authzed

DETERMINISTIC AUTHORIZATION FOR CODING AGENTS

Your AI has **too much access.**

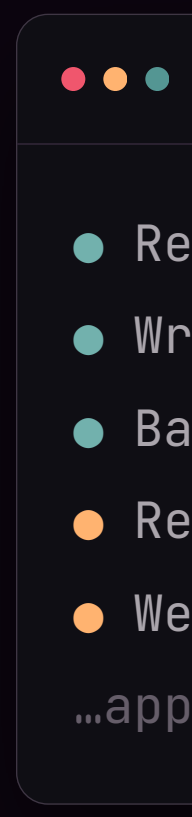
Sohan Maheshwar
Developer Relations · AuthZed

THE NEW NORMAL

Plan together, then let it run unattend

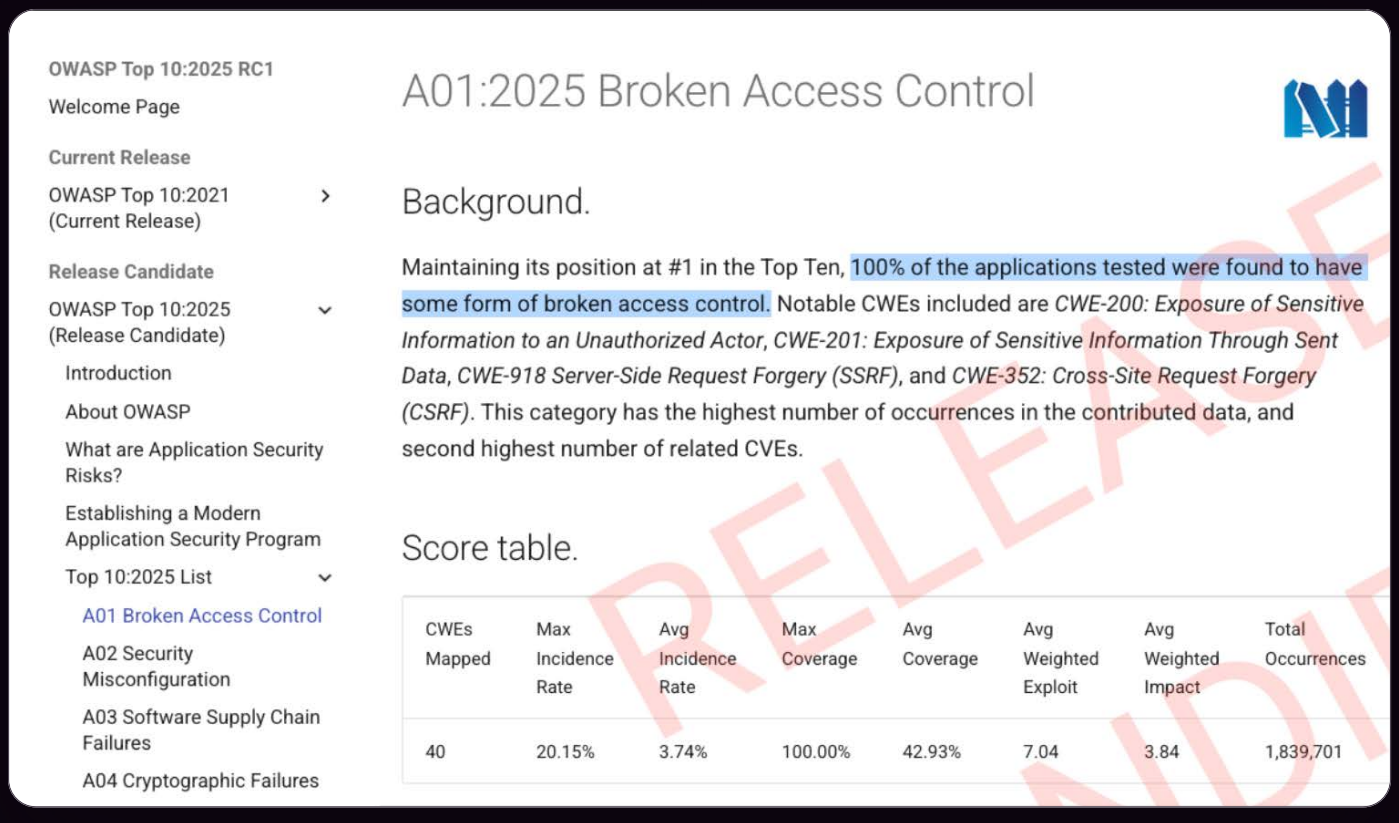
Plan with the agent, then let Claude Code run autonomously

With no boundaries, it takes any action its context suggests - any file, any command, any host.



THE STAKES

Broken access control is still #1.



Ranked #1

100% of a
access co

Now hand

owasp.org/Top1

CLAUDE CODE · AUTO-PERMI

false-negative rate on dangerous actio
analysis, which is upfront ab

> *"Claude can make mistakes that allow harmful c*
> *recommended to only use in isolated environme*
>
> — Anthropic documentation

https://www.anthropic.com/engineering/claude-code-auto-mode

THE MISMATCH

Authorization demands **deterministic**

❌Probabilistic judgment❌

A model weighs context and returns its best guess at what's safe. Run it again and the answer can change.

"probably fine" → 83% of the time

✅D

A po
ques

chec

SECTION 01 / 06

A primer on authoriza

MODELS OF ACCESS CONTROL

Four ways to model access

ACL

Access lists

A list, per object, of who can do what. Accurate, but it doesn't scale.

RBAC

Roles

Permissions grouped into roles. Simple but does not scale

ABAC

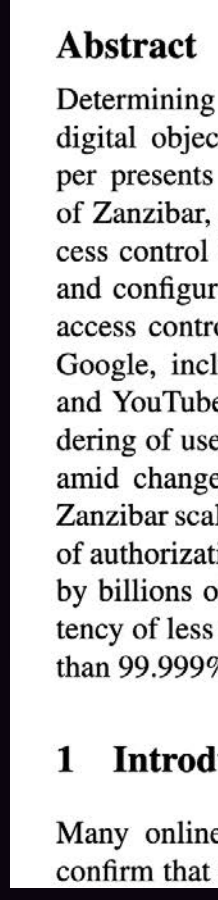
Attrib

Rules resour attribu hard to

GOOGLE ZANZIBAR · REBAC

Google Zanzibar models access as **relationships.**

- Globally distributed,> 10M+ checks per second.
- AuthZ system behind Google Docs, YouTube, Maps, Photos & more
- To the developer, it's just an API:
  Is this actor allowed to perform this action on this resource?



Abstract

Determining
digital objec
per presents
of Zanzibar,
cess control
and configur
access contr
Google, incl
and YouTube
dering of use
amid change
Zanzibar scal
of authorizat
by billions o
tency of less
than 99.999%

1 Introd

Many online
confirm that

REBAC · THE VOCABULARY

Three building blocks, **one tuple.**

User

A natural person or service account.

`user:10957135`

Object

Any non-user entity, with a typ
prefixed id.

`doc:123 · folder:plans`

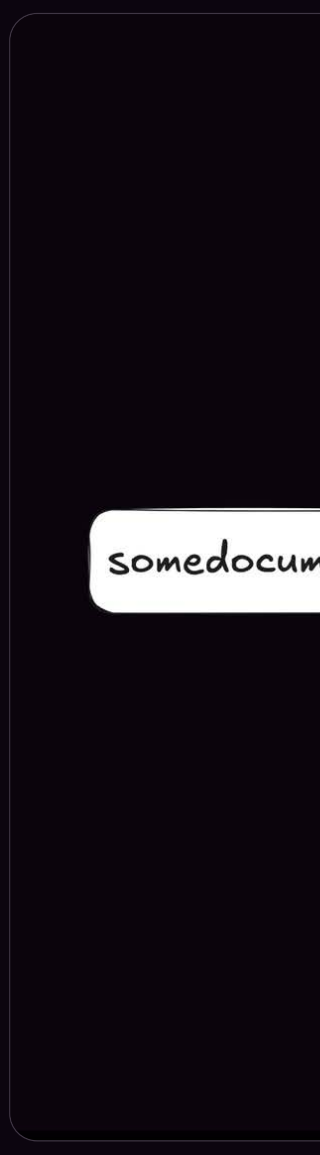
A RELATION TUPLE

`document:123#owner@user:3`

REBAC · THE SHAPE OF IT

Chain the tuples and you get a **graph**

- Objects relate to other objects — and to other objects' relations.
- Chain enough tuples and you've built a directed graph — a DAG.
- A permission is just reachability in that graph.

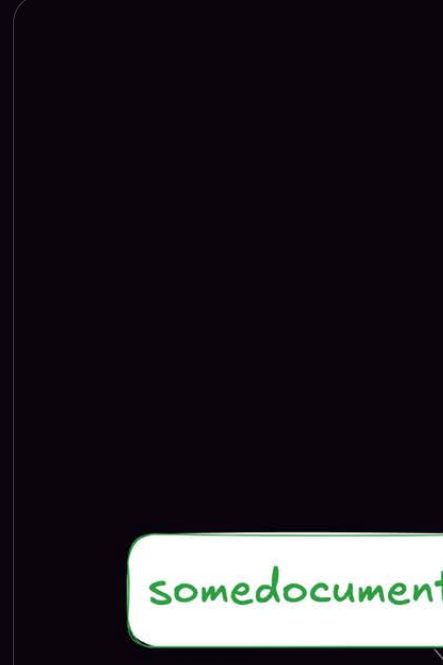


REBAC · ANSWERING A CHECK

A check is a **walk through the graph**

THE QUESTION

```
check(
  somedocument,
  "reader",
  jill
)
```



STAYS FAST: Parallel subproblems · Cache everything · Dedupe requests

WHY RBAC ISN'T ENOUGH

A role can't say **path, command, or**

`role: developer`

One coarse bucket. All-or-nothing.

WHAT AN AGENT AC

Write only ur

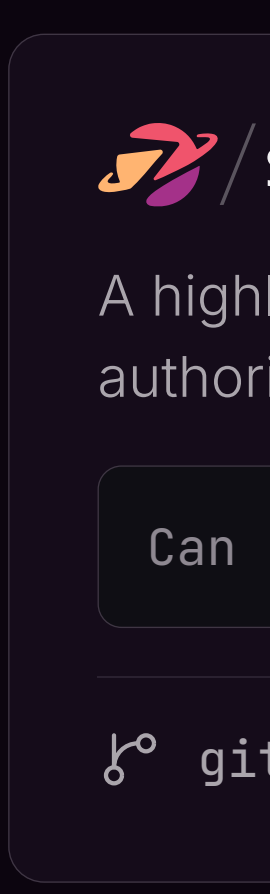
Run only go

Fetch only g

THE OPEN-SOURCE ZANZIBAR

Introducing SpiceDB

- Google published the Zanzibar whitepaper — the model, not the code.
- SpiceDB is its open-source implementation — the engine you actually run.
- In production at companies like OpenAI and Zoom, with contributors from the Fortune 500.



SECTION 02 / 06

Introducing SpiceBox

WHAT IT IS

SpiceBox

SpiceDB (the authz engine)
+ Sandbox (OS containment)

Fine-g
codin

$ br

Apach

THE CORE IDEA

Permission checks are still determinis

01

You decide

Describe what the agent may do, in plain English.

"Read & write Go under src/. Run go test. Fetch github.com."

02

SpiceBox translates

AI maps it to structured Spic permissions — you review a approve.

```
file rw src/**/*.go
bash go test   web githu
```

FROM DESCRIPTION TO ENFORCEMENT

Six steps, **one session.**

THE TASK

```
the ask

"Add retry logic to the Go API client."
```

One ordinary coding task — watch it pass through all six steps, scoped the whole way.

service-api · src/

01 Describe

02 Translate

03 Review

04 Enforce

05 Run

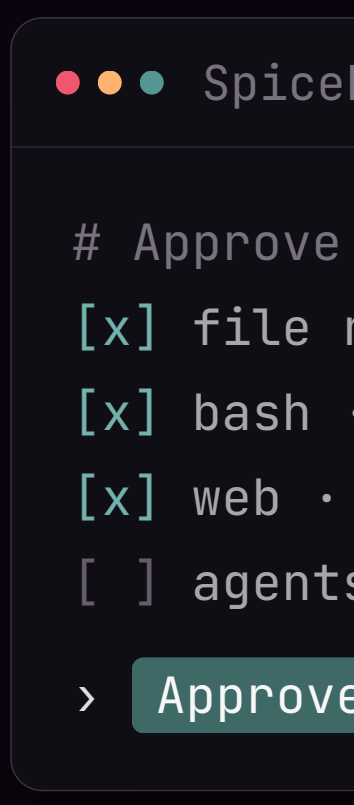
06 Cleanup

QUICK START

One command to launch

```
$ spicebox claude
```

1. Describe what Claude can do
2. Review the translated permissions
3. Approve, and launch Claude Code



SECTION 03 / 06

Three layers of enforc

OVERVIEW · DEFAULT-DENY

Every tool call runs the same **gauntlet.**

Tool call

APPLICATION

Hook server

Checked against SpiceDB before it runs.

OPERATING SY

OS sandbox

Filesystem limits kernel.

Only explicitly granted permissions pass.

LAYER 1 APPLICATION

The hook server checks **before** the

Default-deny. Only what you granted passes. Everything else is blocked.

Cross-tool. Read, Glob and Grep all check the same file permission.

Out-of-process. The model can't reach or rewrite the rules.

ONE

Re

LAYER 2 OPERATING SYSTEM

The kernel is the **backstop.**

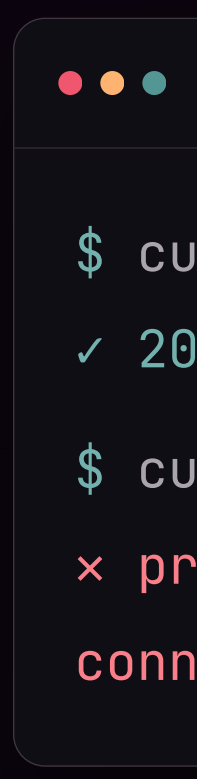
- A `sandbox-exec` profile restricts the process at the OS level.
- No write permission granted? The working directory is **read-only.**
- Bash tricks like `echo > file` are blocked by the kernel anyway.

```
# ho
$ ec
sand
Oper
```

LAYER 3 NETWORK

Egress goes through an **allowlist.**

- All outbound traffic is routed through a **localhost proxy.**
- Connections are allowed only to **permitted domains.**
- A `curl evil.com` via bash dies at the proxy, not the model.



WHAT YOU CAN SCOPE

Five permission types

Blanket	Category-wide access
File	Path prefix + suffix filter
Bash	Command prefix matching
Web	Domain-based URL filtering
MCP	Tool, prefix, or parameter value

APPROVAL CONTROL

Granted isn't the same as **silent.**

[auto]

Proceeds silently

Matching tool calls run with no interruption. Reads default here.

Pro

The
defa

SECTION 04 / 06

The threat model

ENFORCEMENT THAT HOLDS UNDER ATTACK

Deterministic checks survive a **compromis**

Assume the model is compromised.

Prompt injection can hijack the agent's reasoning. It doesn't matter — every check runs out-of-process and deterministically, so a hijacked model still can't exceed its grant.

Prompt injecti
denied regardle

Bash tool wor
kernel blocks th

Data exfiltrati
host isn't on the

Privilege esca
rules live where

Alternative to
Grep can't reac

BEING HONEST ABOUT THE EDGES

What SpiceBox does **not** protect aga

Bash is Turing-complete

Broad bash prefixes can be gamed. The OS sandbox — not the hook — is the backstop.

Trans

Natura
you ap

No suffix exclusions yet

"Read all files except .env" approximates to blanket read — with a warning.

Sand

Other
OS-le

SECTION 05 / C

Live de

Scoping paths · network isolation · a crypto

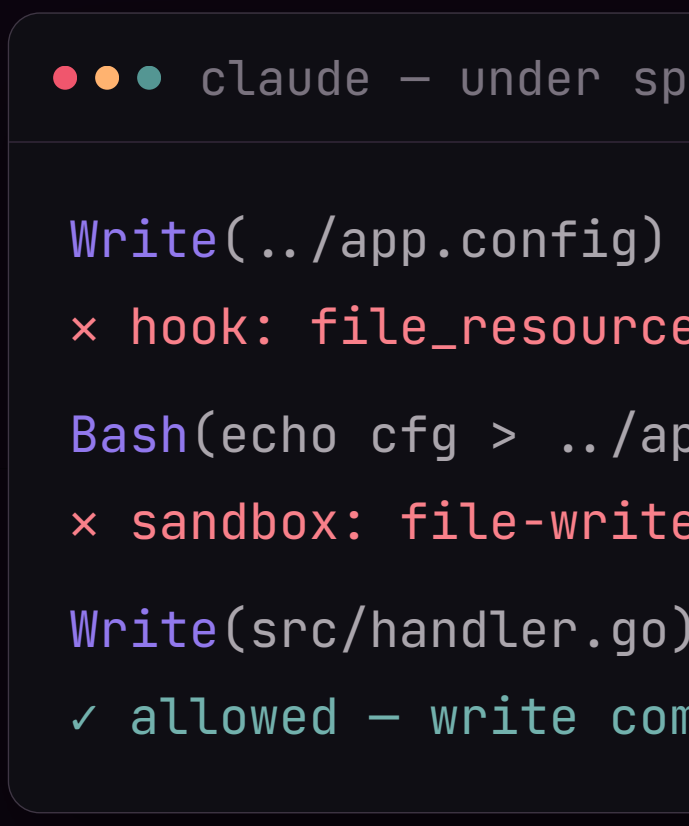
DEMO 01 · PATH SCOPING

An agent that can't write outside its lane

POLICY

```
write *.go
under src/ only
```

Anything outside `src/` is denied — twice.



DEMO 02 · NETWORK ISOLATION

What network isolation looks like

On the allowlist

```
$ curl github.com/authzed/...
✓ 200 OK
```

B

```
$ c
× p
```

Routed through the proxy whether it's a tool call

DEMO 03 · MID-SESSION GRANT

The agent asks for more. **A human gra**

/spicebox-grant

```
# Claude requests:
"Also let me run npm test"

# translated · HMAC-signed
+ bash npm test [confirm]

Grant this permission? › Approve   Deny

✓ relationship written — live now
```

SECTION 06 / 06

Beyond Claude Code

WHY NOT JUST SETTINGS.JSON?

Tool-level rules leave **gaps.**

- **No cross-tool policies**
  → SpiceBox: shared file checks
- **No parameter constraints**
  → SpiceBox: MCP value filtering
- **Static for the session**
  → SpiceBox: cryptographic grants

- **Allo**
  → S
- **No**
  → S
- **Eve**
  inte

WHAT TO TAKE AWAY

Four things to remember

01 Model judgment is not enforcement. Keep the two separ

02 Deterministic, out-of-process checks hold even under pr

03 ReBAC scopes by path, command, and domain — RBAC

04 Let the AI draft the policy — but a human approves it.

NEXT STEPS

Try it, read it, talk to us

SpiceBox
`github.com/authzed/spicebox`

Install
`brew install --cask authzed/tap/spicebox`

AuthZed

Thank you!

Sohan Maheshwar

linkedin.com/in/sohanmaheshwar