Обфускация потока управления программ с использованием LLVM в целях защиты интеллектуальной собственности.

# Введение

**Обфускация** — приведение исходного текста или исполняемого кода программы к виду, сохраняющему ее функциональность, но затрудняющему анализ, понимание алгоритмов работы и модификацию при декомпиляции.

* Затруднение декомпиляции проприетарных программ с целью предотвращения обратной разработки или обхода DRM и систем проверки лицензий.
* Нарушение авторских прав программистов и скрытие авторства.
* Генерация различных исполняемых модулей одной программы

# Существующие подходы к обфускации

# CFG flattering

Преобразование графа исполнения из направленной структуры в зацикленную.

# Opaque predicates

Рандомизация и “раздутие” графа выполнения программы. Добавленные узлы графа никогда не получают управление, тем самым не влияя на поведение программы, но определить такие узлы с помощью статического анализа сложно.

# Junk code

“Замусоривание” кода инструкциями не влияющими на логику программы с целью увелечения объема кода.

# Constant unfolding

Вычисление известных констант во время исполнения программы с целью затруднения статического анализа.

# Pattern-based obfuscation

Замена инструкций на их аналоги/последовательность инструкций выполняющих аналогичные действия.

# Uncommon instructions

Использование специализированных(и редко используемых) инструкций для замены инструкций защищаемого кода.

# Virtual Machine

Преобразование защищаемого кода в код абстрактного процессора и выполнение данного кода на эмуляторе такого процессора.

# Похожие работы

1) Obfuscator-llvm (<https://github.com/obfuscator-llvm/obfuscator/wiki>)

2) Kryptonite ([http://download.tuxfamily.org/overclokblog/Obfuscation%20of%20steel%3a%20meet%20my%20Kryptonite/0vercl0k\_Obfuscation\_of\_steel\_meet\_kryptonite.pdf](http://download.tuxfamily.org/overclokblog/Obfuscation of steel%3A meet my Kryptonite/0vercl0k_Obfuscation_of_steel_meet_kryptonite.pdf))

3) CONFUSE (<http://www.cs.columbia.edu/~aho/cs4115_Spring-2013/lectures/13-05-16_Team11_Confuse_Paper.pdf>)

4) <http://www.ruscrypto.ru/resource/summary/rc2014/05_kurmangaleev.pdf>

5) [Deobfuscation: recovering an OLLVM-protected program](http://blog.quarkslab.com/deobfuscation-recovering-an-ollvm-protected-program.html)

(http://blog.quarkslab.com/deobfuscation-recovering-an-ollvm-protected-program.html)

# Описание работы

<https://bitbucket.org/Daeh0f/capybara>

На данный момент реализовано:

- рандомизация графа исполнения

- замусоревание бессмысленными инструкциями

# Литература

1) Practical Reverse Engineering: x86, x64, ARM, Windows Kernel, Reversing Tools, and Obfuscation

Bruce Dang, Alexandre Gazet, Elias Bachaalany, Sebastien Josse

2) Compilers: Principles, Techniques, and Tools. Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman