ИСПОЛЬЗОВАНИЕ ГПУ ПРОЦЕССОРОВ ДЛЯ КЛАССИФИКАЦИИ ЗАПРОСОВ ПРИ DDOS-АТАКЕ

А.И. Рейбандт, С.А. Сартин

В современном мире очень часто поднимается вопрос о кибербезопастности. Одной из самых распространённых угрозой являются DDoS-атаки. Многообразие видов подобного рода трафика не позволяет иногда не только блокировать DoS или DDoS-атаку, но и в некоторых случаях своевременно опознать её начало.

При DDoS-атаке создаются искусственные условия, при которых легитимные (правомерные) пользователи системы не могут получить доступ к предоставляемым системой ресурсам (серверам), либо этот доступ затруднён.

При этом сетевые устройства и традиционные технологии периметрической защиты, в частности, межсетевые экраны и системы выявления вторжений, хотя и служат важными составляющими стратегии безопасности в целом, не обеспечивают полномасштабную защиту от DDoS.

По данным Arbor Networks Inc.[1] мощность DDoS-атак возрастает в среднем в 2 раза за год и на данный момент может достигать мощности 100 Гбит/сек., а выявленные сети зараженных компьютеров – ботнеты превышали сотни тысяч:

Kraken - 400 тысяч компьютеров.

Srizbi - 315 тысяч компьютеров.

Bobax - 185 тысяч компьютеров.

Rustock - 150 тысяч компьютеров.

Storm - 100 тысяч компьютеров.

Psybot - 100 тысяч ADSL-маршрутизаторов, основанных на Linux.

Понятно, что для обработки запросов даже небольшого по размерам ботнета требуются колоссальные мощности сервера, в то время как процесс анализа и адекватности каждого подключения, займёт на порядок больше ресурсов, в убыток аудитории правомерных пользователей, а возможно и ограничит доступ на какое-то время, что в условиях современного рынка просто не приемлемо. Покупка сервиса защиты от DDoS, наращивание мощностей сервера с усложнением конфигурации являются хоть и эффективными, но всё же пока ещё дорогими способами противодействия злоумышленникам.

Предлагаемый подход является более простым и дешевым решением, основанным на возможности независимой обработки каждого подключения и как следствие достижения большой степени параллелизма. Алгоритм распознавания DDoS-атаки среди пользовательских запросов имеет ряд усовершенствований, отличающих его от уже существующих подходов, в том числе:

- обучение на основе анализа логов (логи Apache, nginx, IIS , логи операционной системы, фаервола, и т.п.);
- классификация трафика нейронной сетью на основе анализа заголовков пакетов
- поведенческий анализ подключения во времени;
- перенос основных ресурсоёмких алгоритмов фильтрации на GPU

Для тестирования алгоритмов распознавания DDoS-атаки, и полного контроля над её ходом, разработана специальная среда, позволяющая в рамках единой рабочей станции моделировать различные виды DDoS-атак. Симулятор создан на базе операционной системы GNU/Linux и гипервизора Xen. Для симуляции атакующих рабочих станций использованы виртуальные машины с операционными системами семейств Windows и Linux с установленными на них инструментарием DDoS: LOIC и Stacheldraht соответственно. В качестве атакуемой рабочей станции выступает Windows Server 2008 R2 с выделенной ей видеокартой Nvidia.

Разработан промежуточный драйвер для взаимодействия с сетевыми картами, включающий в себя возможности пассивного анализа и фильтрации пакетов. В тело драйвера включен флуд-детектор распознающий начало DDoS-атаки и посылающий сигнал более интеллектуальным фильтрам.

Флуд-детектор представляет собой, так называемый первый рубеж защиты, он работает постоянно, в режиме «сниффера» анализирует весь трафик и ведёт запись логов. Данные записи используются в работе статистического анализатора и для обучения нейронной сети.

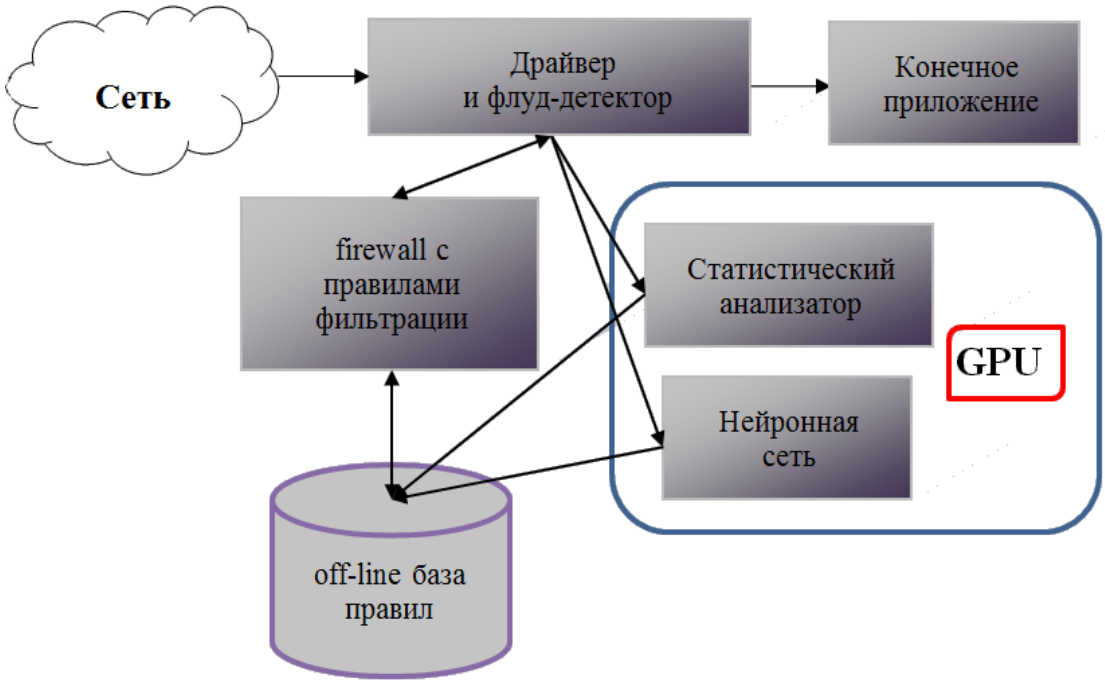

Рис.1 Простейшая схема функционирования алгоритма

При начале DDoS-атаки флуд-детектор посылает сигнал файерволлу, статистическому анализатору и нейронной сети на начало работы.

Статистический анализатор добавляет данные в off-line базу правил основываясь на временном анализе подключения, составляя таблицу с историей запросов данного IP адреса на продолжительном промежутке времени, так, например, IP адреса, создавшие более чем N подключений за некоторое время T, классифицируются как злонамеренные, и т.п.

Нейронная сеть в процессе своей работы оперирует только данными из заголовка пакетов, частоты, размера и битрейта пакетов, и не использует анализ данных, так как использование шифрования подключения может сделать систему фильтрации фактически бесполезной.

В качестве входных параметров используются:

- средний размер пакета с данного IP адреса в промежуток времени T;
- количество пакетов с данного IP адреса в промежуток времени T;
- частота появления SYN, ACK, FIN, PSH, URG, RST флагов с данного IP адреса в промежуток времени T;
- временной интервал между пакетами с одного IP адреса;
- первый октет IP адреса и т.п.

В результате мы имеем достаточно большой объём вычислений на один пакет, что в сочетании с независимостью обработки каждого запроса делает необходимым, в нашем случае, перенос наиболее ресурсоёмкой части алгоритмов на ГПУ.

Наиболее развитой в этой области технологией на сегодняшний день является продукт компании Nvidia – CUDA. Основными преимуществами данной технологии являются открытость архитектуры, кроссплатформенность, полная техническая и информационная поддержка со стороны компании NVIDIA, максимально продуктивное использование ресурсов ГПУ и динамическое развитие технологии. Стоит учесть что программы пишутся на подмножестве языка C, что облегчает изучение технологии и перенос уже имеющихся алгоритмов.

Оптимизация алгоритмов анализа путём переноса на ГПУ с применением алгоритмов поиска, таких как Ахо-Корасик [2] и Ву-Манбера [3] даёт существенный выигрыш в производительности. Так же, к примеру, в [4] авторы показывают, что их реализация обучения нейронной сети на Nvidia CUDA имеет скорость обучения в сотни раз выше, чем на ЦПУ.

В результате перестройка конфигурации системы, с целью использования ГПУ для оптимизации классификации трафика, позволит снизить задержки и уменьшить время отклика для легитимного пользователя. Причём в отличие от существующих подходов аппаратная часть практически не претерпевает изменений. Так же следует отметить, что дальнейшее увеличение скорости обработки трафика потребует незначительных затрат для смены или приобретения дополнительного модуля имеющего базовый ГПУ.

ЛИТЕРАТУРА:

1. http://arbornetworks.com
2. http://aho-corasick.narod.ru

3. Sun Wu, U. Manber. "A fast algorithm for multi-pattern searching"
4. "A Neural Network on GPU". The code project – development resource University of California, USA, 2007. – Mode of access: www.codeproject.com/KB/graphics/GPUNN.aspx– Date of access: 15.11.2009.
5. L. Ming Li, "An approach to reliably identifying signs of DDOS flood attacks based on LRD traffic pattern recognition" //Computers & Security, Vol. 23, Issue 7, Elsevier, ISSN 0167-4048, April 2004
6. L. Feinstein, D. Schnackenberg, R. Balupari, D. Kindred, "Statistical Approaches to DDoS Attack Detection and Response" //DARPA Information Survivability Conference and Exposition, 2003.Shuyuan Jin, Daniel S.
7. Yeung, "A Covariance Analysis Model for DDoS Attack Detection." //IEEE Communications Society, 2004.
8. D. Gavrilis, I. Tsoulos, E. Dermatas, "Feature selection for robust detection of distributed Denial-of-Service attacks using genetic algorithm" //Methods and Applications of Artificial Intelligence: Third Hellenic Conference on AI (SETN 2004), Samos, Greece, May 2004.
9. Lippmann, R., Cunnigham, R. "Improving intrusion detection performance using Keyword selection and neural networks"// Computer Networks, 34 (2000) 596-603.
10. http://www.securelist.com
11. http://www.securitylab.ru
12. http://xen.org
13. http://www.cisco.com
14. http://nvidia.com