Е.В. Пальчевский, А.Р. Халиков

РАЗРАБОТКА СИСТЕМЫ ОБНАРУЖЕНИЯ НИЗКОАКТИВНОГО НЕСАНКЦИОНИРОВАННОГО СЕТЕВОГО ТРАФИКА

(Уфимский государственный авиационный технический университет)

Одним из ярко выраженных направлений в информационной безопасности являются *DoS*- и *DDoS*-атаки [1]. Под «*DoS*» (*Denial of Service*) подразумевается атака с амплификацией, направленная на переполнение внешнего сетевого канала [2]. «*DDoS*» (*Distributed Denial of Service*) представляет собой распределенную атаку, направленную на переполнение внешнего сетевого канала с последующей перегрузкой физических ресурсов ЭВМ [3]. Данные атаки направлены на отказ в удаленном обслуживании внешнего сетевого ресурса (физического сервера, вычислительного кластера, персонального компьютера) [4]. За последний десяток лет количество таких атак выросло в несколько тысяч раз [5]. Сложность и мощность атак типа «*DoS*» и «*DDoS*» увеличились в несколько раз. К примеру, в 2016 году компанией «*Akamai*» была зафиксирована атака, суммарные мощности потоков которой составляли 623 Гбит/с, тогда как в 2010 году максимальная атак составляла 100 Гбит/с [6]. Вместе с тем, существующие методы обнаружения вредоносного трафика неэффективны для обнаружения низкоактивных атак типа «*DoS*» и «*DDoS*» [7]. Основой для исследования и обнаружения несанкционированного трафика является построение характеристик трафика по задаваемым лимитам внешнего сетевого интерфейса [8-12].

Целью работы является разработка системы обнаружения низкоактивного несанкционированного сетевого трафика. Это позволит снижать нагрузку на ресурсы ЭВМ, а также повысить доступность физического сервера по внешнему сетевому каналу. Первым этапом рассмотрена математическая модель разработанной системы обнаружения вредоносного сетевого трафика. На втором этапе представлены схема работы и фрагмент исходного кода. Третьим этапом являлась апробация, показывающая нагрузку на физические ресурсы ЭВМ.

Математическая модель, с применением цепей Маркова, разработанной системы обнаружения низкоактивного вредоносного трафика представлена на рисунке 1.

Состояния разработанной системы обнаружения низкоактивного сетевого трафика:

- *SL* – состояние проверки сетевого лимита в значениях сетевого стека;
- *PRN1* – состояние проверки сетевой загруженности внешнего сетевого интерфейса;
- *PRN2* – состояние проверки загруженности вычислительными процессами;
- *OBNF* – состояние распределения общей нагрузки (вычислительными процессами и сетевой).

Рисунок 1 – Состояния разработанной системы при активированном режиме

Значение 0,83 показывает время перехода от одного состояния к другому, а также соответствует количеству задействованных дуг. Матрица вероятности переходов строится на основе перехода от состояния к состоянию, суммируя общий проделанный путь.

Таким образом, матрица вероятности переходов:

$$P = \begin{pmatrix} 0{,}83 & 0{,}83 & 0{,}83 & 0{,}83 \\ 0{,}83 & 0{,}83 & 0{,}83 & 0{,}83 \\ 0{,}83 & 0{,}83 & 0{,}83 & 0{,}83 \end{pmatrix}$$

Из матрицы видно, что вероятность перехода в каждое состояние приравнивается к $0 \le p \le 1$. Таким образом, $\sum_{j=1}^{m} p = 1$. Это дает возможность переходов по системе за один шаг, позволяя производить операции с более высокой скоростью.

Схема работы разработанной системы обнаружения низкоактивного трафика представлена на рисунке 2.

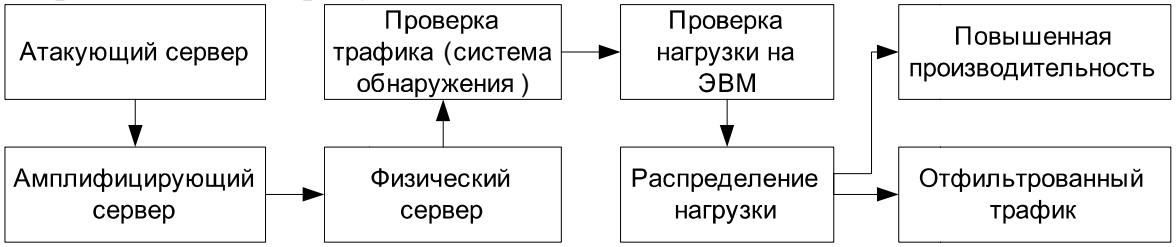

Рисунок 2 – Схема работы разработанной системы обнаружения вредоносного трафика

Фрагмент исходного кода, написанный на языке программирования высокого уровня «*C*» и отвечающий за определение сетевого интерфейса физического сервера.

```
int get_interface_number_by_device_name(int socket_fd, std::string interface_name) {
  struct ifreq ifr;
  memset(&ifr, 0, sizeof(ifr));
  if (interface_name.size() > IFNAMSIZ) {
```

```
    return -1;}
  unsigned int current_block_num = 0;
  struct pollfd pfd;
  memset(&pfd, 0, sizeof(pfd));
  pfd.fd = packet_socket;}
```

Данная операция позволит определять: по какому именно интерфейсу производить блокировку *IP*-адреса.

Апробация разработанной системы низкоактивного сетевого трафика (в активированном и деактивированном режиме, а также в течение десяти дней) представлена в таблице 1. В таблице: 1,00/2,00 – активированный/деактивированный режимы.

Таблица 1 – Тестирование разработанной системы при атаке «*DDoS*»

День	Атака, Гбит/с	Нагрузка на *CPU*, %	Нагрузка на ОЗУ, %	Нагрузка на *SSD*, %
1	0,10	1,00/2,00	0,10/0,20	0,05/0,10
2	0,20	2,00/4,00	0,20/0,40	0,10/0,20
3	0,30	3,00/6,00	0,30/0,60	0,15/0,30
4	0,40	4,00/8,00	0,40/0,80	0,20/0,40
5	0,50	5,00/10,00	0,50/1,00	0,25/0,50
6	0,60	6,00/12,00	0,60/1,20	0,30/0,60
7	0,70	7,00/14,00	0,70/1,40	0,35/0,70
8	0,80	8,00/16,00	0,80/1,60	0,40/0,80
9	0,90	9,00/18,00	0,90/1,80	0,45/0,90
10	1,00	10,00/20,00	1,00/2,00	0,50/1,00

Таким образом, система обнаружения низкоактивного вредоносного сетевого трафика способствует снижению потребления ресурсов центрального процессора, оперативной памяти и твердотельного накопителя в 2 раза. Подобный результат объясняется невозможностью распределения нагрузки, а также исследования трафика в режиме реального времени стандартными средствами. Разработанная система обнаружения внешнего низкоактивного несанкционированного сетевого трафика позволяет снижать нагрузку на ресурсы ЭВМ и фильтровать вредоносный сетевой трафик, с последующим повышением производительности.

Литература

1. Е.В. Пальчевский, А.Р. Халиков. Равномерное распределение нагрузки аппаратно-программного ядра в UNIX-системах. Труды института системного программирования РАН, Том 28 (Выпуск 1), 2016 г., стр. 93-102. DOI: 10.15514/ISPRAS-2016-28(1)-6.

2. Е.В. Пальчевский, А.Р. Халиков. Техника инструментирования кода и оптимизация кодовых строк при моделировании фазовых переходов на языке C++ Труды института системного программирования РАН, Том 27 (Выпуск 6), 2015 г., стр. 87-96. DOI: 10.15514/ISPRAS-2015-27(6)-6.

3. Пальчевский, Е.В. Параллелизация нагрузки аппаратно-программного ядра в UNIX-системах / Е.В. Пальчевский, А.Р. Халиков // Перспективные информационные технологии. – Изд-во: СГАУ, Самара, 2016. – С. 521-525.

4. Пальчевский, Е.В. Разработка методики защиты от несанкционированного трафика при помощи управляемого компонента NGINX / Е.В. Пальчевский, А.Р. Халиков // Сборник научных статей Международной научно-технической конференции «ШЛЯНДИНСКИЕ ЧТЕНИЯ-2016», Пенза, 2016. – С. 92-95.

5. Пальчевский, Е.В. Реализация кластерной мощности на базе процессоров INTEL XEON x5660 / Е.В. Пальчевский, А.Р. Халиков // Труды научно-технической конференции «Суперкомпьютерные технологии», Таганрог, 2016. – С. 83-86.

6. Пальчевский, Е.В. Анализ и фильтрация протоколов в UNIX-подобных системах, посредством IPTABLES / Е.В. Пальчевский, А.Р. Халиков // Приоритетные задачи и стратегии развития технических наук. Изд-во: «Эвенсис», Тольятти, 2016. – С. 6-9.

7. Crist, E.F. Mastering OpenVPN / E.F. Crist., Keijser J.J. – Изд-во: «Packt Publishing», – 2015. – 364 с.

8. Каретин, И.И. Энергосберегающая оптимизация кода за счет использования отключаемых компонентов процессора / И.И. Каретин, В.А. Макаров // Труды Института системного программирования РАН Том 19. – Изд-во: «ИСП РАН», Москва, 2015. – С. 187-194.

9. Дугин, А. Защита от DDoS подручными средствами. Часть 1. DNS Amplification / А. Дугин // Системный администратор. 2016. № 5 (162). Изд-во: Синдикат 13, Москва, 2016. – С. 22-26.

10. Жарова, О.Ю. Метод определения типа атаки по статистическим параметрам сетевого трафика / О.Ю. Жарова, В.А. Федорова // Вопросы радиоэлектроники. Изд-во: Центральный научно-исследовательский экономики, систем управления и информации «Электроника», Москва, 2016. – С. 39-43.

11. Сокольников А.М. Сравнительный анализ подходов к разработке архитектуры и систем управления базами данных для высоконагруженных web-сервисов // Кибернетика и программирование. №4. 2014. С. 1-13.

12. Krylov V., Kravtsov K. DDoS attack and interception resistance IP fast hopping based protocol: 23rd international conference on software engineering and data engineering, sede 2014. 2014. С. 43-48.