УДК 004.7, 004.05
DOI: 10.20535/SRIT.2308-8893.2016.2.02

ГЕНЕТИЧЕСКИЙ АЛГОРИТМ ЗАЩИТЫ SDN ОТ СЕТЕВЫХ АТАК

С.И. ЗАБЕЛИН

Рассмотрены проблемы обнаружения и блокирования сетевых атак при помощи SDN. Эта проблема сформулирована как задача нахождения «правильного» вектора и является задачей бинарного целочисленного программирования. В качестве сетевой атаки рассматривается DDoS-атака. Построена математическая модель задачи и разработан ее алгоритм, позволяющий выявить множества атакующих хостов по зарегистрированным данным в сети. При обнаружении множества задача сводиться к предотвращению атаки, т.е. блокировке по IP-адресам. Для оценки эффективности предложенного алгоритма оперативного управления трафиком и обнаружения сетевых атак были проведены экспериментальные исследования. Для моделирования SDN сетей использовался эмулятор сетей Mininet. Решена задача бинарного целочисленного программирования. Анализ результатов подтверждает, что, используя преимущества SDN, а именно централизацию управления, гибкость, удалось реализовать генетический алгоритм, защищающий сеть от одной из самых распространённых сетевых атак — DDoS.

ВВЕДЕНИЕ

Ценность SDN сетей состоит в их особенности обеспечивать сетевую виртуализацию, динамическую политику управления данными и полный контроль над сетевыми сущностями по всей сети [1]. Такие протоколы, как OpenFlow позволяют реализовать эти возможности SDN. Но при централизации плоскости управления SDN становится лабиринтом для администраторов с целью обеспечения безопасности и корректного функционирования всей сети. Некоторые вредоносные сетевые сущности могут быть использованы для перехвата ценной информации пользователей или прекращения работы всей сети [6].

В работе рассматривается задача выявления множества атак (IP-адресов злоумышленников) по зарегистрированным данным о событиях в сети. При обнаружении множества задача сводиться к предотвращению атаки, т. е. блокировке по IP-адресам, например, методом black hole filtering (фильтрация черной дырой).

В качестве сетевой атаки рассматривается DDoS-атака — атака типа «отказ в обслуживании» (Distributed Denial of Service) — один из наиболее популярных видов атак на вычислительную систему. Цель — парализовать работу атакуемого веб-узла.

ПОСТАНОВКА ЗАДАЧИ

Пусть N_b — количество типов событий, генерируемых DDoS-атакой.

Пусть N_a — количество известных типов DDoS-атак.

Пусть AE это $N_e \times N_a$ — матрица атак-событий, которая представляет множества событий генерируемых каждой атакой; AE_{ij} — количество событий типа i, сгенерированных сценарием атаки j ($AE_{ij} \geq 0$).

Пусть R это N_a-размерный вектор весов, где R_i ($R_i > 0$) — вес i-й атаки.

Пусть O это N_e-размерный вектор, где O_i — количество событий i-го типа (O — наблюдаемый след).

Пусть H это N_a-размерный вектор гипотезы, где $H_i = 1$, если i-я атака соответствует гипотезе, и $H_i = 0$ — в противном случае.

Чтобы правильно проанализировать наблюдаемый след (O), который генерируется одной или несколькими типами DDoS-атак, необходимо найти вектор H, который максимизирует $R \times H$ (задача пессимиста), при ограничениях $(AE.H)_i \leq O_i$ $(1 \leq i \leq N_a)$.

Нахождение «правильного» H вектора, по сути, является задачей бинарного целочисленного программирования, которое является *NP*-полной задачей. Применить классические алгоритмы невозможно, так как в данной задаче N_e равняется нескольким сотням.

Выбран следующий эвристический подход к решению *NP*-полной задачи: генерируется случайная гипотеза, проверяется правильность гипотезы, тестируется улучшенная гипотеза и т.д. до тех пор, пока решение не будет найдено.

Необходимо найти алгоритм, который мог генерировать новые гипотезы на основе предыдущих гипотез: этим алгоритмом стал генетический алгоритм.

ПОСТРОЕНИЕ ФУНКЦИИ ПРИГОДНОСТИ

Две подзадачи возникают при применении генетического алгоритма для решения задачи:

1) кодирование решения задачи в строку из битов;

2) нахождение функции пригодности как критерия для операции отбора.

Для решения задачи кодирования используется следующий подход: длина каждого индивидуума равна N_a и каждый индивидуум в популяции соответствует конкретному вектору H.

Для нахождения функции пригодности необходимо произвести поиск среди всех множеств возможных атак, представляющих наибольшую опасность системе, т.е. максимизировать произведение R и H. Получаем функцию пригодности:

$$F = \sum_{i=1}^{N_a} R_i.I_i,$$

где I — индивидуум.

Но функция пригодности не учитывает ограничения, т.е. из всего множества возможных индивидуумов не все являются «реалистичными». В этом случае для некоторых типов событий i не будет выполняться $(AE.H)_i \le O_i$. Для выполнений ограничений введем штраф для «нереалистичных» индивидуумов [3]. Введем штрафную функцию P, которая увеличивается, если «реализм» индивидуума уменьшается. Пусть T_e — количество типов событий, для которых $(AE.H)_i > O_i$. Тогда

$$P = T_e^p.$$

Для функции пригодности выберем квадратную штрафную функцию ($p = 2$). Тогда функция пригодности приобретает следующий вид:

$$F(I_i) = \alpha + \left(\sum_{i=1}^{N_a} R_i.I_i - \beta.T_e^{\,2} \right),$$

где β — вес штрафной функции; α — порог для обеспечения положительных значений в функции пригодности. Если негативная пригодность найдена, то соответствующий индивидуум не может быть отобран. Параметр α позволяет отсеять слишком нереалистичные гипотезы.

ГЕНЕТИЧЕСКИЙ АЛГОРИТМ ОБНАРУЖЕНИЯ DDOS-АТАК

Прототип генетического алгоритма обнаружения DDoS-атак находит вектор H, который максимизирует произведение $R.H$ с ограничениями $(AE.H)_i \le O_i$ $(1 \le i \le N_a)$. Если наблюдаемый след слишком длинный, то след ограничивается N_a-размерным вектором. Для экспериментального исследования избран базис для фазы наблюдения 15 с. После фазы обнаружения по наблюдаемым данным заполняются вектора наблюдаемых следов. Разработана матрица атак-событий с 6 типами DDoS-атак и с 28 возможными событиями.

Каждый эксперимент можно охарактеризовать следующим набором параметров: (P_c, P_m, L, a), где P_c — вероятность размножения; P_m — вероятность мутации; L — размер популяции; a — количество атак, присутствующих в следе. Для каждого набора произведено 10 запусков (все результаты усреднены по 10 запускам).

Генетический алгоритм обнаружения DDoS-атак был реализован на языке Python с помощью API контроллера SDN POX [4]. Выбранный контроллер является наиболее удачным для целей исследования. Контроллер POX в своем составе имеет программный интерфейс (API) на языке Python для приложений управления сетью [2].

Для моделирования SDN сетей использовался эмулятор сетей Mininet. Эмулятор позволяет создать виртуальные сети с реальными рабочими компонентами (коммутаторами, рабочими станциями, контроллерами) [5].

Mininet ведет себя как настоящая машина и запускает тот же код, что и она. Хост Mininet представляет собой оболочку машины, в которую будут

запускаться различные программы. Хосты могут посылать, получать и обрабатывать пакеты так, будто программа является настоящим Ethernet, но в сущности является виртуальным коммутатором/интерфейсом. Пакеты виртуальных коммутаторов, которые для Mininet хостов становятся реальными Ethernet коммутаторами или маршрутизаторами, обрабатываются в зависимости от того, как они настроены [1].

РЕЗУЛЬТАТЫ

Наиболее важным аспектом для алгоритма является безопасность, поэтому для исследования выбраны два параметры: T_p и T_a:

T_p отвечает за долю индивидуумов, которые обнаружили существующую атаку;

T_a отвечает за долю индивидуумов, которые обнаружили несуществующую атаку.

В начале работы алгоритма $T_{p0} \cong 0{,}5$ и $T_{a0} \cong 0{,}5$ (так как изначальные популяции сгенерированы случайно). В конце работы алгоритма теоретически параметры должны быть такими: $T_p = 1$ и $T_a = 0$ (все существующие атаки обнаружены, а несуществующие атаки не обнаружены).

Эволюция T_p и T_a при генерации новых поколений показана на рис. 1 и 2. На рис. 1 по оси абсцисс отложены номера поколений (итерации), по оси ординат — T_p и T_a. После 100 поколений T_p и T_a равны 0,997 и 0,0042 соответственно, что близко к их оптимальным значениям.

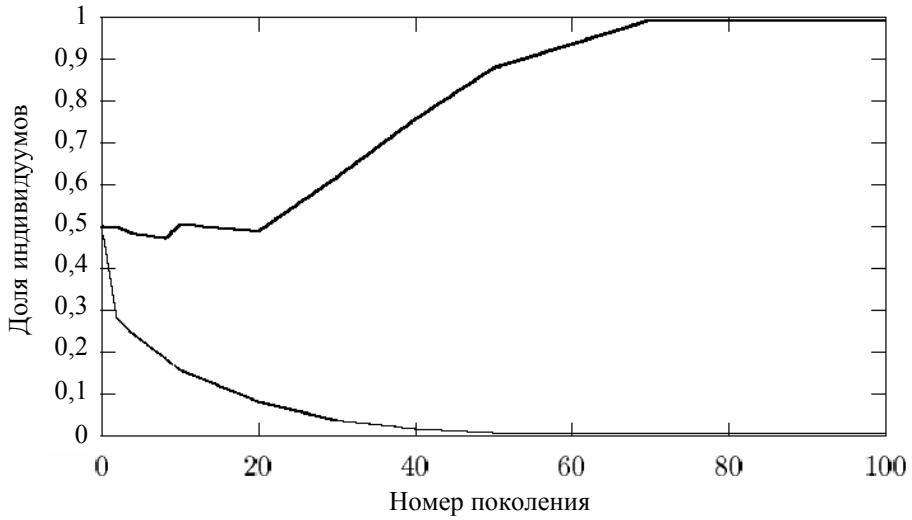

Рис. 1. Эволюция T_p и T_a при генерации новых поколений

Из рис. 2, на котором показана зависимость функции пригодности от номера поколения, видно, что максимальное значение функции пригодности

быстро сходится до оптимума (примерно через 20 поколений). Количество генерируемых атак не повлияло на результат.

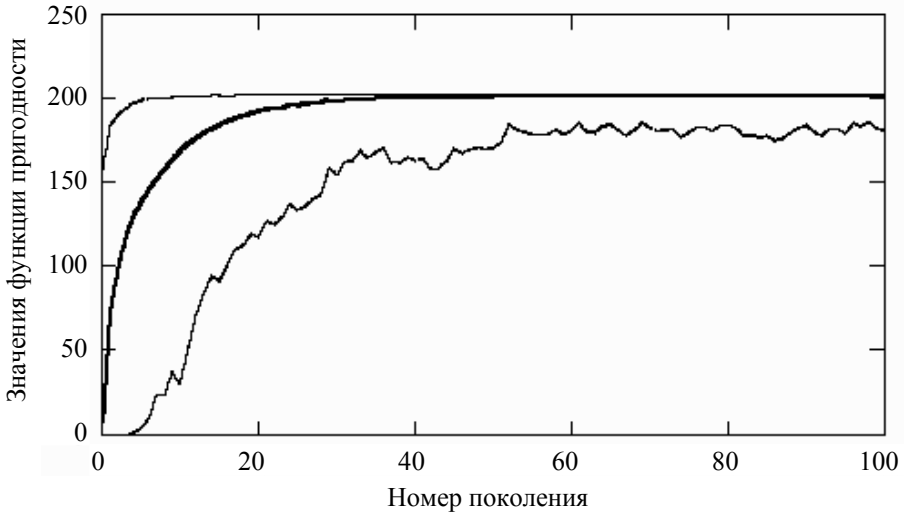

Рис. 2. Значение функции пригодности на протяжении 100 поколений

Соотношение времени выполнения алгоритма (в секундах) при изменении количества DDoS-атак показано на рис. 3. Так, при 200 атаках генетическому алгоритму обнаружения DDoS-атак требуется 10 мин 35 с для анализа.

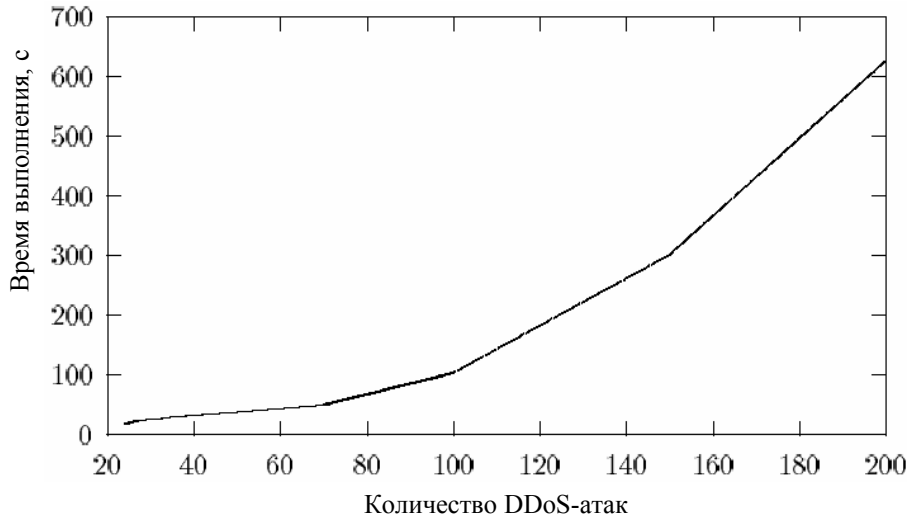

Рис. 3. Время работы выполнения алгоритма (ось абсцисс) при изменении количества DDoS-атак (ось ординат)

ВЫВОДЫ

Используя преимущества SDN, а именно централизацию управления, гибкость, удалось реализовать генетический алгоритм, защищающий сеть от одной из самых распространённых сетевых атак — DDoS.

Приведены результаты экспериментальных исследований, которые направлены на подтверждение применения разработанных методов и инструментов в практической работе.

ЛИТЕРАТУРА

1. *Nadeau T.* SDN: Software Defined Network [Text] / T. Nadeau, K. Gray. — Washington: O'Reilly Media, 2013. — P. 9–11.
2. *Open* Networking Lab - Confluence [Електронний ресурс] : POX Wiki.Ali Al-Shabibi. — Режим доступу: https://openflow.stanford.edu/display/ONL/POX+Wiki
3. *Holland J.* Adaptation in natural and artificial systems [Text] / J. Holland. — University of Michigan Press, Ann Arbor, 1975. — P. 2.
4. *Limoncelli T.* Adaptation in natural and artificial systems [Text] / T. Limoncelli. — ACM, 2012. — 55 p.
5. *Composing* software-defined networks / C. Monsanto, P. Private, A. Monsanto etc. — New York, USA: USENIX NSDI, 2013. — 13 p.
6. *A security* enforcement kernel for openflow networks / P. Porras, S. Shin, V. Yegneswaran etc. — New York, USA: ACM, 2012. — 10 p.

Поступила 28.01.2016