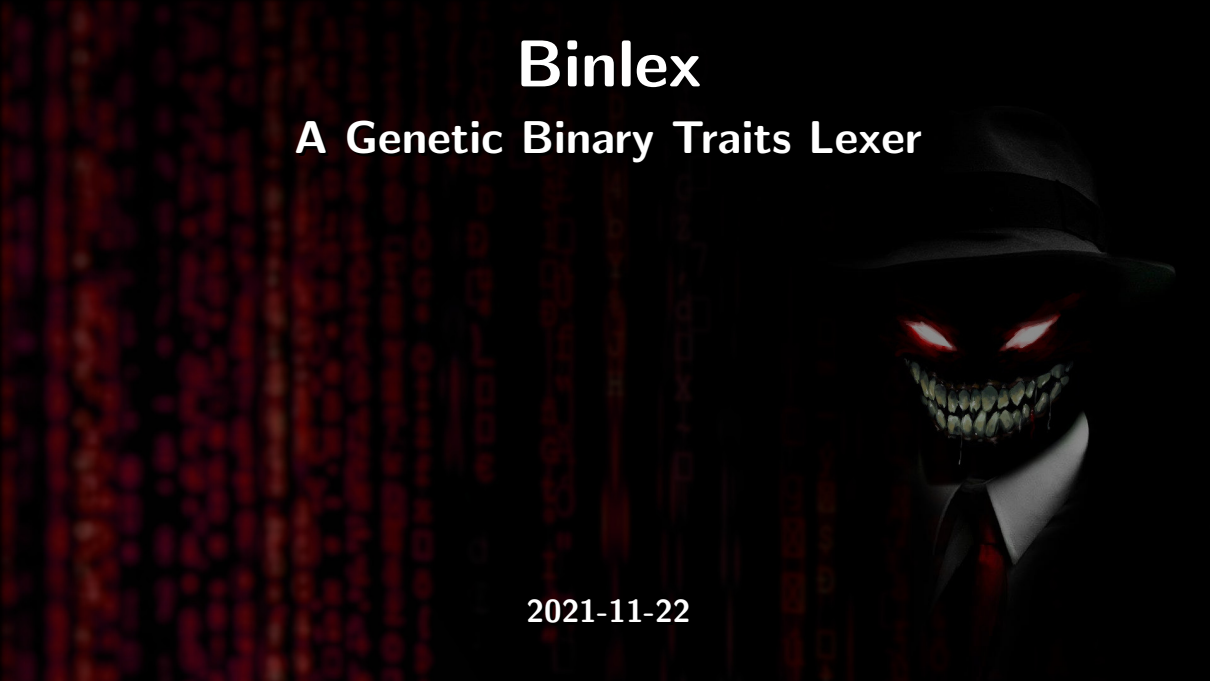
Binlex

A Genetic Binary Traits Lexer



2021-11-22

Binlex

whois

terminal

```
c3rb3ru5@d3d53c $ whois c3rb3ru5d3d53c
Job: Director of Malware Lab
Experience: 6+ years
Skills: Reversing Engineering, Malware Analysis, C++, C, Assembly, Python
Hobbies: Guitar, Machine Learning
Twitter: https://twitter.com/c3rb3ru5d3d53c
GitHub: https://github.com/c3rb3ru5d3d53c
```

Binlex

Problems Facing Malware Analysts and Devlopers

- Hunting Code-Reuse
- Writing YARA Signatures based on Code
- Binary Genetic Trait Lexer
- Machine Learning Feature Collection

Binlex

What is a Lexer?

lexer.log

In computer science, lexical analysis, lexing or tokenization is the process of converting a sequence of characters into a sequence of tokens.

Binlex

Installing

terminal

```
c3rb3ru5@d3d53c $ git clone https://github.com/c3rb3ru5d3d53c/binlex.git
c3rb3ru5@d3d53c $ cd binlex/
c3rb3ru5@d3d53c $ sudo apt install -y git libcapstone-dev cmake make
c3rb3ru5@d3d53c $ sudo apt install -y parallel
c3rb3ru5@d3d53c $ make threads=4
c3rb3ru5@d3d53c $ sudo make install
```

Binlex

Getting Help

terminal

```
c3rb3ru5@d3d53c $ binlex –help
binlex v1.0.0 - A Binary Genetic Traits Lexer
-i –input input file (required)
-m –mode set mode (required)
-lm –list-modes list modes
-h –help display help
-o –output output file
-v –version display version
Author: @c3rb3ru5d3d53c
```

Binlex

Supported Modes

terminal

```
c3rb3ru5@d3d53c $ binlex -lm
elf:x86
elf:x86_64
pe:x86
pe:x86_64
raw:x86
raw:x86_64
raw:cil
```

Binlex

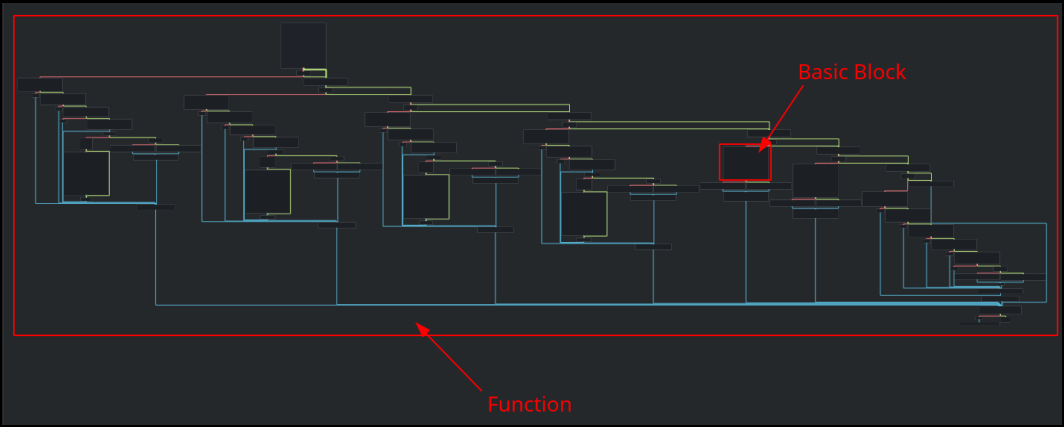
Genetic Traits - What are they?

genetic_traits.log

Genetic binary traits consist of byte sequences. These byte sequences consist of hexadecimal strings with memory operands and others subject to change wild carded with question marks.

Binlex

What Makes a Trait?



Binlex

What Makes a Trait?

- Basic Blocks
  - jmp
  - jne
  - jz
- Functions
  - ret

Binlex

Extracting Traits

terminal

```
c3rb3ru5@d3d53c $ binlex -m pe:x86 -i tests/pe/pe.x86 | head -2
50 b9 60 44 41 ?? c7 00 40 05 41 ?? e8 78 46 ?? ?? c3
c7 41 ?? e0 e0 e0 ?? 89 51 ?? 39 50 ?? 56 8b 74 24 ?? 74 12
```

Binlex

Difference from Other Tools

- Current Tools
  - VirusTotal Diff
  - Intezer
  - Yargen
  - yara-signator
- Differences
  - Wildcarding
  - Framework
  - Write your own Logic
  - Not a Blackbox, It's OPEN!

Binlex

Build your Own!

genetic_programming.log

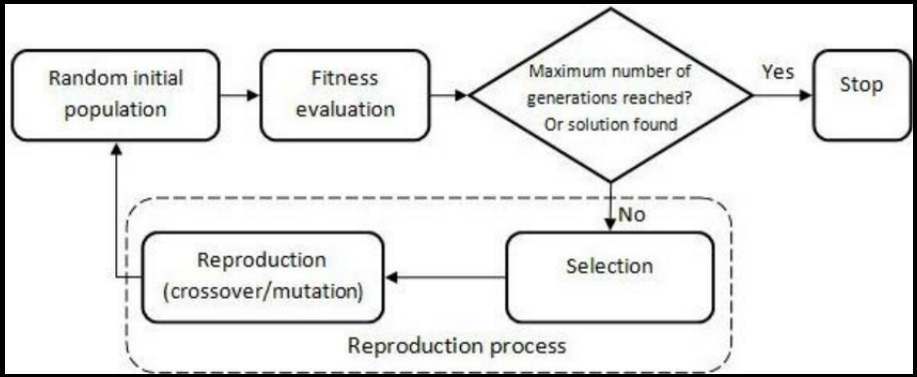
In artificial intelligence, genetic programming (GP) is a technique of evolving programs, starting from a population of unfit (usually random) programs, fit for a particular task by applying operations analogous to natural genetic processes to the population of programs.

fitness.log

A fitness function is a particular type of objective function that is used to summarise, as a single figure of merit, how close a given design solution is to achieving the set aims. Fitness functions are used in genetic programming and genetic algorithms to guide simulations towards optimal design solutions.

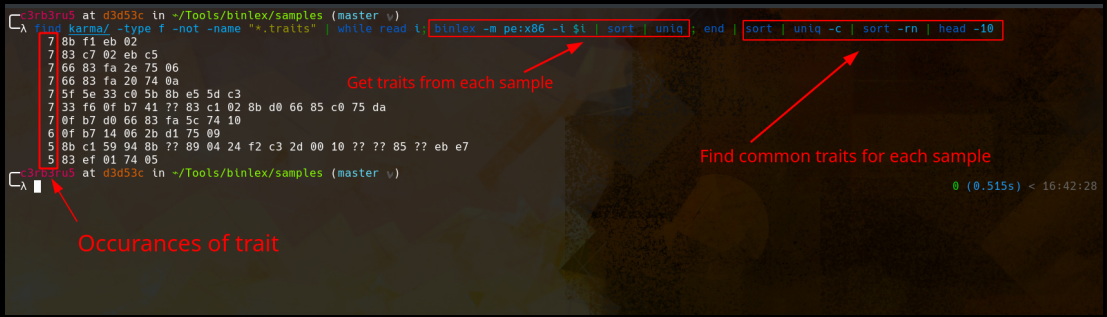
Binlex

Building your Detection Logic - Fitness



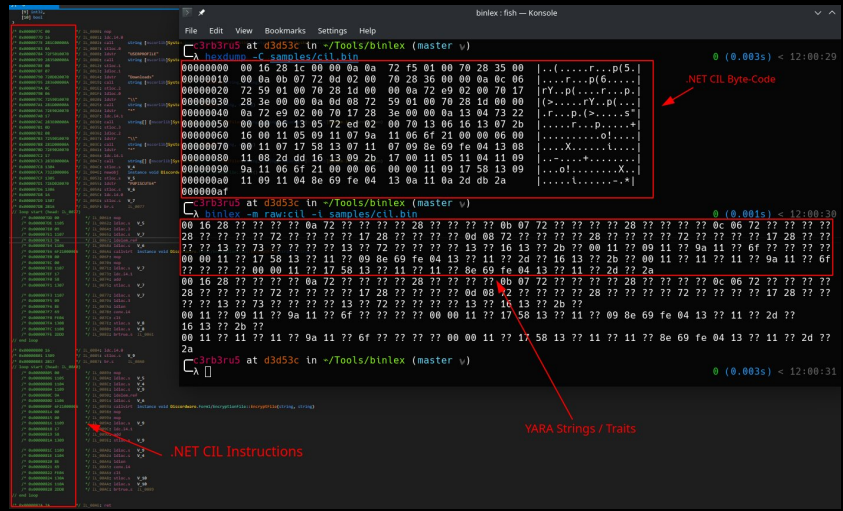
Binlex

Building your Detection Logic - With Bash



Binlex

Building your Detection Logic - CIL/.NET



Binlex

Building your Detection Logic - C++ API

```
if (strcmp(args.options.mode, (char *)"raw:cil") == 0 &&
    args.options.io_type == ARGS_IO_TYPE_FILE){
    Raw raw_cil;
    if (raw_cil.ReadFile(args.options.input, 0) == false){
        return 1;
    }
    CILDecompiler cil_decompiler;
    if (cil_decompiler.Setup(CIL_DECOMPILER_TYPE_BLCKS) == false){
        return 1;
    }
    if (cil_decompiler.Decompile(raw_cil.sections[0].data, raw_cil.sections[0].size, 0) == false){
        return 1;
    }
    if (cil_decompiler.Setup(CIL_DECOMPILER_TYPE_FUNCS) == false){
        return 1;
    }
    if (cil_decompiler.Decompile(raw_cil.sections[0].data, raw_cil.sections[0].size, 0) == false){
        return 1;
    }
    if (args.options.output == NULL){
        cil_decompiler.PrintTraits();
    } else {
        cil_decompiler.WriteTraits(args.options.output);
    }
    return 0;
}
```

Binlex

Future Work

- Cutter Plugin
- Ghidra Plugin
- IDA Plugin
- C++ API Documentation
- Python Bindings
- Machine Learning
- Default Fitness Function (built-in)
- Looking for Developers

Binlex

References

- https://en.wikipedia.org/wiki/Fitness_function
- https://en.wikipedia.org/wiki/Genetic_programming
- https://www.capstone-engine.org/

Binlex

Demo Time!