_848c9ba48f21166e0203a339`	grace@larkspur.example	2,793	`11464e6728aed866e08a5a170a1822ac` `bab2a2e1c92c7d96b5b75a9aad5a50cd`
hannah-pc `dev_d5f95b5efd957f0626f7e363`	hannah@larkspur.example	2,837	`0ef9af3ea63f63023b10a2b36ffc115c` `bce2e7db0de7e0f18a1f7cd51d0a8772`
jonah-mbp `dev_b99613f805e9f97a8ed8555f`	jonah@larkspur.example	3,002	`94e291d34d2205505f78c96526e31e33` `bfa40ce9fdc3e205e9789a84c33ce7fa`
leah-linux `dev_6a11281c54ae6ef1e86e39bc`	leah@larkspur.example	2,687	`23dc3005d13ede3bd6ce8d98c7a47548` `1ef91de4ea19b08974b175aa0af879da`
marcus-mbp `dev_7b443f9bd534b79d1a7578d8`	marcus@larkspur.example	2,680	`1e99a3eaf314fb53ee66afb2a03eeda3` `0acc7f4c9323eda8743fd8db4d965677`
maya-mbp `dev_021f6439f8cb76fdc182276f`	maya@larkspur.example	2,756	`5e1a9348a914d2c5652aaeadf5852b3a` `3ccee775a37d561e378fde046d65e4a7`
omar-mbp `dev_1fa9a9c67af1e7d9a24e4061`	omar@larkspur.example	2,846	`3c361a9c1131337a68918bb603cbb61d` `40112220e2d9a06bf8c4db924b18676d`

Sample report. The team, people and activity are invented to show what a real report contains.

Machine	Person	Last record	Hash of last record
priya-linux `dev_e372060a34fb62006c8ef12a`	priya@larkspur.example	2,786	`f3f15a764b1f8932de2dd4bffc6182e3` `859124eb6019203f885ca7362038d5d2`
sofia-mbp `dev_4b74e93b8f997d4a443213db`	sofia@larkspur.example	3,206	`2fdc92114ba1ef9ce4550cea8e968e7c` `1bd6dbb219b302691f183eaf0d75a38f`
tom-mbp `dev_8ca14e10eada6ad6ae9bf3c3`	tom@larkspur.example	2,859	`da5757b63bd55d7c1a411413105383bf` `cf119940a46c284e3a1abf1d9791a24b`

Report rep_sample. The SOC 2 labels show which trust services criteria each section can support; your auditor decides what evidence it accepts. To check this file has not changed since it was generated, open https://app.sigdek.com/verify and drop it on the page. Generated by SigDek (sigdek.com), a product of Tekton Software LLC.