УДК 004.94 DOI 10.17223/2226308X/10/46

О МЕТОДЕ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ОБНАРУЖЕНИЯ АНОМАЛИЙ В SQL-ЗАПРОСАХ

А. И. Мурзина

Предлагается метод обнаружения аномалий в SQL, основанный на кластеризации и использовании рекуррентных нейронных сетей для разрешённых SQL-запросов.

Ключевые слова: *машинное обучение, обнаружение аномалий, SQL-инъекции, кластеризация, рекуррентные нейронные сети.*

На сегодняшний день SQL-инъекции являются одной из наиболее распространённых атак на веб-приложения [1], с помощью которых можно получить несанкционированный доступ к системам управления базами данных (СУБД). Этот вид атаки основывается на внедрении произвольного SQL-кода, который меняет логику запроса. Известные реальные методы обнаружения SQL-инъекций [2 – 4] используют признаки их классификации и соответствующие сигнатуры и, как правило, не защищают от всех векторов данной атаки. В настоящей работе предлагается метод, позволяющий обнаруживать нетипичные для конкретного приложения, взаимодействующего с СУБД, SQL-запросы, в том числе не зависящие от уже существующих типов SQL-инъекций. Такие запросы в дальнейшем называются аномалиями.

Предлагаемый метод работает в три этапа: формирование вектора признаков из разрешённых (валидных) SQL-запросов обучающей выборки; кластеризация этих запросов на основе вектора признаков; обучение LSTM-сети для классификации на основе результатов кластеризации. Основная идея состоит в том, что SQL-запросы обучающей выборки можно разделить на несколько классов так, что если новый проверяемый SQL-запрос не «похож» ни на один запрос из этих классов, то будем считать его аномальным. Степень «похожести» определяет сама нейронная сеть.

Вектор признаков формируется на основе SQL-запросов обучающей выборки для конкретного приложения. Каждый запрос формирует вектор признаков, где каждый элемент вектора — число от 0 до 1 — частота вхождения лексемы языка SQL в данный запрос. Если лексемы нет в новом запросе, но она есть в запросах выборки, то ей присваивается 0. В силу того, что не все лексемы языка SQL могут использоваться на практике, длина вектора признаков может отличаться для каждого конкретного приложения. Таким образом, по обучающей выборке SQL-запросов строится матрица размера $m \times n$, где m — количество запросов, а n — количество уникальных лексем SQL в выборке.

Как правило, бизнес-логика приложения меняется не так часто, поэтому разные запросы к СУБД различаются пользовательскими параметрами, через ввод в которые и может быть проэксплуатирована SQL-инъекция. Данные запросы можно кластеризовать — разбить множество запросов на группы по схожим признакам. Для кластеризации применяется метод Mean-Shift [5], на вход которому поступает матрица из сформированных на предыдущем этапе векторов, а на выходе — вектор размера m, и для каждого вектора определён класс в виде числа.

Теперь, когда каждый запрос отнесён к какому-то классу, можно обучить LSTM-сеть, которая будет стараться отнести новый неклассифицированный запрос к одной из групп. LSTM (Long short-term memory) — вид архитектуры рекуррентных нейронных сетей (RNN) [6], который может запоминать зависимости между элементами входного

вектора. На этапе обучения на вход данная нейронная сеть принимает вектор признаков и число — номер класса, к которому относится вектор.

На этапе классификации LSTM-сеть принимает вектор признаков нового запроса, а на выходе выдаёт два числа — номер класса, на который наиболее похож запрос, и число от 0 до 1 как вероятность того, что данный запрос принадлежит к этому классу.

В результате работы метод был реализован и подготовлены тестовые данные. В дальнейшем планируется провести серию экспериментов для оценки точности работы метода.

ЛИТЕРАТУРА

1. OWASP Top 10 Application Security Risks — 2017. `https://www.owasp.org/index.php/Top_10_2017-Top_10`
2. *Halfond W. G. J., Viegas J., and Orso A.* A classification of SQL injection attacks and countermeasures // Proc. ISSSE 2006, Washington, USA, Mar. 2006. `https://pdfs.semanticscholar.org/81a5/02b52485e52713ccab6d260f15871c2acdcb.pdf`
3. *Rawat R. and Shrivastav S. K.* SQL injection attack detection using SVM // Intern. J. Comput. Appl. 2012. V. 42. No. 13. P. 1–4.
4. *Buehrer G. T., Weide B. W., and Sivilotti P. A. G.* Using parse tree validation to prevent SQL injection attacks // Proc. SEM'05. N. Y.: ACM, 2005. P. 106–113.
5. *Cheng Y.* Mean shift, mode seeking, and clustering // IEEE Trans. Pattern Analysis Machine Intelligence. 1995. V. 17. Iss. 8. P. 790–799.
6. *Schmidhuber J. and Hochreiter S.* Long short-term memory // Neural Computation. 1997. No. 9. P. 1735–1780.

УДК 004.94 DOI 10.17223/2226308X/10/47

МЕТОД ИДЕНТИФИКАЦИИ СОГЛАШЕНИЙ О ВЫЗОВЕ ФУНКЦИЙ В БИНАРНЫХ ПРИЛОЖЕНИЯХ

М. А. Станчин, Н. В. Сороковиков

Предлагается метод определения соглашений о вызове функций в бинарных приложениях.

Ключевые слова: *статический анализ, бинарные приложения, соглашение о вызове.*

При статическом анализе бинарных приложений возникает задача определения соглашений о вызовах функций. *Соглашением о вызове* (calling convention) называется способ передачи параметров (аргументов) подпрограммам [1]. В известных средствах анализа бинарных приложений (например, Radare2, Angr, IDA Pro) эта задача решается с использованием встроенных инструментов, что, как правило, не позволяет использовать их в собственных разрабатываемых средствах анализа. В данной работе предлагается собственный метод анализа соглашений о вызове функций.

Для определения соглашения о вызове необходимо определить следующие параметры:

1) наличие регистров для передачи параметров в функции;
2) используемые регистры для передачи параметров в функции;
3) используемые области памяти;
4) мнемоника инструкции возврата.