

Code Therapy, use of computer viruses in executable editing.

---

Dr Bheemaiah, Anil Kumar, A.B Seattle W.A 98125  
miyawaki@yopmail.com

---

#### Abstract:

Binary editing and profiling tools abound, production code in serverless and server based deployment and container based systems, viral vectors for binary editing for run time optimization and reconfigurability for optimization and tailoring of legacy applications is presented in this publication, using a variety of non - malware virus generators of hex payloads, precompiled as hex payloads. Examples include Java embedded software in coffee machines like the Mugsy and the use of the open source PASTA software , by the Toyota Corporation for reconfiguring the ECU towards true flex vehicles.

Keywords: PASTA, Viral Vectors, Binary Editing, snippet, hex code payload,

#### What:

Code Therapy is defined analogously to Gene Therapy in personalized medicine, with the effect of modifying legacy binaries with code tailored as code therapy for a therapeutic effect. In this paper we use virus vectors, which are non pathogenic, with a code payload for targeted code corrections for therapeutic effects.

#### How:

We experiment with several proven viruses, modified for use as a viral vector for code therapy and executable code editing with threads of binary called snippets. We use java byte code snippets as an example and the design of bytecode viruses.

Another example is the conversion of a toyota corolla ECU code with a viral vector for the use of the fuel sensor, to detect the composition of the fuel and change the oxygen and fuel ratio accordingly, using defined formulas.

Why: The Positifaction principle of H.H Dalai Lama, a major proponent of this approach is to convert anything pathogenic to something positive, as defined in Godel's proof of God in a multiverse. Thus positified, there is value and usefulness from these technologies. Computer viruses are thus positified for executable editing and for code therapy.

## Introduction.

Java byte code viruses like beehive and other viruses, can be easily generated from

virus generators. Virus Generators are code applications, which are low code approaches to writing vectors for binary payloads for executable editing.

<sup>1</sup> describes the tools and the methodology for the editing of binary code in ECU units of automobiles. Given hex code snippets, byte compiled for a specific instruction set, in an ECU, it is possible to use generic viral generators for the editing of executables with hex code snippets.

CarMa<sup>2,3</sup> is an optimization tool for automobiles, which can be used to write an app for FFV conversion, a future exercise.

As an example we consider the Toyota Corolla, since all cars since 1994 have a OBD or onboard diagnostics port, we can use the OBD-II port with a toyota code reader unit for manipulation of the code and editing of the code. We next prove the efficiency of using a viral vector in the manipulation of code with a library of snippets, <sup>4</sup>.

To convert a vehicle to a FFV or flexible fuel vehicle, we need to add hex code snippets, to read fuel composition and add formulas to modify the fuel air ratio and the compression ratio.

PASTA is an open-source software for the CAN bus and OBD-II based editing of the ECU software of toyota automobiles.<sup>5</sup>

## Problem Definition.

Positification is defined as the conversion in a multiverse, by Godel's Ontological definition of positives of the conversion in multiple values to positive values of any process or technology. In this paper a computer virus is posited to code or

executable editing and automation of editing.

P1: Proof that virus based executable is a better approach than a remapping or flashing of an entire image, to embedded memory in legacy applications.

P2: An example of a java virus generator, that can modify a byte code file at an arbitrary location with a hex code snippet, for executable editing.

## Background.

Traditionally OBD scanners are used by experts and amateurs for performance tuning, while every scanner can modify the ECU code by reimaging, real time executable editing is possible by the use of a viral vector, rather than reimaging the entire image.

Low code approaches to the design of viruses, lead to the use of virus generators with customizable payloads and targets, for use in executable editing. Using the deprecated JOS or java operating system, with the use of Java ME on coffee makers, for example, we illustrate a Coffee Virus, a vector on a coffee machine with Java ME.

## Need For a Viral Vector.

Analogous to the use of viral vectors in gene therapy, the need for self replication and global infection of all files or gene targets in the case of gene therapy or code therapy, require a viral vector to introduce target hex code as a junk code payload. Thus code is introduced in every infected file, allowing for the splicing of executable snippets at precise positions in executables,

creating in the process of positification a powerful tool for the editing of executables.

## Virus Generators.

Some of the prominent virus generators , code generators and code evolution machinery. Several virus generators like next generation virus generators and other generators exist for low code creation of viruses, all of which support the ability to add junk code to applications and remain in a stealth mode.

## Pasta, for True Flex Fuel.

Using the pasta application, defined similarly, the [h,a] for maps for sensor readings and a lookup table for fuel air ratios for given VOCs.

We use the overflow sensor data for VOC pressure profiles, using procedural AI for determination of the fuel combination from a 1D stream of overflow fuel pressure readings, using a perceptron based decoder for fuel composition.

Given the composition, [c], a lookup table at an address ca, the fuel air ratio is injected at address cf, with the payload hf at [hf, af].

## Discussion and Future Work.

The open source automotive security code framework called PASTA, by Toyota, enables the creation of truly flexible fuel hybrid vehicles, using a perceptron based

prediction of fuel air ratios and fuel constituents using the overflow VOC sensor, for evaporative emissions. Future work constitutes proprietary designs for perceptron based 1D waveform transformation functions for determination of Fuel composition, similar to data and branch prediction architectures in Ryden architectures from AMD.

The efficiency of file dependent payload with viral binary editing is proven in this publication, with the example of editing ECU code towards flexible fuel air ratios. We thue prove the use of viral generators and use of viruses in binary editing with known hex payloads.

## References.

1. freeCodeCamp.org. How to hack a car — a quick crash-course. *freeCodeCamp.org*  
<https://www.freecodecamp.org/news/hacking-cars-a-guide-tutorial-on-how-to-hack-a-car-5eafcbbb7ec/> (2017).
2. [No title].  
<https://dl.acm.org/doi/abs/10.1145/2070942.2070957?download=true>.
3. [No title].  
<https://dl.acm.org/doi/abs/10.1145/2070942>

.2070957?download=true.

4. VehicleHistory.com. Vehicle History.
5. pasta-auto. pasta-auto/PASTA1.0.  
<https://github.com/pasta-auto/PASTA1.0>.