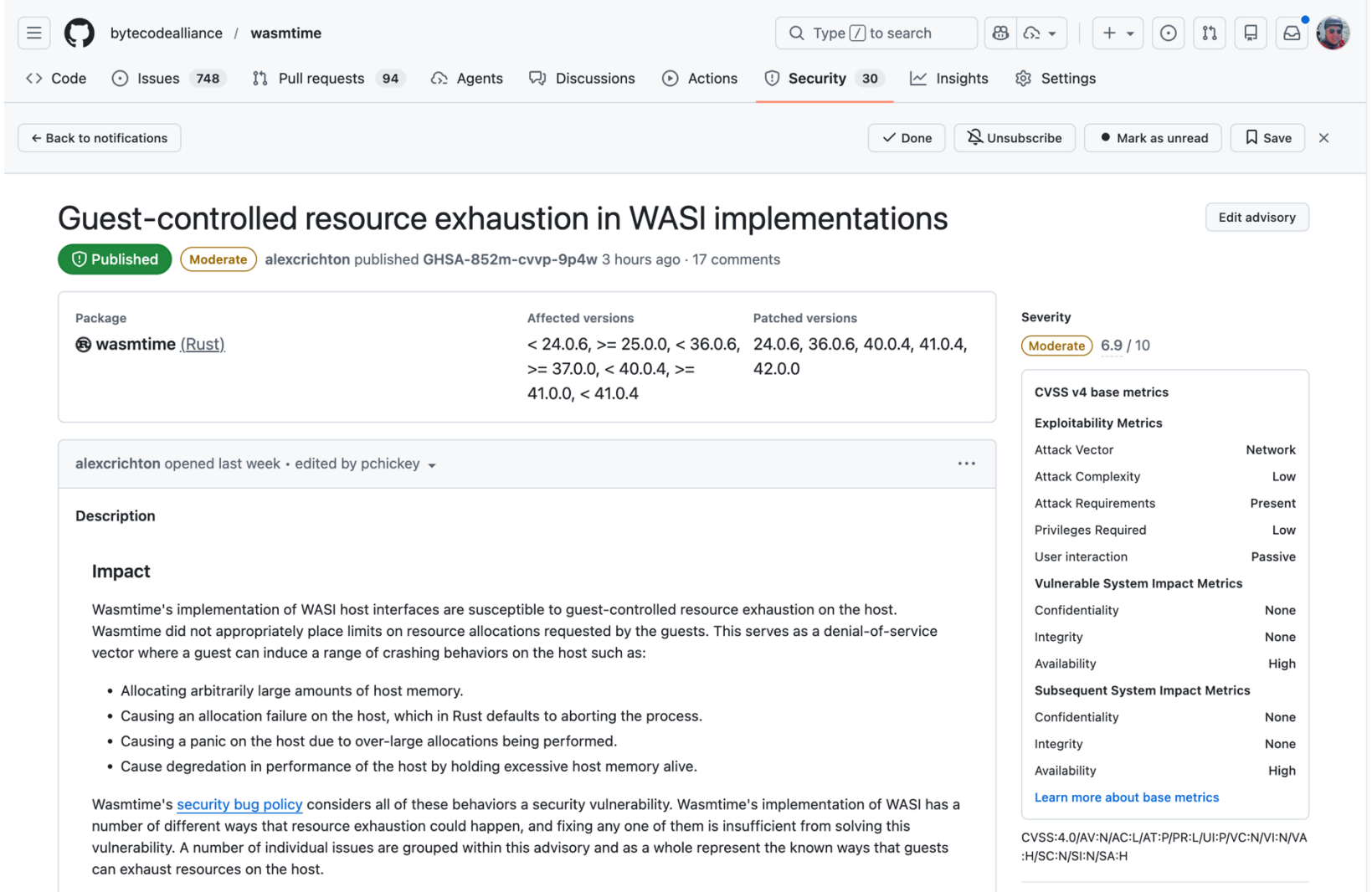
# Follow-on for Wasmtime's Feb 24, 2026 Security Advisory

Pat Hickey

F5

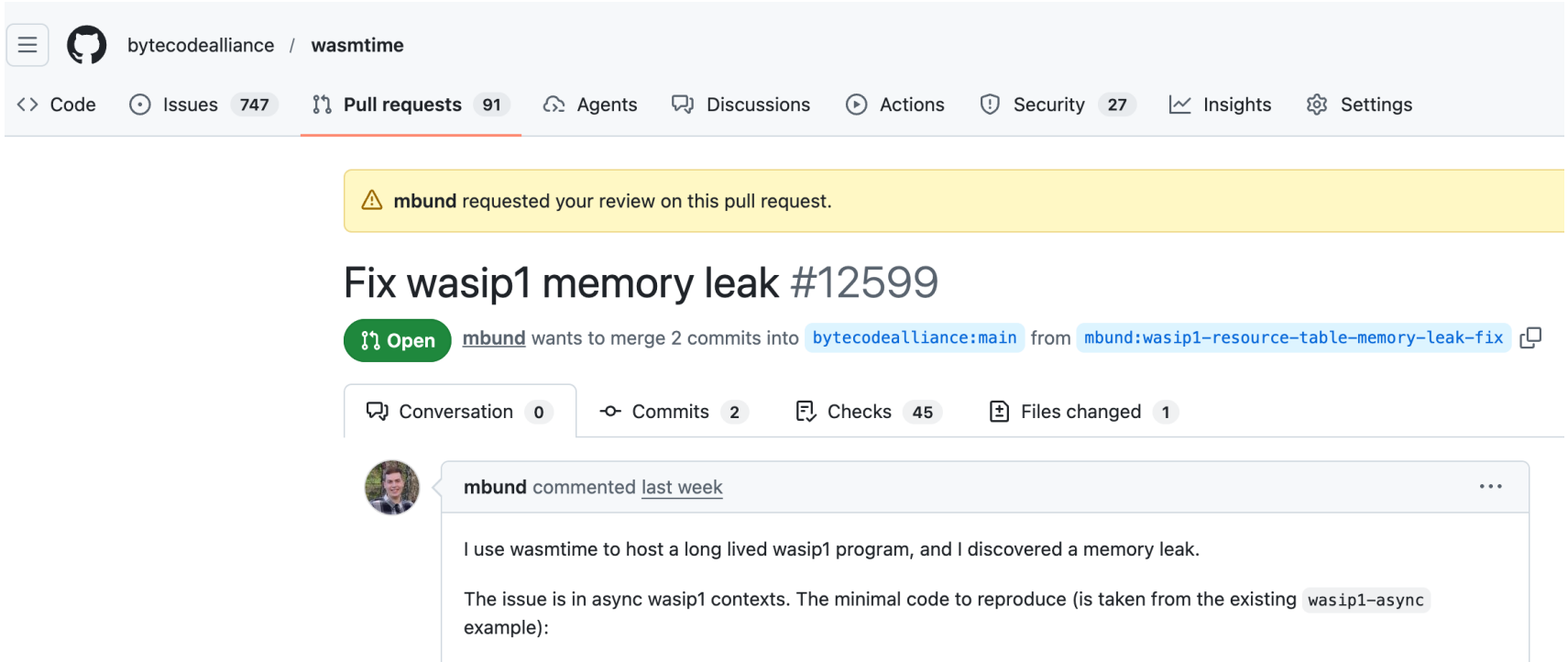
# The Advisory

https://github.com/bytecodealliance/wasmtime/security/advisories/GHSA-852m-cvvp-9p4w



# Credit

- New contributor @mbund reported memory leak in wasmtime-wasi's p1 impl. Thank you!!!
- Alex investigated and realized it was a much bigger issue
- Alex did most of the remediation, I helped



# Guest input can consume Host memory

- By creating resources – *new limit on how many, traps*
- By calling **any** import function accepting lists or strings with large lists and strings – *new limit on total size, traps*
- By calling **particular** import functions with arguments that cause allocations, some of which are long-lived
  - Stream or file read takes a size argument: allocates a BytesMut. Now caps at maximum read size, which is allowed by spec
  - Get random bytes takes a size argument: allocates a Vec<u8>. New limit, traps
  - HTTP Fields gives arbitrary Vec<u8> to a resource to own forever. New limit, traps
    - Bonus: when this exceeded 32k it would panic the host in http::HeaderMap. Fixed!

# Security mitigations, not long-term solutions

- We broke userland
  - There may be valid programs which now trap
- We did so unilaterally
  - Because it was a security issue, we couldn’t seek community input
- If the host changes a limit, the guest can’t discover it
  - Writing a guest which doesn’t trap today requires reading the wasmtime release notes.
    - wasi-libc and wstd could help, but do not (yet)
  - Writing a guest which will always respect host limits is impossible

If you maintain a host which isn't just wasmtime-{wasi + wasi-http}, *you probably have special cases you need to discover and fix*

And you might have to break userland too

# This needs to flow back up to the standards

- Amorphous and large design space. First, socialize some actual plans, then figure out the standards
- Does the CM spec take on resource and lifting fuel implementation limits?
- wasip1 and wasip2 spec – need to document these new limits, which requires on settling on them
- wasip3 spec: can we design our way out?
  - It's the 11th hour before release, but I don't think it should go out without thinking more about these issues
- Alex filed https://github.com/WebAssembly/WASI/issues/888, 889, 890