Министерство образования и науки Российской Федерации
Московский физико-технический институт
(национальный исследовательский университет)

Физтех-школа радиотехники и компьютерных технологий
Кафедра системного программирования ИСП РАН

Выпускная квалификационная работа бакалавра

Автоматическая генерация сигнатур сетевых протоколов и приложений

Автор:
Студент Б01-009а группы
Дурнов Алексей Николаевич

Научный руководитель:
канд. физ.-мат. наук
Гетьман Александр Игоревич

Москва 2024

Аннотация

Автоматическая генерация сигнатур сетевых протоколов и приложений

Дурнов Алексей Николаевич

Краткое описание задачи и основных результатов, мотивирующее прочитать весь текст

Содержание

1 Введение

Интернет-провайдеры и сетевые администраторы хотят идентифицировать тип сетевого трафика, который проходит через их сеть, для того, чтобы предоставлять своим клиентам лучший сервис, предлагая им высокое качество обслуживание (QoS), а также планировать свою инфраструктуру и управлять ею. Сетевой трафик можно классифицировать как в соответствии с использующимся протоколом, так и в соответствии с использующимся приложением. Вторая классификация более трудоёмкая, но позволяет решать более широкий спектр задач: формирование трафика в сети, улучшение качества обслуживание, а также предоставление более детального биллинга.

В области классификации сетевого трафика было проведено множество исследований, что привело к разработке многих методов. Самый наивный метод классификации сетевого трафика это идентификация по номеру порта. Однако современные приложения используют динамическое распределение портов и туннелирования трафика, например, по протоколу HTTP, данный метод даёт очень плохие и неточные результаты. Чтобы преодолеть эти ограничения идентификации были введены более совершенные методы.

Первый подход основан на сопоставлении сигнатур полезной нагрузки. Сигнатура полезной нагрузки - это часть данных полезной нагрузки, которая является статичной и различимой для приложений и может быть описана, как последовательность строк или шестнадцатиричных чисел. Второй подход основан на алгоритмах машинного обучения. Для этого подхода используются такие признаки потоков и пакетов, как задержки между пакетами, размеры пакетов и другие, а полезная нагрузка пакетов не анализируется, поэтому он менее точен, чем сигнатурный подход. Однако он может применяться для зашифрованного сетевого трафика, так как в приближении полезная нагрузка зашифрованного трафика представляет собой белый шум, поэтому сигнатурный подход невозможен. Также существуют и гибридные методы, которые используют эти два подхода вместе.

Методы, которые анализируют полезную нагрузку пакетов, являются очень эффективными и точными для идентификации трафика. Их часто называют глубокой проверкой пакетов (DPI). DPI является очень трудоемким и ресурсоемким процессом. Система DPI должна искать сигнатуры в полезной нагрузке пакетов, чтобы точно классифицировать поток. Такой подход не нов, системы обнаружения вторжения (IDS), основанные на DPI, с помощью сигнатур находят интернет-червей и другой трафик, угрожающий безопасности. Далее рассматриваемые методы будут сосредаточены на идентификации приложений среди безопасностного трафика.

Поначалу сигнатуры извлекались вручную. Постоянное появление новых приложений и их частые обновления подчёркивают необходимость автоматической генерации сигнатур, так как ручная операция извлечения сигнатур занимает много времени, а также может быть разница в качестве сигнатур в зависимости от оператора извлечения. Методы автоматической генерации сигнатур должны быть основаны не на семантическом анализе протоколов, так как хотя они и повышают точность сигнатур, но не могут быть применены к анализу высокоскоростного трафика в режиме реального времени.

Поэтому данная работа посвещена исследованию различных методов автоматической генерации сигнатур полезной нагрузки для классификации сетевого трафика по протоколам и приложением.

2 Постановка задачи

Целью данной работы является разработка и реализация метода автоматической генерации сигнатур полезной нагрузки сетевого трафика для классификации этого трафика в соответствии с использующимся протоколом или приложением в режиме реального времени.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Провести исследование литературы по соответствующей теме.
2. Собрать набор сетевых трасс для последующего тестирования и сравнение методов.
3. Выбрать оптимальный метод для автоматической генерации сигнатур.
   (a) Выбрать оптимальный набор параметров метода для каждого тестируемого протокола и приложения, если метод обладает настраиваемыми параметрами.
   (b) Найти ограничения рассматриваемых методов.
4. Встроить генератор сигнатур и классификатор как модули в систему анализа высокоскоростного сетевого трафика, разрабатываемую в ИСП РАН.

3 Обзор существующих решений

3.1 Формат сигнатур

Прежде чем говорить непосредственно о методах автоматической генерации сигнатур, стоит сначала понять какие бывают сигнатуры и в каком формате они будут представлены.

Существует несколько представлений сигнатур. Некоторые из этих видов использовались для представлений сигнатур червей. Однако для обычного трафика можно выделить два основных представления:

1. сигнатуры, представленные регулярными выражениями
2. сигнатуры, представленные строками

Использование регулярных выражений для описания сигнатур приложений становится очень распространённым в классификации потоков. Однако процесс сопоставления регулярных выражений требует огромной вычислительной мощности, которая не масштабируется для идентификации сетевого трафика в режиме реального времени. Способ построения регулярного выражения оказывает непосредственное влияние на классификацию потоков и на общую производительность сопоставления. Несмотря на это, некоторые системы DPI используют регулярные выражения для представления сигнатур приложений. Система обнаружения/предотвращения вторжений Snort (IDS/IPS) имеет более 1000 подписей приложений и предлагает пользователю возможность вставлять новые регулярные выражения по требованию.

Представление сигнатур в виде строк это компромисс между мощностью выражения и эффективностью сопоставления. Также такой подход позволяет преобразовывать сигнатуры, представленные строками, в регулярные выражения.

Будем дальше рассматривать сигнатуры в виде строк, преобразование в регулярные выражения останется за рамками данной работы.

3.2 Структура сигнатур

Большинство форматов сигнатур в предыдущих работах представляют собой простые подстроки, которые часто появляются в полезной нагрузке. Следовательно, всё ещё существует вероятность того, что извлеченные сигнатуры полезной нагрузки могут быть не специфичными для конкретного приложения, некоторые могут принадлежать и другому приложению. Это называется избыточностью сигнатур.

Выделим три типа сигнатур:

1. сигнатура содержимого (полезной нагрузки),
2. сигнатура пакета,
3. сигнатура потока.

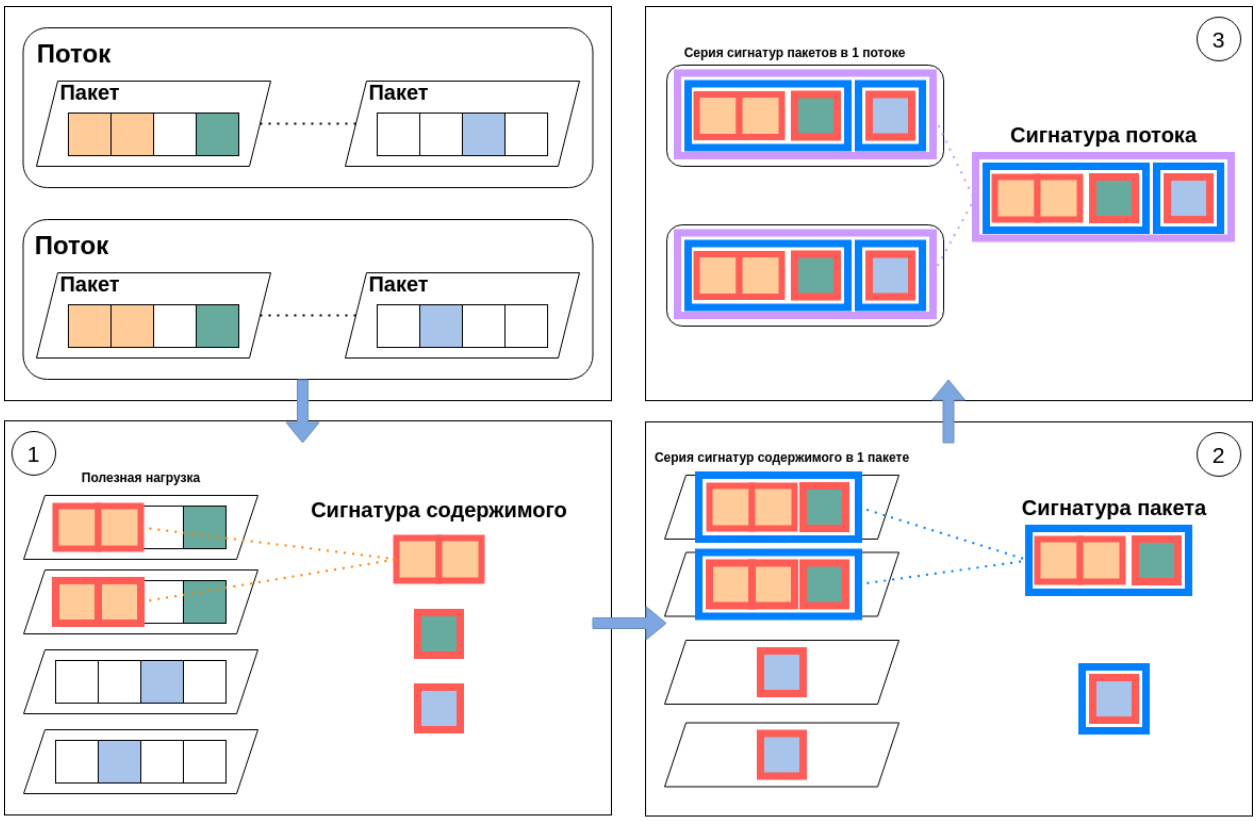

Рис. 1: Процесс извлечения предлагаемой структуры сигнатур полезной нагрузки

Сигнатура содержимого определяется как различимая и уникальная подстрока полезной нагрузки, состоящая из непрерывных символов или шестнадцатеричных значений. На самом деле уникальность с помощью одной подстроки тяжело обеспечить, например, такие строки "GET"или "HTTP которые часто встречаются в HTTP, не могут служить конечными сигнатурами, так как они не различают приложения.

Сигнатура пакета состоит из серии сигнатур содержимого, которые появляются в одном пакете. Так как классификация может выполняться без накопления пакетов, т.е. без сбора потока, то анализируется всегда хотя бы один пакет. Это значит, что для классификации не имеет смысла использовать отдельно сигнатуру содержимого.

Сигнатура потока состоит из серии сигнатур пакетов, которые появляются в одном потоке, где под потоком понимается набор пакетов, имеющий одни и те же IP-адресс источника, IP-адресс назначения, порт источника, порт назначения и используемый протокол транспортного уровня. Сигнатура потока гораздо более специфична для конкретного приложения, чем сигнатура пакета, и значительно повышает точность.

3.3 Метрики оценки качества сигнатур

Для оценки качества получаемых сигнатур рассмотрим матрицу ошибок: 4 стандартные категории, к котором можно отнести результат работы классификатора на полученной сигнатуре. В нашем случае рассматриваемый класс это целевой протокол или приложение. Под классификацией трафика будем понимать классификацию конкретного пакета или потока в зависимости от того, какой уровень сигнатур используется.

	Принадлежит классу (P)	Не принадлежит классу (N)
Предсказана принадлежность к классу (T)	TP	TN
Предсказано отсутствие принадлежности к классу(F)	FP	FN

- истинно положительный (TP): указывает, что трафик правильно классифицирован, как относящийся к определенному классу.
- истинно отрицательный (TN): указывает, что трафик правильно классифицирован, как не относящийся к определенному классу.
- ложно положительный (FP): указывает, что трафик неправильно классифицирован, как относящийся к определенному классу.
- ложный отрицательный (FN): указывает, что трафик неправильно классифицирован, как не относящийся к определенному классу.

Наиболее часто используемые показатели для классификации трафика определяются следующим образом:

- Accuracy (достоверность) = $\frac{TP+TN}{TP+TN+FP+FN}$ - доля правильных классификаций.
- Recall (полнота) = $\frac{TP}{TP+FN}$ - отношение верно классифицированного трафика определенному классу к общему числу трафика этого класса, т.е. описывает способность сигнатуры обнаружить данный целевой протокол или приложение.
- Precision (точность) = $\frac{TP}{TP+FP}$ - доля верно классифицированного трафика среди всего трафика, который классификатор отнёс к этому классу.
- F-мера = $2\frac{recall \cdot precision}{recall+precision}$ - гармоническое среднее между точностью и полнотой. Метрика accuracy может терять свой смысл в задачах с сильно неравными классами. Напротив же recall и precision не зависят от соотношения классов и поэтому применимы в случае несбалансированных классов, что является правдой для сетевого трафика. Часто на практике возникает задача найти оптимальный баланс между precision и recall. Для этих целей подходит F-мера, которая достигает максимума при recall и precision равным 1, и стремится к минимуму, если хотя бы один из параметров стремится к нулю.

Для сигнатур полезно ещё ввести такое понятие как:

- Redundancy (избыточность) опредялется как:

  $$\text{Redundancy} = \frac{\text{Объём трафика идентифицированный двумя и более сигнатурами}}{\text{Объём трафика идентифицированый набором сигнатур}}$$

  Redundancy имеет значение от 0 до 1, где 0 - наилучшее значение, которое указывает на то, что все сигнатуры набора классифицируют исключительно только свою часть трафика, т.е. являются уникальными и незаменяемыми. Если redundancy близка к 1, то в наборе присутствует ненужные сигнатуры, которые идентифицируют перекрывающийся трафик. По мере увеличения количества сигнатур увеличиваются и накладные расходы системы, поэтому данное значение должно оставаться низким.

3.4 Обзор существующих методов автоматической генерации сигнатур

4 Исследование и построение решения задачи

Здесь надо декомпозировать большую задачу из постановки на подзадачи и продолжать этот процесс, пока подзадачи не станут достаточно простыми, чтобы их можно было бы решить напрямую (например, поставив какой-то эксперимент или доказав теорему) или найти готовое решение.

5 Описание практической части

Если в рамках работы писался какой-то код, здесь должно быть его описание: выбранный язык и библиотеки и мотивы выбора, архитектура, схема функционирования, теоретическая сложность алгоритма, характеристики функционирования (скорость/память).

6 Заключение

Здесь надо перечислить все результаты, полученные в ходе работы. Из текста должно быть понятно, в какой мере решена поставленная задача.