



From **CVE** to **Fix** in Minutes

AI-augmented security lifecycle management for Linux distributions

Triage



Patch



Review

Balakumaran Kannan · Kanishk Bansal

Microsoft, Azure Linux



LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA / OCT. 5-7, 2026

Contact : balakumaran.kannan@microsoft.com
kanbansal@microsoft.com

The CVE queue grows faster than maintainers do.



Triage

Does it even apply to what we ship?



Patch

The fix targets a tree 3+ releases newer



Review

15–30 minutes per PR.
Every PR.

~5

CVEs a week per engineer, by hand. The queue always won.

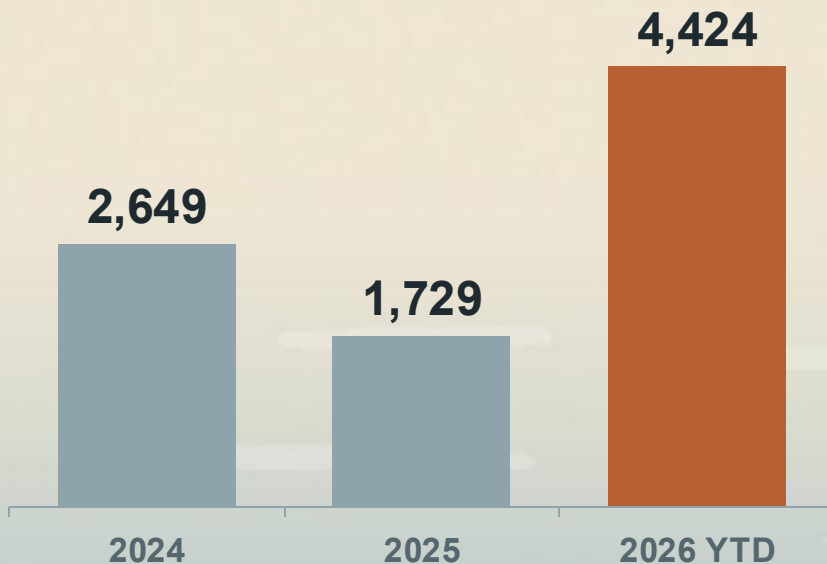


LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026

THE TREND

CVEs on Azure Linux 3.0 keep climbing.



2.6×

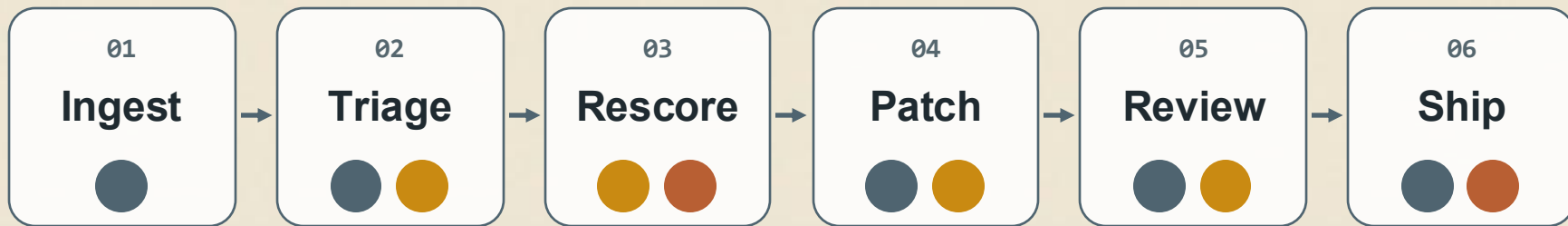
2026 so far vs. all of 2025

~470

CVEs a month in 2026 (vs ~145 in 2025)

Azure Linux 3.0, excluding kernel · 2026 to 5 Oct

One pipeline. Six stages. Three roles.



● AI proposes

● Rules decide

● Humans stay accountable



LINUX
PLUMBERS CONFERENCE PRAGUE, CZECHIA / OCT. 5-7, 2026



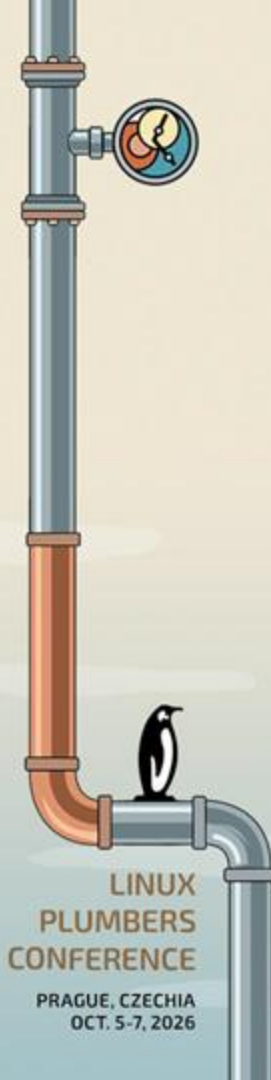
01 Triage

Does this CVE even apply to what we ship?



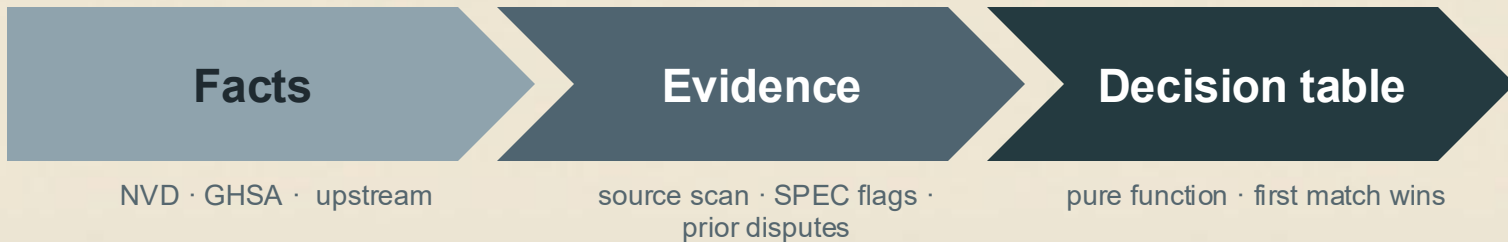
LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026



TRIAGEPILOT

Collect facts. Build evidence. Decide once.

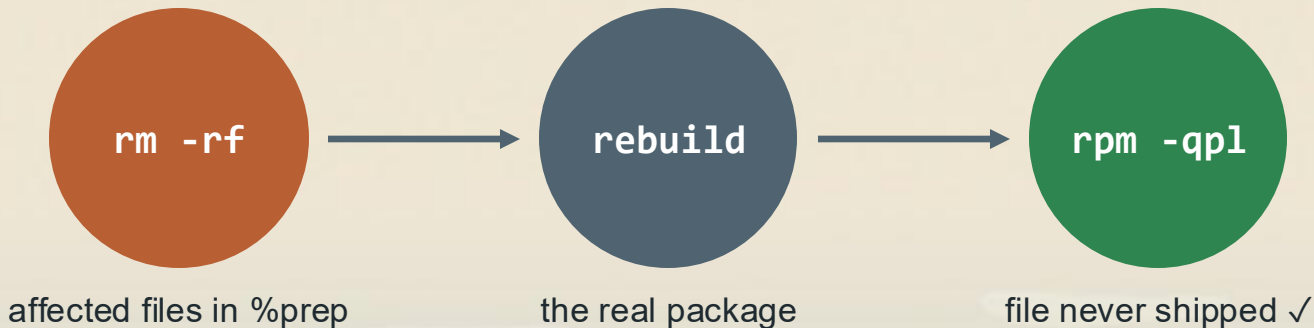


No AI in the verdict. Same input → same answer → auditable.

AI only suggests: patch links, file names, a dispute-tag hint

EARNING TRUST

“Not affected” has to be provable.



Measured, too: every AI verdict is logged next to the human outcome, so we track acceptance and tag match over time.

** Compiled packages only. Not used for interpreted languages such as Python: deleting a source file does not break the build, so a green build proves nothing.*

BUILD-FLAG DISPUTE · WORKED EXAMPLE

cmake · CVE-2026-82209

A curl bug (8.2 HIGH) flagged on cmake, because cmake ships its own copy of curl

cmake 3.30.3 source

~~Utilities/cmcurl~~ bundled curl

`rm -rf` in %prep: never compiled

`--system-curl`

libcurl.so.4

the system curl package

patched once, in curl

SPEC %prep
`rm -rf Utilities/cmcurl`

Build flag
`./bootstrap --system-curl`

RPM metadata
Requires: libcurl.so.4

DISPUTE (accepted, tag NOSHIP) The vulnerable bundled copy is never built or shipped.

A REAL TRIAGE RUN

coredns · CVE-2026-35579

```
● ● ●  
$ triagepilot --cve CVE-2026-35579 --package coredns  
ingest NVD · GHSA · Ubuntu · Debian → High, < 1.14.3  
scan core/dnsserver/server_grpc.go present  
spec 1.11.4 · already_patched=True  
ai existing patch adds dns.TsigVerify() ✓  
prior no dispute match  
verdict rule: affected_file_already_patched  
▶ DISPUTE · tag=FIXED
```

VERDICT

DISPUTE

FIXED

Already fixed by an existing
SPEC patch.
The rule decided;
AI only confirmed it.

LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026

DISTRO-SPECIFIC RESCORING

CVSS scores the **code**. We score our **build**.

The CVE applies to us. How bad is it, as we ship it?



Opt-in flag?

Must be explicitly enabled →
still fixed, lower score



Default listener?

Network or local only? →
attack vector



Already privileged?

Gains nothing new? →
privileges, impact



Hardening?

FORTIFY · SSP · PIE: code
exec → abort



Sandboxed?

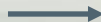
systemd limits, non-root →
blast radius



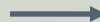
Not rescored

Not shipped or compiled out
→ DISPUTE in triage

AI drafts



Human approves



Sets our SLA

not published externally

RESCORING · WORKED EXAMPLE

moby-engine · CVE-2026-84304

Memory-exhaustion DoS in the gRPC server vendored into the Docker daemon

UPSTREAM

8.7 HIGH

AV:N · PR:N



AS WE SHIP IT

LOW

HIGH if swarm manager

UPSTREAM ASSUMES

DEFAULT DOCKER INSTALL



Reach

Network

Local Unix socket only



Privileges

None

root or docker group



Gain

Daemon DoS

Nothing: already root



Remote

Always there

Only after docker swarm init

Still patched. Rescoring changes how urgent the fix is, not whether we fix it.

LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026

02

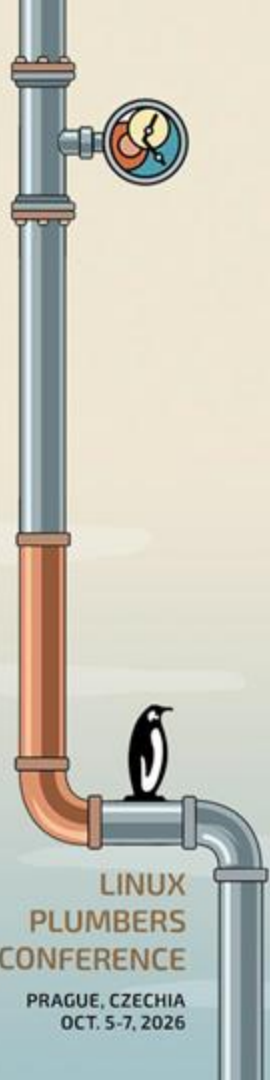
Patch

Take the cheapest fix that works.



LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026



UPGRADE FIRST

Two boring fixes first. No model involved.

①

x.y.z

Micro-version upgrade

- ✓ Upstream-first: nothing to carry
- ✓ One bump, many CVEs

②

patch -p1

Clean cherry-pick

- ✓ Upstream author + reference kept
- ✓ SPEC edits generated

Every fix → PR + CI build → automated review

LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026

PATCH BACKPORTING

Rewriting upstream CVE fixes for Azure Linux stable releases

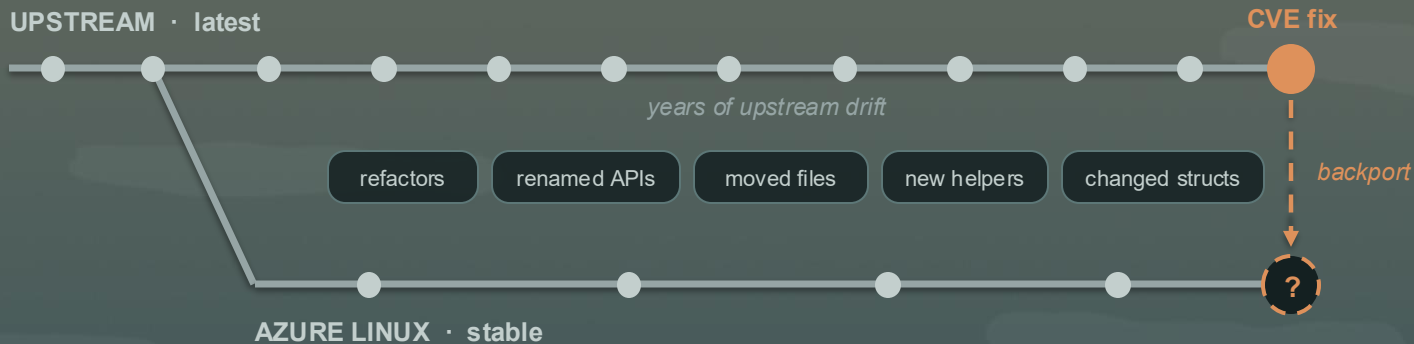


LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026

One fix, two very different codebases

A CVE patch written for the latest upstream has to be rewritten for our older stable code



A correct backport must...

Fix the CVE

Close every vulnerable path in the old code, not just the lines upstream touched.

Fit the old code

Rewrite against older APIs, data structures and file layout.

Add no new issues

No regressions and no behaviour change beyond the fix.



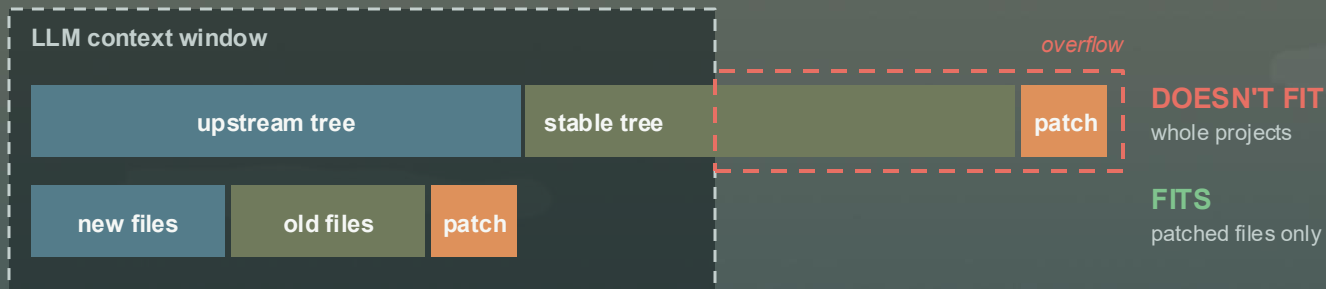
LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026

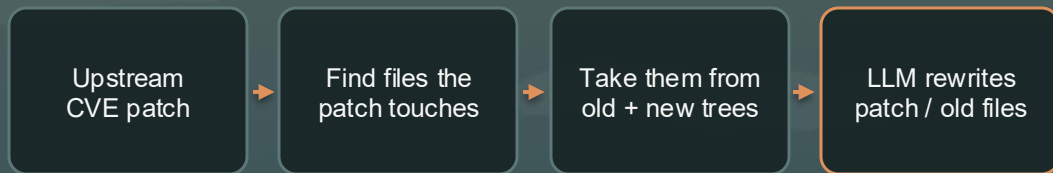
First attempt: squeeze it into one prompt

Oct – Nov 2024

No coding agents yet — the LLM only saw what fitted in its context window



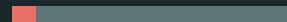
So we tried



Variations: rewrite the diff · edit the old files directly

< 10%

success rate



only trivial patches made it



**LINUX
PLUMBERS
CONFERENCE**

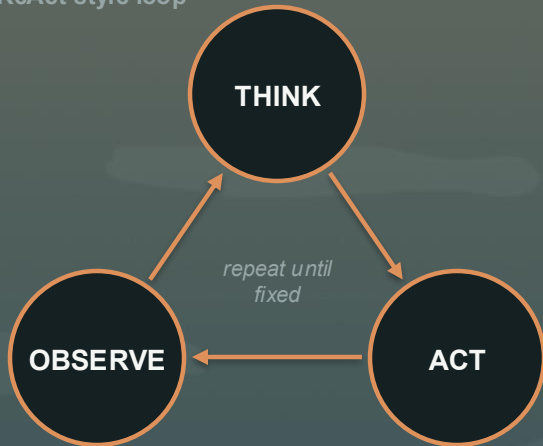
PRAGUE, CZECHIA
OCT. 5-7, 2026

SWE-agent: think, act, observe, repeat

Princeton · NeurIPS 2024

Fix code the way developers do: many small steps, a little context at a time

ReAct-style loop



Each step reads a little, then picks the next move

- 1 THINK** locate the PVSystem class
ACT search_dir "class PVSystem"
OBS 1 match: pvlib/pvsystem.py
- 2 THINK** read the constructor
ACT open pvlib/pvsystem.py
OBS 100-line window, line numbers
- 3 THINK** patch the constructor
ACT edit <start>:<end> ... end_of_edit
OBS lint OK → updated view

adapted from the paper's pvlib-python example

"Humans can flexibly ignore unnecessary information, whereas all content has a fixed cost in memory and computation for LMs."

— Yang, Jimenez et al., SWE-agent (arXiv 2405.15793)



LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026

Tools built for LLMs, not humans

GPT-4 Turbo · Claude 3 Opus

Agent-Computer Interface (ACI): fewer options, smaller outputs, built-in guardrails

Linux shell (built for humans)

`cat file.py`

1,000+ lines flood the context

`grep / find`

dozens of flags, noisy output

`sed -i / echo >`

silent, error-prone edits

SWE-agent ACI (built for LLMs)

`open · scroll · goto`

100-line window with line numbers

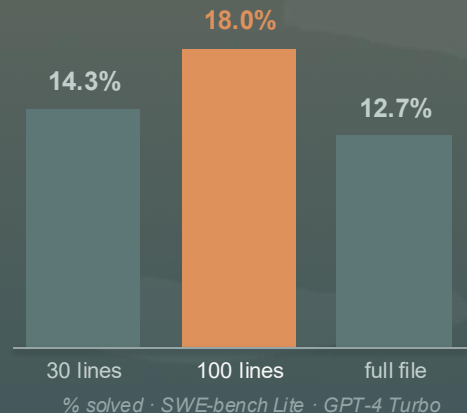
`find_file · search_dir`

≤ 50 results, else “refine query”

`edit <start>:<end>`

linter rejects broken edits

Viewer window size



Same model, better interface (SWE-bench Lite, GPT-4 Turbo)

Shell only

11.0%

SWE-agent

18.0%

The idea stuck

Anthropic built its Claude 3.5 Sonnet SWE-bench agent on SWE-agent (Oct 2024). Agents still read files in windows:

```
sed -n '200,300p' kernel/fork.c
```



LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026

SWE-agent, repurposed for backporting

Feb 2025

Built for SWE-bench bug fixing; we pointed the same loop at CVE backports

INPUT

Stable repo

git checkout <azl-stable>

Upstream fix

patch file or commit ID

SWE-agent loop

think → act → observe

open · scroll

find_file

search_dir

edit + lint

git

ag · bash

OUTPUT

Backported patch

rewritten for the stable tree

stock SWE-agent tools + everyday CLI

Backport success rate

Single prompt

< 10%

SWE-agent loop

> 50%

0%

50%

100%



**LINUX
PLUMBERS
CONFERENCE**

PRAGUE, CZECHIA
OCT. 5-7, 2026

Works locally, hard to ship

An autonomous agent running shell commands inside Azure Pipelines

Developer machine

WORKS — runs fine locally

deploy to CI

Azure Pipelines build agent

NO SECURE SANDBOX

SWE-agent

\$ <any command>

LLM decides what runs,
with the job's access

Harm the build servers

LLM-generated commands run directly on the pipeline servers

Trigger other pipelines

a stray command could kick off unrelated pipelines

Fragile model layer

litellm (used by SWE-agent) kept breaking as provider APIs changed



LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026

Microbots: a sandboxed, pluggable loop

`pip install microbots`

A generic ReAct framework: backporting is just one task we give it

MicroBot

think → act → observe

JSON: thoughts · command · done

safety validator

max_iterations · timeout

runs on the host / pipeline

prompt

reply

command

output

LLM

native SDKs

Azure OpenAI

OpenAI

Anthropic

Ollama

Code to work on

mounted READ_ONLY or READ_WRITE

host copy is never touched

mount

Ephemeral Docker sandbox

one per bot, destroyed after

bash shell

unprivileged user, no sudo

overlaysfs

writes stay in the sandbox

YAML tools

cscope, browser, sub-agent

Guardrails

- throwaway container per bot
- shell never runs as root
- read-only mount + overlaysfs
- risky commands blocked, safer hint
- iteration & time budgets

Swap any layer

LLMInterface any provider
Environment any sandbox
Tool ~15 lines of YAML
Bots Reading · Writing · Browsing
LogAnalysis · AgentBoss

Backporting: one call

```
bot = WritingBot(  
    model="azure-openai/gpt-5",  
    folder_to_mount="azl-src")  
bot.run("Backport upstream "  
        "commit <sha>",  
        timeout_in_seconds=1800)
```



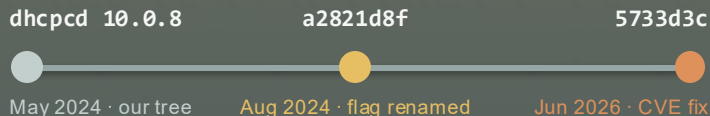
LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026

Case study: dhcpcd CVE-2026-56113

PR #17842 · AI only

A 5-hunk upstream fix, but two years of drift between it and our stable tree



40 upstream commits to src/dhcp6.c · +766 / -738 lines

```
$ git cherry-pick 5733d3c
CONFLICT (content): Merge conflict in src/dhcp6.c
```

```
$ patch -p1 < 5733d3c.patch
Hunk #1 succeeded at 2507 (offset -54 lines).
Hunk #2 succeeded at 2539 (offset -57 lines).
Hunk #3 succeeded at 2554 (offset -57 lines).
Hunk #4 FAILED at 2637.
Hunk #5 succeeded at 2593 with fuzz 1
```

Hunk 4 rejected

upstream says `IPV6_AF_PFXDELEGATION`, our tree still has `IPV6_AF_DELEGATEDPFX`

Hunk 5 fuzzed

patch skipped a context line that differs: upstream checks `!(flags & IPV6_AF_EXTENDED)`, our tree the opposite

MICROBOTS SETUP

```
upstream_patch = Mount(
    "5733d3c.patch",
    "/workdir/upstream/5733d3c.patch",
    READ_ONLY, MountType.COPY)

bot = WritingBot(
    model="azure-openai/gpt-5",
    folder_to_mount="azl-dhcpcd") # RW

bot.run(TASK,
    additional_mounts=[upstream_patch],
    max_iterations=80, timeout_in_seconds=1800)
```

INITIAL PROMPT (TASK)

Backport the upstream security fix for CVE-2026-56113 to this repository.

`/workdir/azl-dhcpcd` is `dhcpcd 10.0.8` as shipped in Azure Linux 3.0, including our earlier CVE patches. The upstream fix was written for a much newer `dhcpcd` and does not apply cleanly here.

- Start by applying the upstream patch with `patch -p1`; then study every hunk that was rejected or applied with fuzz, and adapt it to this code base.
- Make small, targeted edits. Fix the vulnerability, change nothing else.
- Make sure it still builds (`./configure && make`).



LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026

Inside the run: 14 steps, < 3 minutes

GPT-5 · Microbots

The maintainer's workflow: apply, read the reject, adapt to the old code, build

2

```
patch -p1 < ../upstream/5733d3c.patch
```

Hunk #4 FAILED, #5 fuzz 1: rejects saved

4

```
sed -n '1,200p' ../upstream/5733d3c.patch
```

reads what the fix is meant to do

5

```
sed -n '2550,2680p' src/dhcp6.c
```

finds the old flag name: IPV6_AF_DELEGATEDPFX

7

```
sed '/IPV6_AF_DELEGATEDPFX/{n;s/.../.../}'
```

ports the rejected hunk onto the old name

11

```
sed -n '2488,2565p' src/dhcp6.c
```

checks hunks 1–3: bool return, freed = true

13

```
./configure && make -s
```

builds cleanly → task_done: true

AGENT'S PATCH · src/dhcp6.c (excerpt)

```
@@ dhcp6_deprecateaddrs() @@
    struct ipv6_addr *ia, *ian;
+ #ifndef SMALL
+   bool again;
+ #endif
+ again:
    TAILQ_FOREACH_SAFE(ia, addrs, next, ian) {
        ...
        if (ia->flags & IPV6_AF_DELEGATEDPFX)
-         dhcp6_deprecatedele(ia);
+         again = dhcp6_deprecatedele(ia);
+     else
+         again = false;
        ...
        if (ia->flags & IPV6_AF_EXTENDED)
            ipv6_deleteaddr(ia);
            ipv6_freeaddr(ia);
+ #ifndef SMALL
+   /* Deletion may invalidate the next pointer so restart */
+   if (again)
+       goto again;
+ #endif
```

■ stable tree's context, kept as is ■ the ported hunk

Same change as merged PR #17842 · same +/- lines as upstream

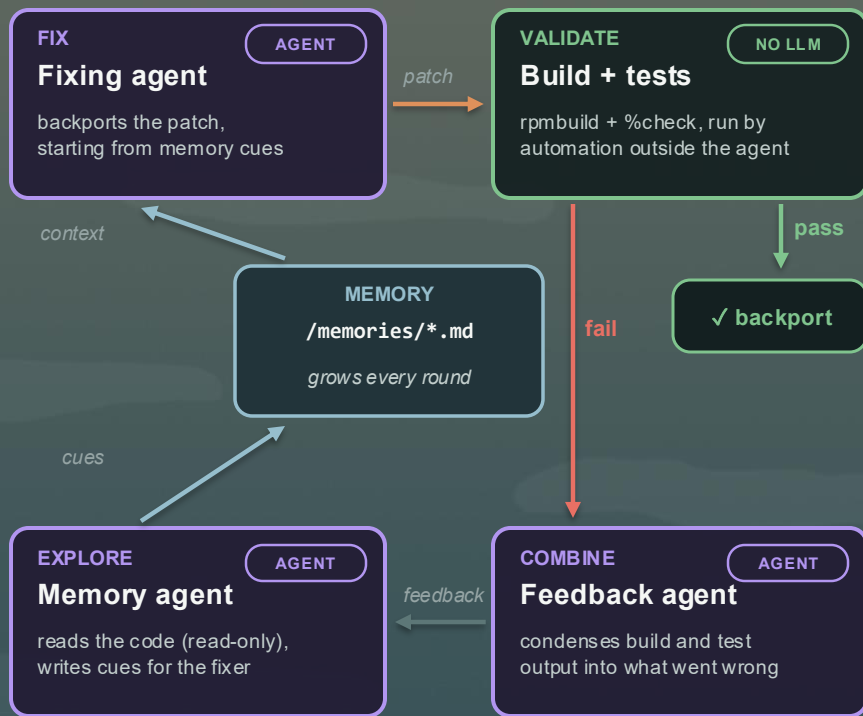


LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026

Closing the loop: every failure teaches

A deterministic validator, and code exploration split from code fixing



Ground truth, not opinion

The build and the package tests decide pass or fail, not an LLM.

Explore once, fix focused

The memory agent explores and leaves cues; the fixer doesn't restart from scratch.

Failures pay forward

Each failed round adds verified notes, so memory gets sharper.



**LINUX
PLUMBERS
CONFERENCE**

PRAGUE, CZECHIA
OCT. 5-7, 2026

■ LLM agent (Microbots) ■ deterministic automation ■ shared memory

Example: abseil-cpp CVE-2025-0838

PR #15779 · human-fixed

Fails three times, then passes, because memory keeps getting more precise

✗ Round 1

1 of 214 tests fails

reserve() left unguarded

memory at start: empty

✗ Round 2

compile error

helper used before declared

memory at start: 8.5 KB

✗ Round 3

compile error

syntax error in its own edit

memory at start: 9.5 KB

✓ Round 4

builds · 214/214 pass

guard in AllocSize()

memory at start: 13.6 KB

VALIDATOR · ROUND 1

```
[ RUN      ] Table.MaxSizeOverflow
raw_hash_set_test.cc:2685: Failure
Death test: t.reserve(overflow)
Result: failed to die.
[ FAILED ] Table.MaxSizeOverflow
```

99% tests passed, 1 tests failed
out of 214

→ feedback: build OK, test
absl_raw_hash_set_test now fails

MEMORY AGENT'S NOTES (excerpts)

after round 1

reserve(n): computes m via
GrowthToLowerboundCapacity(n) with a signed
cast; ... the int64_t cast wraps negative

after round 2

AllocSize is defined early at
raw_hash_set.h:1018 ... the forward declaration
for HashTableSizeOverflow must therefore be
placed before ~1018

after round 3

Update AllocSize ... guard overflow: if
(capacity != 0 && slot_size > ((max -
offset)/capacity)) -> HashTableSizeOverflow()

ROUND 4 · FIXER LOG

```
12:40:17 memory view /memories/...
12:40:27 patch -p1 -i ../5a0e2cb.patch
Hunk #1 FAILED at 24.
12:42:32 ✖ Add forward declaration
for HashTableSizeOverflow() early
... so AllocSize can reference it
12:42:46 ✖ Replace AllocSize with
overflow-checked version
12:51:11 task_done
```

100% tests passed, 0 tests failed
out of 214

Round 4 guards AllocSize(): $\text{slot_size} > (\text{SIZE_MAX} - \text{offset}) / \text{capacity}$ → the same condition the human engineer merged



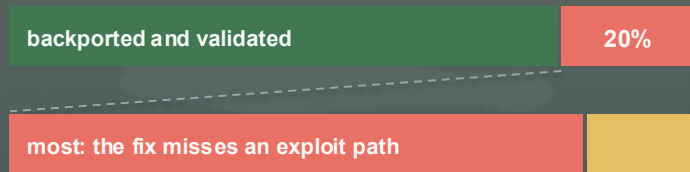
LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026

The last 20% needs the exploit

Most remaining failures build and pass every test, yet leave an attack path open

50% → 80% with the feedback loop



few: hard backports

The missing validator: a PoC exploit

Build and tests prove the patch works. Only the exploit proves the hole is closed.

abseil-cpp, round 1: built fine, reserve() path still open.

*Exploits can't go public. Public models won't write them.
Attackers' models will.*

SCALE THE EMBARGO. TOGETHER.

Coordinated disclosure is our shield. Strengthen it.

- 1 **Bring in more** distros, vendors and enterprises.
- 2 **Embargo more** critical CVEs.
- 3 **Share the PoC exploit** inside the embargo, with the fix.
- 4 **Validate every backport** against the exploit before release.

2026 brought exploits at AI speed. Build the shield at the same speed, with the same tool.



LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026

03

Review

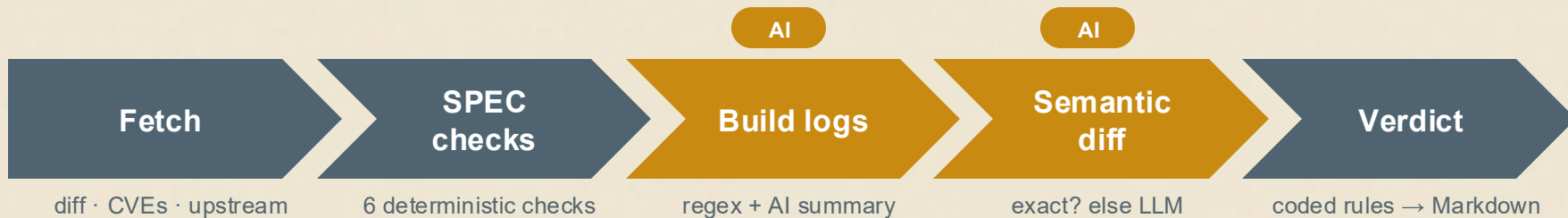
A review a human can verify in a minute.



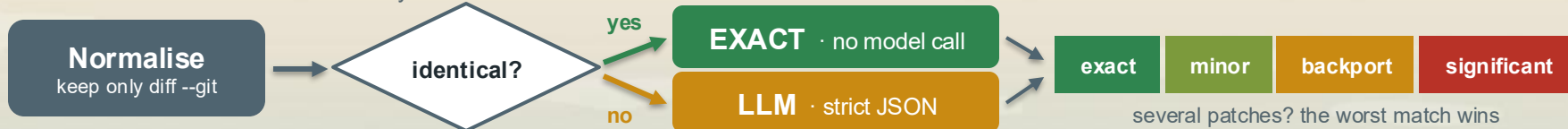
LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026

Five stages. Only **two** ask a model.



SEMANTIC DIFF ask the model only about real differences



The model gives a **signal**. Code gives the **verdict**.



LINUX

PLUMBERS CONFERENCE

PRAGUE, CZECHIA / OCT. 5-7, 2026





Questions & Thank you

AI **proposes** · Rules **decide** · Humans stay **accountable**

THE TEAM

Balakumaran Kannan · Kanishk Bansal · Kavya Sree Kaitepalli · Kshitiz Godara
Madhur Aggarwal · Rohit Rawat · Sivashanmugam Kannan

Microsoft, Azure Linux



LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA / OCT. 5-7, 2026

LinkedIn - <https://www.linkedin.com/in/balakumaran-kannan/>
<https://www.linkedin.com/in/kanishkbansal>