электроники. Изд-во: Центральный научно-исследовательский экономики, систем управления и информации «Электроника», Москва, 2016. – С. 39-43.

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

Е.В. Пальчевский, А.Р. Халиков

РАЗРАБОТКА СИСТЕМЫ УВЕЛИЧЕНИЯ ПРОПУСКНОЙ СПОСОБНОСТИ СЕТЕВОГО СТЕКА ДЛЯ ПОВЫШЕНИЯ ОТКАЗОУСТОЙЧИВОСТИ ФИЗИЧЕСКОГО СЕРВЕРА

(Уфимский государственный авиационный технический университет)

В современном мире интенсивно развиваются методы и системы защиты информации [1-3]. Одним из ярко выраженных направлений информационной безопасности является исследование атак несанкционированным трафиком («*DoS*» и «*DDoS*») [4-6]. С каждым годом атаки данного вида не только увеличиваются в мощности, но и становятся все более сложными [7-9]. Исследованиями в области *DDoS*-атак, в настоящее время, занимаются следующие организации: «*DDoS Guard*»: собственная георасределенная сеть фильтрации; «*OVH*»: поставщик телекоммуникационных услуг; «*Qrator*»: распределенная сеть фильтрации с несколькими узловыми точками [10-12]. На сегодняшний день количество *DDoS*-атак возрастает, а их средняя мощность увеличивается на 25-40 Гбит/с в год. Это ставит под угрозу большинство ресурсов, имеющих выход во внешнюю глобальную сеть.

Целью работы является разработка системы увеличения пропускной способности сетевого стека для повышения отказоустойчивости физического сервера. Это позволит снижать нагрузку на ресурсы ЭВМ, а также повысить доступность физического сервера для удаленного обслуживания по внешнему сетевому каналу. Первым этапом рассматривается математическая модель разработанной системы увеличения пропускной способности. На втором этапе представлены схема работы и фрагмент исходного кода. Третьим этапом являлась апробация, показывающая нагрузку на физические ресурсы ЭВМ при сетевой атаке внешним несанкционированным трафиком.

Математическая модель, представляющая собой граф состояний, разработанной системы увеличения пропускной способности сетевого стека (рисунок 1). Состояния разработанной системы увеличения сетевой пропускной способности:

- *LST* – состояние проверки значений лимита в сетевом стеке;
- *PST1* – состояние проверки загруженности физических ресурсов сетевым стеком;
- *PST2* – состояние повторной проверки загруженности физических ресурсов сетевым стеком;
- *OBNF* – состояние распределения общей сетевой нагрузки.

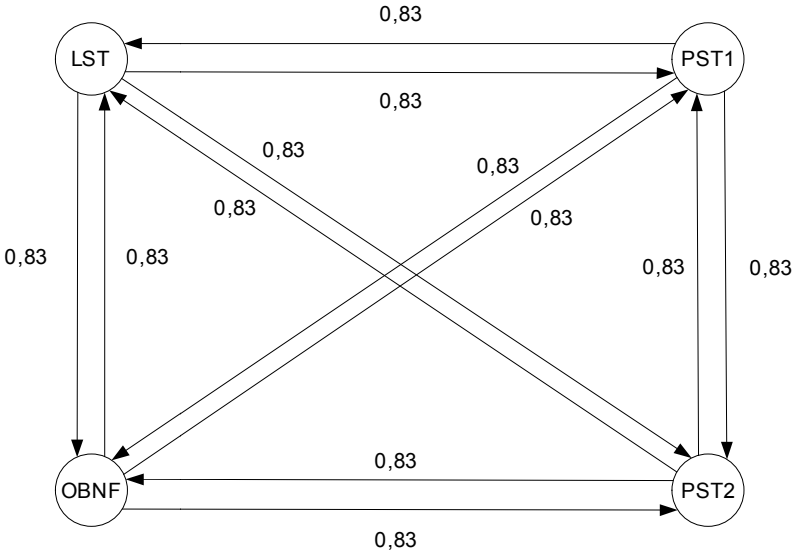

Рисунок 1 – Графы состояний разработанной системы при активированном режиме

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 системы увеличения пропускной способности представлена на рисунке 2.

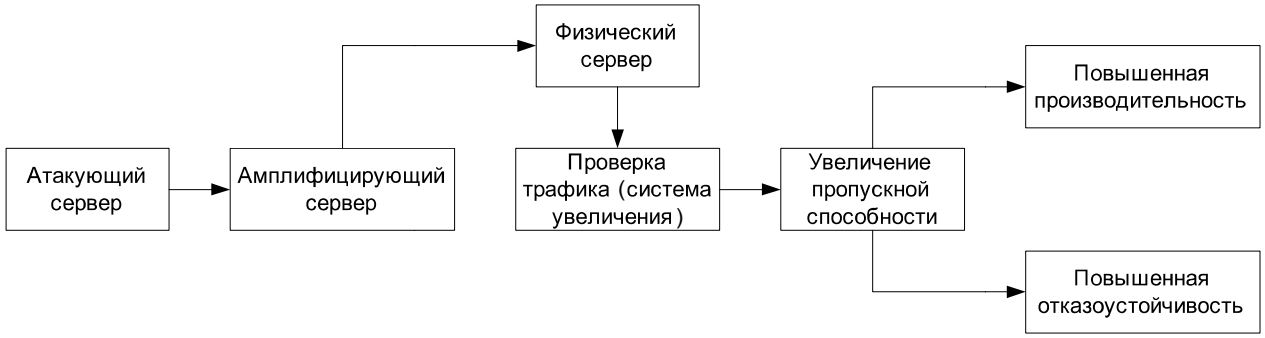

Рисунок 2 – Схема работы разработанной системы увеличения пропускной способности

Фрагмент исходного кода, написанный на языке программирования высокого уровня «*C*» и отвечающий за распределение сетевых потоков по ядрам сервера.

```
if (use_multiple_fanout_processes) {
    boost::thread_group packet_receiver_thread_group;
    unsigned int num_cpus = 24;
    for (int cpu = 0; cpu < num_cpus; cpu++) {
boost::thread::attributes thread_attrs;
void flush_block(struct block_desc *pbd) {
  pbd->h1.block_status = TP_STATUS_KERNEL;}
```

Это позволит равномерно распределить сетевую нагрузку по физическим и логическим ядрам ЭВМ.

Апробация разработанной системы увеличения пропускной сетевой способности (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	Количество входящих пакетов, шт/с	Нагрузка на *CPU*, %	Нагрузка на ОЗУ, %	Нагрузка на *SSD*, %
1	0,10	1000000	12,00/24,00	0,60/1,20	0,15/0,30
2	0,20	3000000	13,00/26,00	0,70/1,40	0,16/0,32
3	0,30	5000000	14,00/28,00	0,80/1,60	0,17/0,34
4	0,40	7000000	15,00/30,00	0,90/1,80	0,18/0,36
5	0,50	9000000	16,00/32,00	1,00/2,00	0,19/0,38
6	0,60	11000000	17,00/34,00	1,10/2,20	0,20/0,40
7	0,70	12000000	18,00/36,00	1,20/2,40	0,21/0,42
8	0,80	13000000	19,00/38,00	1,30/2,60	0,22/0,44
9	0,90	14000000	20,00/40,00	1,40/2,80	0,23/0,46
10	1,00	15000000	21,00/42,00	1,50/3,00	0,24/0,48

Таким образом, система увеличения пропускной способности физического сервера способствует снижению потребления ресурсов центрального процессора, оперативной памяти и твердотельного накопителя в 2 раза. Подобный результат объясняется невозможностью распределения нагрузки, а также изменения значений сетевого стека в режиме реального времени стандартными средствами. Разработанная система увеличения пропускной способности сетевого стека для повышения отказоустойчивости физического сервера позволяет снижать нагрузку на ресурсы ЭВМ и увеличивать лимит входящих сетевых пакетов, 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. Olifer, V.G. Computer networks. Principles, technologies, protocols / V.G. Olifer, N.A. Olifer – Publishing house: SPb. – Peter, 2010.

8. Колисниченко, Д. Linux. От новичка к профессионалу. 2-е издание / Д.Колисниченко, – 2010. – 764 с.

9. Hein R. Linux Administration Handbook/ R. Hein – Изд-во: «Вильямс», – 2007. – 1071 с.

10. Орлофф, Д. Ubuntu – бесплатная альтернатива Windows / Д. Орлофф – Изд-во: «Эксмо», 2009. – 352 с.

11. 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.

12. 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.