

Forum:[Hack Forums](#)

Thread: [Monomorph - MD5-Monomorphic Shellcode Packer](#)

DI?E - February 6, 2023, 1:45 pm

What does it do?

It packs up to 4KB of compressed shellcode into an executable binary, near-instantly. The output file will always have the same MD5 hash: 3cebbe60d91ce760409bbe513593e401

Currently, only Linux x86-64 is supported. It would be trivial to port this technique to other platforms, although each version would end up with a different MD5. It would also be possible to use a multi-platform polyglot file like APE.

Why?

People have previously used single collisions to toggle a binary between "good" and "evil" modes. Monomorph takes this concept to the next level.

Some people still insist on using MD5 to reference file samples, for various reasons that don't make sense to me. If any of these people end up investigating code packed using Monomorph, they're going to get very confused.

How does it work?

For every bit we want to encode, a colliding MD5 block has been pre-calculated using FastColl. As summarised here, each collision gives us a pair of blocks that we can swap out without changing the overall MD5 hash. The loader checks which block was chosen at runtime, to decode the bit.

To encode 4KB of data, we need to generate $4 \times 1024 \times 8$ collisions (which takes a few hours), taking up 4MB of space in the final file.

To speed this up, I made some small tweaks to FastColl to make it even faster in practice, enabling it to be run in parallel. I'm sure there are smarter ways to parallelise it, but my naive approach is to start N instances simultaneously and wait for the first one to complete, then kill all the others.

Since I've already done the pre-computation, reconfiguring the payload can be done near-instantly. Swapping the state of the pre-computed blocks is done using a technique implemented by Ange Albertini.

Is it detectable?

Yes. It's not very stealthy at all, nor does it try to be. You can detect the collision blocks using detectcoll.

Check it out here: <https://github.com/DavidBuchanan314/monomorph>