7. *Девянин П. Н.* Подходы к моделированию управления доступом в СУБД PostgreSQL в рамках МРОСЛ ДП-модели // Прикладная дискретная математика. Приложение. 2018. № 11. С. 95–98.

УДК 004.056.53, 004.032.26 DOI 10.17223/2226308X/12/46

ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ КАК МЕХАНИЗМ ОБФУСКАЦИИ ВЫЧИСЛЕНИЙ

В. Л. Елисеев

Рассмотрены возможности использования искусственных нейронных сетей в качестве механизма строгой обфускации вычислительного алгоритма. Обсуждается задача обфускации, основные положения и современные подходы к её реализации. Вводится понятие нейросетевого обфускатора и доказываются его свойства. Приводятся достоинства и недостатки предложенного подхода.

Ключевые слова: *искусственная нейронная сеть, обфускация.*

Введение

Обфускация (сокрытие) представляет собой преобразование алгоритма к форме, затрудняющей изучение его свойств. Актуальность рассмотрения подобных механизмов основана на том факте, что программное обеспечение в подавляющем большинстве случаев представляет собой объект интеллектуальной собственности, но при этом используется в вычислительной среде, не контролируемой владельцем прав на программу. Наиболее часто обфускации подвергаются части программы, реализующие механизмы защиты от копирования и иного использования в обход лицензии. Примитивные механизмы обфускации применяются также для защиты от изучения человеком исходного кода библиотек, поставляемых на интерпретируемых языках (PHP, Python, JavaScript, Java).

Значительное число работ, посвящённых обфускации исходного кода программ (например, [1]), предлагает разнообразные эвристические методы, автоматизирующие эквивалентную трансформацию программы с целью запутывания оригинальной структуры алгоритма путём изменения представления используемых данных и констант. Такое преобразование даёт визуально достаточно впечатляющие результаты, однако для каждого из эвристических обфусцирующих алгоритмов есть деобфусцирующий, восстанавливающий исходную структуру. По этой причине эвристическая обфускация не может считаться надёжным подходом для защиты алгоритмов, содержащих конфиденциальную информацию, в том числе криптографические ключи и «ноу хау» (скоринговые алгоритмы страховых компаний и банков). Ряд задач, решение которых могло бы упроститься при создании стойких обфускаторов, приведено в [2].

1. Обзор

Начиная с момента появления до работы [3], все методы обфускации представляли собой эвристики, которые было трудно оценивать и сравнивать друг с другом в части стойкости к действиям злоумышленника, пытающегося восстановить исходный алгоритм. Общеизвестно, что любые эвристические подходы всегда имеют ограничения по области применения. В случае с обфускацией для каждого эвристического метода можно построить алгоритм деобфускации, восстанавливающий исходный алгоритм.

В работе [3] впервые введено строгое понятие стойкой обфускации. Неформально обфускатор $\mathcal{O}$ — это транслятор, принимающий на вход программу P и производящий программу $\mathcal{O}(P)$, такую, что выполняются два условия:

— функциональность $\mathcal{O}(P)$ эквивалентна функциональности P;
— всё, что может быть эффективно вычислено из $\mathcal{O}(P)$, может быть эффективно вычислено случайным оракулом при доступе к P.

Второе свойство получило название виртуального чёрного ящика, поскольку случайный оракул — это идеализированная хэш-функция со случайным детерминированным выходом. По сути это свойство означает, что нет более эффективного способа для изучения обфусцированного алгоритма, чем его многократный запуск. Иногда добавляется также свойство эффективности обфускации [2], которое обуславливает ограничение на размер и вычислительную сложность $\mathcal{O}(P)$ по сравнению с P.

Вместе с тем сразу было показано, что существует класс программ (схем), для которого свойство виртуального черного ящика недостижимо [3]. Для снижения планки требований в той же работе введено понятие обфускации неразличимости (*indistinguishability obfuscation*). В соответствии с этим определением две различные, но функционально эквивалентные программы P_1 и P_2 ($\forall x\,(P_1(x) = P_2(x))$), подвергнутые обфускации, неразличимы за полиномиальное время. Это означает, что невозможно эффективно определить, является ли $\mathcal{O}(P_1)$ обфускацией P_1 или P_2.

Для обфускации неразличимости в [4] доказано, что это лучший из возможных обфускаторов, то есть такой, который сообщает об исходной программе не больше, чем любая другая программа с функциональностью, эквивалентной исходной.

В 2013 г. представлен первый и до настоящего времени единственный класс методов обфускации, доказуемо реализующих свойство неразличимости, — это так называемое градуированное кодирование (graded encoding) [5]. Известно, что реализация этого подхода для прикладных применений существенно неэффективна с вычислительной точки зрения [6], что делает как процедуру обфускации, так и сам полученный обфускатор малопригодными в реальных задачах.

В одной из работ, посвященных эмпирической обфускации [7], для сокрытия логики переходов в условных операторах предложено использовать искусственную нейронную сеть (далее нейросеть), функционально эквивалентную вычисляемому предикату на множестве допустимых значений аргументов. Отмечено, что после обучения, то есть собственно процедуры обфускации, нейросеть несущественно снижает производительность программы при скромных расходах памяти на хранение матрицы весовых коэффициентов. В то же время применение нейросетей для задачи обфускации подвергнуто критике в [8] главным образом за непредказуемость результатов за пределами обучающей выборки.

Известны многочисленные варианты формальных конструкций, причисляемых к нейросетям, достаточно полный перечень и описание которых приведены в [9]. Далее будем говорить только о нейросетях типа *многослойный персептрон* (*multilayer perceptron — MLP*). Многослойный персептрон вычисляет выходной вектор по входному $x^{(N)} = f(x^{(0)})$, при этом реализует алгебраическое преобразование вида

$$x_i^{(k)} = s\left(\sum_{j=1}^{m_{k-1}} w_{ij}^{(k)} x_j^{(k-1)}\right),\ k = 1, \ldots, N,\ i = 1, \ldots, m_k,$$

где N — количество слоёв нейросети; $x^{(k)} \in \mathbb{R}^{m_k}$ — вектор выходов слоя k; $x^{(0)} \in \mathbb{R}^{m_0}$ — вектор входов нейросети; m_k — количество нейронов слоя k; m_0 — количество вхо-

дов первого слоя нейросети; $w_{ij}^{(k)} \in \mathbb{R}$ — весовой коэффициент j-го входа i-го нейрона слоя k; $s(.)$ — дифференцируемая функция, называемая также функцией активации. Как правило, эта функция имеет вид сигмоиды, например логистической: $s(t) = 1/(1 + e^{-t})$.

В 1987 г. Р. Хехт-Нильсен на основе работ А. Н. Колмогорова доказал представимость непрерывной функции многих переменных с помощью двухслойной нейросети [10]. Этот результат неконструктивен, однако даёт основание считать нейросети универсальными аппроксиматорами одномерных и многомерных непрерывных функций многих переменных. Разработаны не имеющие гарантий успешности, но показавшие высокую результативность методы обучения нейросетей. Под обучением нейросети понимают процедуру задания или, что обычно, итеративного изменения весовых коэффициентов $w_{ij}^{(k)}$ в процессе решения оптимизационной задачи для получения заданной точности аппроксимации таблично заданной неизвестной непрерывной функции. Для обучения нейросетей используется большой спектр методов оптимизации, использующих градиентные, стохастические и другие подходы, а также их комбинации [9].

Следует отметить, что выбор структуры нейросети — числа слоёв и количества нейронов в них — до сих пор является плохо изученной проблемой и на практике обычно используются различные эвристики. Значимым результатом в этой области является получение оценки VC-измерения нейросети с сигмоидальными функциями активации — $\mathrm{O}(W^2)$, где $W = \sum\limits_{k=1}^{N-1} m_{k-1}m_k$ — количество настраиваемых параметров $w_{ij}^{(k)}$ нейросети [9].

2. Предложение

Рассмотрим задачу обфускации программы P с помощью нейросети. Определим P в общем виде как детерминированную функцию вектора переменных x из множества допустимых векторов $X \subseteq \mathbb{R}^n$, вычисляющую значение y из множества $Y \subseteq \mathbb{R}^m$:

$$y = P(x),\ x \in X,\ y \in Y.$$

Определение 1. Назовём *нейросетевым обфускатором функции* P нейросеть $\mathcal{N}$ с количеством входов n и количеством выходов m, достаточно точно аппроксимирующую эту функцию.

В дальнейшем рассмотрении ограничимся программами, оперирующими булевыми векторами конечного размера: $X \subseteq \mathbb{B}^n$, $Y \subseteq \mathbb{B}^m$, где $\mathbb{B} = \{0, 1\}$.

Определение 2. *Нейросетевым обфускатором векторной булевой функции* P является нейросеть $\mathcal{N}$ с количеством входов n и количеством выходов m, такая, что

$$\forall x \in X \left((y = P(x)\ \&\ \hat{y} = \mathcal{N}(x)) \Rightarrow |y_i - \hat{y}_i| < 0{,}5,\ i = 1, \ldots, m\right).$$

Отметим, что при обучении нейросетей аппроксимации неизвестной функции, заданной множеством точек ограниченного размера $\{(x_i, y_i) : i = 1, \ldots, N\}$, возникают трудности, связанные с достижением точности аппроксимации, а также с контролем обобщающей способности нейросети (проблема переобучения нейросети [9]). Для их преодоления имеющееся в наличии множество точек разделяют на обучающее, тестовое и контрольное, что является эвристикой, не дающей гарантий успешного результата.

В рассматриваемом здесь случае нейросеть обучается аппроксимировать заведомо известную функцию P. В этом случае не возникает проблемы переобучения и всё множество данных $\{(x_i, y_i) : i = 1, \ldots, N\}$ должно использоваться как обучающее.

Теорема 1. Нейросетевой обфускатор булевой функции P обладает свойством функциональности.

Доказательство. Построим какую-либо непрерывную функцию $\tilde{P}$, проходящую через все значения функции P на области её определения. По теореме Хехт-Нильсена [11] для функции $\tilde{P}$ можно построить двухслойную нейросеть, аппроксимирующую эту функцию с любой наперёд заданной точностью. Эта нейросеть в точках $x \in X$ будет также аппроксимировать P. ■

Теорема 2. Нейросетевой обфускатор булевой функции P обладает свойством неразличимости.

Доказательство. Возьмём две различные программы P_1 и P_2, реализующие функцию P. Для них можно построить нейросетевые обфускаторы $\mathcal{N}_1$ и $\mathcal{N}_2$ соответственно. Поскольку для построения каждого из обфускаторов используется один и тот же набор данных, обусловленный значениями функции P, то невозможно определить, какая из нейросетей построена как обфускатор конкретной программы P_1 или P_2. ■

Приведённые выкладки могут быть обобщены на векторные функции с элементами, принадлежащими конечному множеству значений. Например, в качестве элементов векторов x и y можно использовать подмножество целых чисел или элементы конечного поля.

3. Обсуждение

Полученный результат позволяет утверждать, что нейросети обеспечивают обфускацию неразличимости для достаточно широкого класса алгоритмов. В то же время известный опыт применения нейросетей позволяет сформулировать следующие особенности предложенного метода обфускации:

1) объём выборки для обучения нейросети экспоненциально растёт с ростом размерности векторов x и y;
2) пока не существует однозначно эффективного метода для выбора структуры нейросети под конкретную задачу;
3) вычислительные ресурсы для обучения нейросети как минимум пропорциональны произведению размера выборки N на количество обучаемых параметров W;
4) вычислительные ресурсы для работы нейросетевого обфускатора пропорциональны количеству обучаемых параметров W.

Оценка вычислительной сложности для обучения нейросети, а также неопределённость с выбором её структуры однозначно являются существенными недостатками предложенного подхода. В то же время следует отметить, что работа нейросетевого обфускатора требует существенно меньших ресурсов. Кроме того, в настоящее время существуют аппаратные устройства для обучения и работы нейросетей, обеспечивающие ускорение до трёх десятичных порядков относительно универсальных процессоров.

Представляется важным исследовать вопросы, связанные с проектированием структуры нейросетевого обфускатора и получением уточнённых оценок вычислительной сложности его обучения. Тем не менее, учитывая большой накопленный опыт применения нейросетей, есть основания считать предложенный метод обфускации прак-

тически значимым и имеющим потенциал по применению в актуальных задачах, требующих надёжного сокрытия алгоритмов.

Выводы

Предложен новый механизм обфускации на основе искусственных нейронных сетей и доказано его соответствие требованиям функционального обфускатора неразличимости. Отмечены его основные свойства, перспективы дальнейших исследований и практического применения.

ЛИТЕРАТУРА

1. *Venkatesh S. and Ertaul L.* Novel obfuscation algorithms for software security // Proc. Intern. Conf. SERP'05. 2005. V. 1. P. 209–215.
2. *Варновский Н. П., Захаров В. А., Кузюрин Н. Н.* Математические проблемы обфускации // Математика и безопасность информационных технологий. Материалы конф. в МГУ 28–29 октября 2004 г. М.: МЦНМО, 2005. С. 65–91.
3. *Barak B., Goldreich O., Impagliazzo R., et al.* On the (im)possibility of obfuscating programs // Crypto'01. LNCS. 2001. V. 2139. P. 1–18.
4. *Goldwasser S. and Guy N. R.* On best-possible obfuscation // J. Cryptology. 2007. No. 27. P. 480–505.
5. *Garg S., Gentry C., Halevi S., et al.* Candidate indistinguishability obfuscation and functional encryption for all circuits // Proc. 54th IEEE Ann. Symp. FOCS'13. October 26–29, 2013. P. 40–49.
6. *Albrecht M. R., Cocis C., Laguillaumie F. and Langlois A.* Implementing candidate graded encoding schemes from ideal lattices // ASIACRYPT 2015. LNCS. 2015. V. 9453. P. 752–775.
7. *Ma H., Ma X., Liu W., et al.* Control flow obfuscation using neural network to fight concolic testing // 10th Intern. ICST Conf., SecureComm 2014, Beijing, China, September 24–26, 2014. Part I. P. 287–304.
8. *Yan Wang* Obfuscation with Turing Machine. A Thesis in Information Sciences and Technology. Pennsylvania State University, 2017. 42 p.
9. *Хайкин С.* Нейронные сети: полный курс. 2-е изд. М.: Вильямс, 2008.
10. *Алексеев Д. В.* Приближение функций нескольких переменных нейронными сетями // Фундаментальная и прикладная математика. 2009. Т. 15б. № 3. С. 9–21.
11. *Hecht-Nielsen R.* Kolmogorov's mapping neural network existence theorem // IEEE First Ann. Int. Conf. Neural Networks. San Diego, 1987. V. 3. P. 11–13.

УДК 519.21 DOI 10.17223/2226308X/12/47

ОЦЕНКА ВЕРОЯТНОСТИ УСПЕШНОЙ АТАКИ НАРУШИТЕЛЯ В БЛОКЧЕЙН-СЕТИ

И. В. Семибратов, В. М. Фомичев

Рассмотрена вероятностная модель, определяющая начала активных периодов функционирования злоумышленника и майнера как случайные величины, распределённые по биномиальному закону. Получены оценки вероятностей успешной атаки злоумышленника (создания ложного блока данных) при различных исходных условиях. Результаты вычислений подтвердили естественные предположения, что вероятность успешной атаки злоумышленника убывает как с ростом положительной разности длительностей сеансов майнера и злоумышленника, так и с ростом числа активных майнеров, и возрастает с ростом в положительном