November 18, 2026 · Changes since the July update

LuaVirtualBox V2

Better Luau compatibility, an optional source optimizer, and a simpler Web IDE for protecting Lua and Roblox scripts.

It's been a little over four months since the July update. That round made parts of the generated output more compact and tightened runtime protection. Work picked up again in late October with Luau compatibility, then moved in November to execution speed, compiler correctness, and the new web editor.

Better compatibility with Lua and Luau

V2 accepts more typed Luau source, including scripts that use type annotations. The compiler also has fixes for cases where generated code could change the behavior of valid input.

- Function calls and returns preserve multiple values and varargs in more cases.
- Numeric loops, compound assignments, and concatenation preserve evaluation order and grouping.
- Table constructors, wider tables, and functions with many live values have additional regression coverage.

We compared protected output with the original scripts on Lua 5.1, Lua 5.4, and the Luau CLI. The Roblox regression suite covers GUI startup and callbacks with simulated services across Mini, Min, Med, and Max. Luau support is still evolving, so test your generated script in the Roblox Studio project where it will run.

Source optimization and execution speed

The new **Source code optimizer** is enabled by default in the Web IDE. It simplifies eligible code before protection and can be switched off from the same panel. V2 also reduces overhead in common VM call and loop paths and cuts repeated allocations during execution.

Max has a separate **Prefer speed over protection** option. It trades some protection for faster execution in eligible arithmetic-heavy loops and is off by default. On one local arithmetic fixture, we recorded the following median times across three paired builds.

Runtime	Default Max	Speed option on
Lua 5.1	5.05 s	2.88 s
Lua 5.4	3.87 s	2.20 s
Luau	2.44 s	1.51 s

Measured from process launch to exit on Debian under WSL. These are execution times for one workload. Build time, output size, and runtime vary by script and preset. Max can take longer to build and produce larger files than earlier versions.

Protection and build reliability

Since July, we've updated string and constant protection, increased variation across generated builds, and strengthened integrity checks in the protected runtime. We've also reduced build memory on larger inputs and added regression tests for both generated output and the protection presets.

One V2 Web IDE

The V2 Web IDE replaces the separate Legacy and Current editors. Existing editor links lead to the same page. Min is selected by default, and you can copy or download the generated script when the build finishes.

- **Min** produces the smallest output of the three web presets.
- **Med** balances protection and output size.
- **Max** applies the strongest protection and offers the optional speed setting.

If you build locally, the CLI also offers Mini and exposes the Max speed option as `--prefer-speed-over-protection`.

Open the Web IDE