

The payload is encrypted with AES CFB and will be decrypted and run via memfd_create by the stub. Key and IV are stored in the binary.

```
0021:7530 2F 69 6F 75 74 69 6C 2E 69 6E 69 74 00 6D 61 69 /ioutil.init.mai f0wL ~ -> Malware > ezuri_crypter > ezuri ./ezuri_crypter
0021:7540 6E 2E 72 75 6E 46 72 6F 6D 4D 65 6D 6F 72 79 00 n.runFromMemory. [?] Path of file to be encrypted: test
0021:7550 6D 61 69 6E 2E 61 65 73 44 65 63 00 6D 61 69 6E main.aesDec.main [?] Path of output (encrypted) file: packedTest
0021:7560 2E 6D 61 69 6E 00 6D 61 69 6E 2E 69 6E 69 74 00 .main.main.init. [?] Name of the target process: TEST
0021:7570 72 40 61 39 51 75 41 4B 77 73 35 79 55 40 57 6D r@a9QuAKws5yU@Wm [?] Encryption key (32 bits - random if empty):
0021:7580 65 44 6D 53 31 7A 6C 67 38 66 4E 7A 61 72 56 45 eDmS1zlg8fNzarVE [?] Encryption IV (16 bits - random if empty):
0021:7590 56 75 50 46 4B 43 64 4C 38 68 73 76 64 4A 25 49 VuPFKcdL8hsvdJ%I [!] Random encryption key (used in stub): r@a9QuAKws5yU@WmeDmS1zlg8fNzarVE
0021:75A0 DA 38 38 D7 03 49 0C A9 DA D6 D7 18 CA 8E 78 3F U8;x.I.eU0x.E.x? [!] Random encryption IV (used in stub): VuPFKcdL8hsvdJ%I
0021:75B0 F4 F8 A2 16 7D B2 37 0B A0 6A E8 F4 12 20 7D 6B 00c.}~7. jêð. }k [!] Generating stub...
0021:75C0 A7 2E 58 70 3B 70 C7 92 C1 E9 0A F4 B2 B7 47 9B $.Xp;pÇ.Áé.ô².G. [!] Creating final executable...
0021:75D0 B1 E5 4E 27 8F F3 F9 B1 CE A1 E9 0D 62 8D 0C A4 ±âN'.òù±î;é.b..# [!] All done!
0021:75E0 37 07 54 98 09 A6 FD 17 F8 C5 0A BD 3C 4B B6 34 7 T. jv.âÀ.¼<K#4
```

Testing the script

1. Build the test payload `gcc test.c -o test`
2. Build and run `guitmz/ezuri`
3. To unpack it again: `go run ezuri_unpack.go packed.bin`

I also tested it with the packed Linux.Cephei sample mentioned in the report. [Link to Virustotal](#)