ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Департамент программной инженерии

СОГЛАСОВАНО
Научный руководитель
профессор департамента
программной инженерии
факультета компьютерных наук
канд. техн. наук
_____ С.М. Авдошин
«___» _____ 201_ г.

УТВЕРЖДАЮ
Академический руководитель
образовательной программы
«Программная инженерия»
профессор департамента программной
инженерии, канд. техн. наук
_____ В.В. Шилов
«___» _____ 201_ г.

Отчет
по курсовой работе

Исследование и разработка методов деанонимизации пользователей сети Tor

по направлению подготовки бакалавров 09.03.04 «Программная инженерия»

Выполнил:
студент группы БПИ141
образовательной программы
09.03.04 «Программная инженерия»
Лазаренко А.В.

_____
Подпись, Дата

Москва, 2016

Реферат

Отчет 28 с., 6 рис., 2 табл., 51 источн., 2 прил.

Ключевые слова: *анонимные сети; луковая маршрутизация; теневой интернет; оверлейные сети; пиринговые сети; слоистое шифрование; DoS; Tor; анализ траффика; деанонимизация; тайминг-атаки;*

В отчете представлены результаты курсовой работы на тему “Исследование и разработка методов деанонимизации пользователей сети Tor“, выполненной на основе приказа Национального исследовательского университета "Высшая школа экономики" № 6.18.1-02/1112-19 от 11.12.15.

Объект исследования – анонимная сеть Tor.

Предмет исследования - технологии идентификации пользователей в анонимной сети.

Цель исследования – построение программного решения, позволяющего деанонимизировать пользователей сети Tor.

Задачи исследования:

- изучение наиболее эффективных и применимых на практике методов деанонимизации пользователей сети Tor;
- построение классификации методов деанонимизации на основе ресурсов, использующихся атаками;
- построение структуры интеллектуальной системы деанонимизации пользователей с использованием облачных технологий и методов машинного обучения;
- разработка программы из компонент с открытым исходным кодом и базовым алгоритмом машинного обучения для деанонимизации пользователей на практике;
- провести эксперимент, позволяющий определить точность деанонимизации пользователей Tor, с помощью предложенного программного решения.

Методы исследования:

- изучение монографических публикаций и статей;
- сравнительный анализ;
- машинное обучение.

Научная новизна работы состоит в следующем:

1. Построена классификация методов деанонимизации;
2. Построена структура интеллектуальной системы деанонимизации пользователей;
3. Предложена методика определения точности деанонимизации.

Достоверность научных результатов подтверждена результатами экспериментальных исследований с прототипом программы деанонимизации пользователей.

Практическая значимость. Результаты работы могут быть использованы правоохранительными органами и корпорациями, борющимися с распространением нелегального контента через анонимные сети, при создании промышленных систем деанонимизации пользователей в государственном и частном секторе в условиях импортозамещения, а так же, могут быть использованы в качестве фундамента для разработки более продвинутых методов и программных комплексов.

Результаты работы

- изучены самые распространенные анонимные сети и методы деанонимизации пользователей в этих сетях;

- построена классификация методов деанонимизации на основе ресурсов, использующихся атаками;
- построена структура интеллектуальной системы деанонимизации пользователей;
- разработана программа для деанонимизации пользователей сети Tor;
- проведен эксперимент, подтверждающий деанонимизацию пользователей Tor, и позволяющий определить точность деанонимизации.

Апробация работы

Основные положения и результаты курсовой работы докладывались и обсуждались на научных конференциях и семинарах:

1. Межвузовская научно-техническая конференция студентов, спирантов и молодых специалистов им. Е.В. Арменского. 18 февраля 2016. (получен диплом 1-ой степени за лучший доклад см. рисунок 5 в приложении 1).
2. Экспертный совет отборочного тура конкурса инновационных проектов по программе «Участник молодежного научно-инновационного конкурса» (УМНИК), 26 февраля 2016. (получен диплом победителя см. рисунок 6 в приложении 2).
3. Научно-практический семинар «Новые информационные технологии в автоматизированных системах», МИЭМ НИУ ВШЭ, 21 апреля, 2016.
4. Научно-практический семинар «Технологии разработки и анализа программ», ВМК МГУ, 21 апреля, 2016.
5. Spring/Summer Young Researchers' Colloquium on Software Engineering, SYRCoSE 30 May – 1 June 2016, (Статья System for Deep Web Users Deanonimization отрецензирована тремя экспертами, все трое поставили оценку «принять»).

Публикации

1. Авдошин С.М., Лазаренко А.В. Технология анонимных сетей. Журнал «Информационные технологии», том 22, №4, стр. 284-291.
2. Авдошин С.М., Лазаренко А.В. Методы деанонимизации пользователей Tor. Журнал «Информационные технологии», том 22, №5, стр. 362-372.
3. А.В. Лазаренко. Структура интеллектуальной системы деанонимизации пользователей сети Tor. Материалы межвузовской научно-технической конференции студентов, аспирантов и молодых специалистов им. Е.В. Арменского, 2016 год, стр. 81-83.
4. А.В. Лазаренко. Технологии деанонимизации пользователей Tor. Новые информационные технологии в автоматизированных системах: материалы девятнадцатого научно-практического семинара. –М.: ИПМ им. М.В. Келдыша: 2016, стр. 257-262.

Оглавление

1 Введение

На сегодняшний день сеть Tor [1] является самой большой в мире развернутой анонимной сетью. Ежемесячное количество активных пользователей сети превышает два миллиона человек, а количество волонтерских серверов, используемых в качестве узлов сети, превосходит шесть тысяч [2].

Поскольку помимо обычных пользователей, преимуществами анонимизации траффика пользуются террористы, продавцы наркотиков и оружия, а так же прочие нарушители закона, то деанонимизация пользователей является достаточно актуальной и важной задачей для специальных служб различных государств [3, 4]. Так, например, МВД РФ объявляло тендер на разработку способов деанонимизации пользователей сети Tor [5].

Прогресс, достигнутый в разработке методов деанонимизации, позволил американским специальным службам осуществить ряд успешных операций по борьбе с наркоторговлей. Так, например, был закрыт доступ к самому большому в теневом интернете магазину наркотиков Silk Road [6].

Tor состоит из волонтерских серверов, являющихся узлами сети. Пользователи через луковый прокси (далее ЛП) загружают список узлов из сервера каталогов и строят анонимные туннели (цепи), используя луковую маршрутизацию. ЛП строит цепь, как правило, состоящую из трех узлов: входной узел (guard), промежуточный узел (middle), выходной узел (exit). Время жизни цепи составляет 10 минут (по умолчанию). Входной узел обычно выбирается из фиксированного набора из трех узлов, уникального для каждого ЛП. Более детальное описание Tor можно найти в работе [1].

Для организации атак необходимо обладать некоторыми ресурсами, например, коррумпированными узлами Tor, или серверами, доступ к которым пытается получить пользователь.

Поскольку сеть Tor является оверлейной сетью, она работает на основе транспортного слоя. Основными организациями, управляющими интернет маршрутизацией являются автономные системы (далее, АС). Атакующий может контролировать одну или несколько АС и, предполагается, что он наблюдает траффик, проходящий через АС. Анализ влияния АС на деанонимизацию пользователей Tor можно найти в работе [7].

Как правило, атакующий преследует цель скомпрометировать как можно больше цепей, относящихся к конкретному пользователю или группе пользователей, поскольку компрометирование цепей влечет за собой деанонимизацию пользователей.

2 Методы деанонимизации

Существует достаточно большое количество трудов, посвященных атакам на приватность пользователей в анонимных сетях. Многие из описанных методов работают и для других анонимных сетей, например, таких как: I2P, Turtle. В целом, все атаки делятся на пассивные и активные.

Пассивные атаки – такие атаки, при которых атакующий не изменяет траффик, а лишь просматривает его. Активные же, в свою очередь, модифицируют каким-либо образом траффик пользователей.

Любые атаки требуют наличия каких-либо ресурсов у атакующего, например, коррумпированных узлов. Под коррумпированным узлом мы будем понимать такой узел, который атакующий встроил внутрь сети Tor и имеет к нему доступ суперпользователя. Таким образом, атакующий на коррумпированном узле может просматривать и изменять протекающий траффик, устанавливать и модифицировать программное обеспечение на нем. Под коррумпированным сервером мы будем понимать обычный веб-сервер, контролируемый атакующим.

2.1 Пассивные методы

2.1.1 Атаки анализа траффика

Анализ траффика это извлечение информации из мета-данных сети, включая объем и временные характеристики сетевых пакетов. Наблюдатель использует эти данные, для выявления связи между инициатором сообщения и конечной точкой назначения. Из-за того, что внутренние механизмы Tor скрывают битовые шаблоны данных, передающихся через цепь, атакующий не может использовать информацию, содержащуюся в сообщении.

Традиционные атаки анализа траффика делятся на два класса.

В первом классе атак анонимная сеть представляется в виде черного ящика, и рассматриваются временные связи между инициацией пользователем соединения и соединениями, установленными вне сети. Данный класс атак подробно не рассматривается в настоящей статье, поскольку он не применим к глобальной сети из-за ее размеров и огромного количества ресурсов, необходимых для проведения атаки. В работе [8] представлено подробное описание атак этого класса. Статистический вариант атак представлен в работе [9], в работе [10] подтвержден экспериментально. Такие атаки эффективны, если атакующим обозревается большая часть сети и имеется возможность регистрировать пользователей, заходящих в сеть и внешние сервисы. Временной анализ атак рассмотрен в работах [10, 11].

Второй класс атак рассматривает траффик и нагрузку на каждом узле только внутри анонимной сети. В работе [12] представлено детальное описание данных атак. В действительности, атакующий просматривает поток данных на каждом узле, например, ответ сервера инициатору запроса. Поток в узле представляется в виде зависимости объема траффика от времени, затем, производится сглаживание этой функции за счет свертки ее с экспоненциально-убывающей функцией, с целью получения шаблона, предсказывающего вид траффика в анонимной сети. Далее, все фрагменты траффика в сети, преобразованные таким же образом, сравниваются с шаблоном. Степень сходства с шаблонами позволяет определить его принадлежность к конкретному узлу в цепочке соединения. Похожие атаки представлены в работах [13, 14].

Очевидная проблема таких атак заключается в том, что атакующий должен просматривать все узлы в сети и иметь возможность регистрировать мета-данные траффика. В результате, данные атаки используют модель пассивного глобального наблюдателя, который не рассматривается в модели угроз Tor. Заметим, что если наблюдатель обозревает не всю сеть, а ее какую-то часть, то он имеет возможность отследить случайные соединения. Исследование эффективности атак анализа траффика в зависимости от реальных моделей атакующего представлены в работе [15]

Малозатратная атака анализа траффика сети Tor [16] базируется на следующем наблюдении: высокая нагрузка на один из узлов в цепи влияет на задержку всех остальных узлов. С помощью перенаправления соединения через специфические узлы и замер возникающих задержек, наблюдатель может получить примерную нагрузку на узел. Эта нагрузка может быть сравнена с известным шаблоном траффика с помощью традиционных техник анализа траффика [12].

Любой пользователь Tor может произвести такие измерения и измерить нагрузку на узел. Предполагается, что атакующий контролирует коррумпированный узел. Этот узел инициирует соединение, проходящее через другие узлы, нагрузка на которых должна быть измерена. Затем, коррумпированный узел наполняет соединение пробным траффиком, который позволяет измерить задержки и вычислить нагрузку на узел, являющийся целью атакующего.

Просматривая соединение пользователя с сетью Tor или соединение сети Tor с конечной точкой назначения, наблюдатель может использовать эту технику для определения узлов, через которые ретранслировалось соединение. Существует более мощный вариант данной атаки: наблюдатель контролирует сервер, к которому подключается отслеживаемый пользователь. Сервер посылает пользователю через Tor

данные, обладающие специфическим шаблоном траффика, что позволяет идентифицировать узлы пользователя.

Малозатратная атака на основе маршрутизации [17] предполагает, что анонимность может быть скомпрометирована с помощью локального набора коррумпированных узлов. Предполагается, что коррумпированные узлы этого набора имеют высокую пропускную способность.

Для применения атаки, наблюдатель должен контролировать набор, состоящий из более, чем одного узла из списка активных узлов Tor.

На первом шаге необходимо внедрить коррумпированные узлы в качестве входных и выходных узлов в цепи пользователя. Существенное сокращение необходимых ресурсов происходит за счет использования узлов с низкой пропускной способностью. Данное улучшение основывается на том, что узел может предоставить недостоверные данные о своих ресурсах службе каталогов, представляясь мощным узлом для всей системы.

Однако вычислительных ресурсов узла должно хватать для установки новых соединений. Таким образом, все ресурсы узла направляются на прием новых соединений от ЛП.

Если коррумпированные узлы находятся в промежуточных позициях (между входным и выходным узлом), они могут разорвать цепь, поскольку такая конфигурация несовместима с атакой. Это вызовет перестройку цепи и, возможно, после этого установится конфигурация, при которой коррумпированные узлы станут концевыми. Наличие коррумпированных входного и выходного узла является обязательным условием для успешного применения атаки, поскольку иная конфигурация не позволит производить корреляцию траффика.

Если цепь полностью состоит из коррумпированных узлов, то компрометирование анонимности является тривиальной задачей. В наиболее вероятном варианте, компрометированные узлы будут только входными и выходными для данного пользователя. Данный тип атаки позволяет компрометировать цепь до того, как пользователь начинает пересылать данные.

Для того, чтобы собрать достаточно информации для корреляции клиентских запросов с ответами сервера через Tor, каждый коррумпированный узел регистрирует следующую информацию для каждой ячейки с данными: ее позицию в цепи (на входном, выходном узле или в середине), местное время, ID прошлой цепи, прошлый IP адрес, прошлый порт соединения, IP адрес следующего промежуточного соединения, порт следующего промежуточного соединения, ID цепи следующего промежуточного соединения. Как только эти данные будут собраны, атакующий может связать пути, в которых содержатся выходной и входной коррумпированные узлы, с ЛП, от которых идут запросы построения цепей. Обладая этой информацией, атакующий может связать адресанта с адресатом, компрометируя анонимность системы.

Для использования атаки данного типа, коррумпированные узлы должны быть координированы. Самый простой способ координации - использовать централизованный сервер для сбора логов с узлов, что позволит наблюдателю выполнить алгоритм компрометирования цепи в реальном времени.

Алгоритм компрометирования цепи работает следующим образом: коррумпированный узел подтверждает, что запрос на создание цепи был отправлен ЛП, а не узлом. Затем, проверяется хронологический порядок шагов создания цепи и, что промежуточное соединение для входного узла совпадает с промежуточным соединением выходного. После этого проверяется, что сообщение, посланное от входного узла выходному, получено для ответа от выходного узла. Если все проверки выполнены успешно, то цепь компрометирована.

В работе [17] приведены результаты экспериментов, которые проводились с использованием закрытой, развернутой сети Tor. Она состояла из 60 легитимных и 6 коррумпированных узлов. Атака успешно скомпрометировала 46% цепей.

Атаки на основе пропускной способности канала [18] являются незаметными как для пользователя, так и для узла. Пропускная способность канала Tor может быть использована в качестве следа для обнаружения узла с минимальной пропускной способностью (далее, УМПС). Через подробное рассмотрение динамики пропускной способности канала, можно выявить, что два потока используют один и тот же набор узлов, что позволяет атакующему связать потоки с цепями и скомпрометировать приватность пользователя.

Фундаментальное наблюдение, используемое в атаке – разнородность пропускных способностей узлов Tor [18]. Существует три ключевых фактора, определяющих пропускную способность цепи: a) пропускная способность УМПС, b) количество активных TCP потоков между УМПС и следующим узлом, c) количество других активных цепей, размноженных через TCP соединение. Т.е., основная идея атаки – корреляционный анализ пропускных способностей цепей.

Первый вариант атаки позволяет определить, что две цепи используют общий набор узлов. На основе мониторинга пропускных способностей узлов проводится простой статистический тест для выявления корреляции между ними. Существует три возможных варианта:

- Обе цепи используют одинаковый набор узлов. При таком варианте коэффициент корреляции пропускных способностей цепей будет высок. Это объясняется тем, что любые вариации потоков или пропускных способностей узлов будут влиять на пропускную способность цепей одинаково
- Обе цепи не имеют общих узлов. В этом случае, степень корреляции будет близка к нулю.
- Обе цепи имеют как минимум один общий узел. В случае, если общий узел является УМПС в обеих цепях, тогда пропускные способности цепей будут иметь высокий коэффициент корреляции. Иначе, изменения потоков или пропускных способностей узлов при пересылке данных не будут влиять на пропускную способность обеих цепей и их пропускные способности будут зависеть от их УМПС.

Второй вариант атаки – идентификация одного или нескольких узлов в качестве участника какого-либо целевого потока (далее, ЦП) [18]. Этот поток может быть любым потоком, инициированным ЛП через сеть. Предполагается, что атакующий может сформировать пробный поток, просматривать траффик ЦП и имеет коррумпированный выходной узел. Кроме этого, у наблюдателя должен быть либо веб-сервер, к которому пользователь пытается получить доступ или ISP, ретранслирующий данные. Атакующему не нужно модифицировать траффик. Он строит цепь с одним узлом, через эти узлы, вычисляет корреляцию между ЦП и пробным потоком. Если пропускная способность имеет высокий коэффициент корреляции с пропускной способностью ЦП, тогда сервер может допустить, что оба потока проходят через общий узел.

Website fingerprinting (далее WF) - класс атак получивший большую популярность среди исследователей [19, 20, 21]. Он демонстрирует, что наблюдатель, способный просматривать зашифрованный траффик какой-то части сети, имеет возможность, при некоторых условиях, скомпрометировать портал, посещенный пользователем в сети. Атакующий должен иметь доступ к входному узлу пользователя, чтобы просматривать его траффик и видеть IP адрес пользователя.

Стратегия атакующего следующая: он пытается смоделировать сетевые условия пользователей путем создания собственного ЛП, посещающего те порталы, связь пользователя с которыми необходимо подтвердить. Затем, обучается классификатор с учителем с использованием большого количества сетевых свойств портала (цепочки пакетов, их размер и временные интервалы между ними). Используя построенную модель, атакующий классифицирует траффик пользователей в сети.

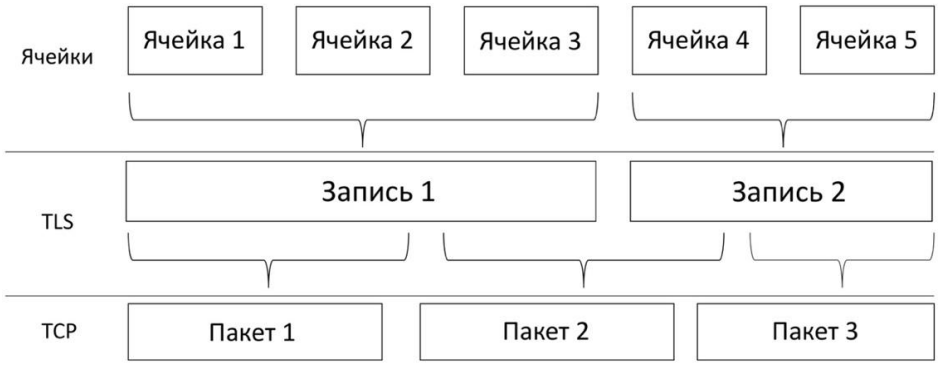

Рисунок 1. Уровни для выделения признаков объектов

Возникает вопрос, на каких уровнях нужно доставать признаки объектов? Достать признаки для них можно на разных уровнях: уровень ячеек Tor, TLS, TCP (см. рисунок 1). На уровне приложения, Tor передает все данные в пакетах фиксированного размера, называемыми ячейками, длины которых 512 байт, а сами ячейки потом превращаются в TLS записи (см. рисунок 3). Примечательно, что много ячеек может быть упаковано в одну TLS запись. И последний – транспортный уровень: TLS записи, как правило, фрагментируются в несколько TCP пакетов, размер которых ограничен MTU. Кроме того, несколько TLS записей могут быть помещены в один TCP пакет. Однако встает очень важный вопрос, какой из слоев содержит больше всего информации, с точки зрения атаки? Большинство исследователей склоняется к тому, что наиболее информативным является уровень ячеек.

Атака начинается со сбора данных. В Tor есть несколько факторов, значительно усложняющих качество собранных данных: процесс конструирования цепей, пропускная способность и загруженность. Например, образцы траффика, собранные через одну цепь, могут отличаться от образцов, собранных через другую. Контент порталов может со временем меняться, что сильно влияет на образцы траффика.

WF атака может быть представлена в виде задачи классификации [19]. Каждый класс может быть группой сайтов, например “Специфический сайт”, “Остальные сайты”. Таким образом, после сбора данных нужно обучить классификатор. После тренировки на полученных образцах, классификатор сможет идентифицировать неизвестные образцы. Наиболее продвинутая модификация WF атаки на Tor позволяет деанонимизировать пользователя с 95% вероятностью в закрытой сети Tor, и с 91% вероятностью в открытой, для узкой группы сайтов.

Существует проблема, связанная с WF атакой. Она называется задачей «Оракула». Так как WF работает с последовательностями пакетов, определить, какая подпоследовательность пакетов траффика из общей последовательности относится к конкретному запросу пользователя в случае Tor достаточно трудно. Для того чтобы упростить исследование данной атаки, исследователи делают два следующих допущения: (1) атакующий располагает таким «Оракулом», (2) жертва загружает странички одна за одной, в одной вкладке. Оракул помогает найти точную подпоследовательность пакетов из общего траффика, относящуюся к конкретному запросу. Любой дополнительный пакет, который попадает на вход классификатору, может существенно понизить точность метода. Вот почему разделение общей последовательности является важной задачей. Другая причина – поведение пользователя в браузере. Большинство людей использует несколько вкладок в процессе просмотра и загрузки веб-страниц. Это обстоятельство тоже усложняет успешное проведение атаки в реальной жизни.

Проблема «Оракул» еще не была решена, однако, Wang предложил решение для Tor, которое работает с одной вкладкой. Он предложил процесс, состоящий из трех шагов, выявляющий правильное разделение последовательностей для одной вкладки между двумя страничками. Wang использовал ячейки Tor вместо пакетов. Первый шаг – разделить последовательность на основе времени. Атакующий делит последовательности,

если временной промежуток между двумя ячейками превысил какое-то пороговое значение. Если же временной промежуток между двумя последовательностями является слишком маленьким, тогда используется метод разделения на основе классификации. Wang использует методы машинного обучения для того, чтобы решить, следует ли разделять ячейки. После разделения, последовательность готова к дальнейшему использованию в последующих алгоритмах. Такой метод на практике позволяет получить довольно высокую точность. Однако предложенное решение не работает с несколькими вкладками в браузере. Работа [22] предлагает временной способ разделения траффика, когда пользователь использует 2 открытые вкладки. Они классифицируют первую страницу с точностью в 75.9%, а вторую с точностью 40.5%.

До января 2016-ого года не было ясно, применимы ли WF атаки к количеству страниц в реальном мире. Панченко в работе [23] проверил атаку с очень большим набором данных, точность же атаки превзошла предыдущий рекорд. Работа доказала, что WF атаки являются серьезной угрозой для анонимности пользователя.

В ответ на успешную попытку применения WF атаки [24], разработчики Tor сделали экспериментальную защиту [25]. Защита содержит три компонента: конвейерная обработка HTTP, размер конвейера и порядок запросов стали задаваться случайно. В работе [19] показали, что защита неэффективна. Более подробную техническую информацию о WF можно найти в работе [26]

Данная атака является самой перспективной с точки зрения программной реализации. Эта атака и будет использоваться в структуре облачной системы деанонимизации и программной реализации.

2.1.2 Timing атаки

Timing атаки являются одними из самых ранних методов деанонимизации. Они наблюдались в самых старых анонимных сетях, включая ранние версии луковой маршрутизации [27]. Данные атаки очень похожи на атаки анализа траффика в сетях, основанных на миксах Чаума [28].

Классическая Timing атака [29] использует метод, при котором атакующий наблюдает временные шаблоны в сетевом потоке и, затем, производит корреляцию выявленных шаблонов с другими, найденными в траффике. Если атакующий имеет возможность наблюдать как пользовательский траффик, так и траффик в конечной точке соединения, то он может установить связь между ними.

Идея основывается на том, что в Tor задержка не может быть большой. Т.е. временной шаблон пакетов данных должен сохраняться при продвижении через цепочку соединения.

Для применения данной атаки, атакующему необходимо иметь коррумпированный узел.

Модель атаки следующая:

- Коррумпированный узел устанавливает соединение с другими узлами Tor, чтобы измерить задержки соединений.
- Коррумпированный узел продолжает производить мониторинг задержек всех установленных соединений, на протяжении определенного временного промежутка.
- Значения задержек используются для расчета транспортной нагрузки тех узлов Tor, с которыми установил соединение злокачественный узел.
- Вычисляются шаблоны траффика, зависящие от транспортной нагрузки.
- Когда атакующий получит шаблоны траффика всех узлов, он может воспроизвести атаку по сценарию атаки анализа траффика.

Для того чтобы сделать атаку более эффективной, необходимо сделать коррумпированный сервер, к которому будет подключаться пользователь. Благодаря этому, нет необходимости просматривать соединение для извлечения шаблона траффика.

Наблюдатель может выбрать шаблон траффика, который легко обнаруживается и посылает свои потоки через коррумпированный сервер. Цель такого улучшения – найти цепь между узлом пользователя и коррумпированным сервером. С таким улучшением анонимность системы падает до уровня простого прокси.

2.1.3 Circuit fingerprinting атака

Данная атака является одной из самых современных, комбинированных атак [30].

Атака спроектирована для компрометирования приватности пользователя, контактирующего со скрытыми службами Tor, что позволяет с высокой степенью точности определить взаимосвязь пользователя со скрытой службой. Как только активность пользователя и скрытой службы идентифицирована, используется WF атака. Предполагается, что атакующий наблюдает траффик между пользователем и сетью Tor.

Траффик, участвующий во взаимодействии со скрытой службой оставляет четкий след, поэтому легко выявить соответствующий шаблон траффика.

Первый шаг атакующего – найти цепи, участвующие во взаимодействии со скрытой службой. Для классификации таких цепей, используются следующие характеристики: длительность активности, количество входящих и исходящих сообщений, последовательность первых десяти сообщений. При этом для классификации цепочки используется дерево решения.

Определив цепочку, атакующий должен получить доступ к входному узлу пользователя, которого он пытается деанонимизировать. Для того чтобы понять, пытается ли пользователь получить доступ к анонимной скрытой службе или к обычному порталу, используется классификатор. Второй классификатор используется, чтобы определить, какую скрытую службу пользователь посетил.

Результаты эксперимента, представленные в работе [30], показали, что данный метод позволяет выявить связь пользователя со скрытым сервисом в 98% случаев при первой атаке, в 99% при второй. Кроме того, в 88% процентах случаев корректно определяется, какую из 50 страниц, за которыми ведется наблюдение, пользователь посетил.

2.2 Активные методы

Активные методы лежат в основе атак, во время которых атакующий пытается изменить данные или каким-либо другим образом вторгнуться в процессы системы.

2.2.1 Timing атаки

Timing атака с использованием браузера [31] позволяет наблюдателю выявить часть пользователей Tor, использующих коррумпированный узел и оставивших открытым окно браузера не менее чем на час. Для реализации атаки необходим коррумпированный сервер, входной и выходной коррумпированные узлы.

Выходной узел, модифицирует HTTP траффик, проходящий через него, вставляя невидимый контейнер iframe, содержащий JavaScript код, в запрашиваемые веб-странички. JavaScript код итеративно контактирует с сервером, посылая уникальный идентификатор и продолжает работать до тех пор, пока человек оставил открытой вкладку с зараженной страничкой в браузере. Полная атака работает следующим образом:

Атакующий разворачивает необходимые ресурсы:

a. вставляет два коррумпированных узла в сеть Tor (входной и выходной),
b. разворачивает веб-сервер, который получает и записывает данные, которые посылает JavaScript код.

Коррумпированный выходной узел модифицирует весь HTTP траффик, вставляя туда невидимый JavaScript код - генератор сигнала, который генерирует уникальный сигнал для каждого клиента Tor.

Веб-браузер клиента запускает JavaScript код, посылая сигнал на сервер. Этот траффик поступает через клиент Tor, но клиент все еще остается анонимным. Каждые 10 минут ЛП строит новую цепь. ЛП выбирает коррумпированный входной узел (случайно).

Атакующий производит анализ траффика для того, чтобы сравнить сигналы на каждой цепи, поступающие через его входной узел с разными сигналами, которые принимает веб сервер. Совпадение сопоставляет клиент Tor с его историей траффика, записанной во время использования коррумпированного выходного узла.

Входному узлу нужно только регистрировать проходящий шаблон траффика на каждой цепи, выходной нужен только для вставок JavaScript кода. Существует модификация атаки, использующая только HTML, описание можно найти в работе [31].

Timing атаки с использованием BGP на уровне АС [32] бывают двух типов.

*Анализ траффика через BGP похищение***.** Для деанонимизации пользователя, наблюдатель может, в первую очередь, применить известные атаки для компрометирования входного узла [18]. Далее, наблюдатель может начать атаку префиксного перехвата против префикса, соответствующему найденному входному узлу. Атака позволяет коррумпированной АС увидеть траффик, предназначающийся входному узлу, за счет поглощения всего траффика входного узла. Поэтому соединение будет активным только какое-то время, а потом оно будет сброшено. Коррумпированная АС может узнать набор клиентов, ассоциированных с входным узлом для продолжительности времени соединения, через инспекцию IP заголовков.

*Анализ траффика через BGP прослушивание***.** Для того, чтобы совершить точную деанонимизацию пользователя через анализ траффика, коррумпированная АС может запустить атаку BGP подслушивания [32]. Эта атака позволяет АС стать промежуточной на пути по направлению к входному узлу, т.е. после перехвата, трафик возвращается обратно к нужной точке назначения. Атака позволяет сохранить соединения, оставляя возможность АС точно деанонимизировать клиента через тайминг анализ.

2.2.2 Атаки анализа траффика

Атака с пометкой ячеек (подтверждающая атака) [34]. заключается в том, что атакующий имеет контроль над входным и выходным узлом пользователя. Атака начинается с коррумпированного входного узла. Входной узел выбирает сообщение в TCP потоке данных и дублирует это сообщение. Исходный IP-адрес сообщения и момент времени дублирования регистрируются. Дублированное сообщение проходит весь путь через цепь и пребывает в выходной узел. Атакующий, управляя выходным узлом, должен засечь дублированное сообщение и записать время, IP адрес назначения сообщения и порт. Тем самым, он подтверждает, что ячейка использует коррумпированные узлы. Таким образом, атакующий устанавливает входные и выходные узлы.

В оригинальной статье про Tor [1] такой класс атак называют “tagging attack” (помечающие атаки). Суть в том, что помечается какое-то сообщение, которое потом ищется в потоке данных.

RAPTOR атака [35] является совершенно новой техникой деанонимизацией пользователя с помощью анализа траффика. В качестве наблюдателя здесь используется АС. RAPTOR атака использует динамические аспекты протокола BGP.

Атака состоит из трех компонент, совместное применение которых дает синергетический эффект. RAPTOR использует асимметричную природу маршрутизации интернета (BGP путь от посылающего к приемнику может отличаться от BGP пути от приемника к посылающему). Эта асимметрия повышает шансы атакующего, обладающего коррумпированной АС, просмотреть хотя бы одно из направлений.

*Первый компонент - ассиметричный анализ траффика***.** Данная форма является новой формой анализа траффика, позволяющей коррумпированной АС деанонимизировать пользователей. Традиционный анализ траффика рассматривает только один сценарий: наблюдатели обозревают траффик от клиента к входному узлу и от

выходного узла к веб-серверу. Как правило, пути через интернет ассиметричны, так что путь от выходного узла к веб-серверу может отличаться от пути от веб-сервера к выходному узлу. Возможен и такой вариант: наблюдатель не имеет возможность просматривать траффик на пути от выходного узла к серверу, но может просматривать траффик TCP подтверждения доставки на пути от сервера к выходному узлу.

Ассиметричный анализ траффика позволяет наблюдателю деанонимизировать пользователей до тех пор, пока наблюдатель может просматривать любое направление траффика на обоих концах соединения. Анализ работает для четырех сценариев: а) траффик данных от клиента к входному узлу и траффик от выходного узла к серверу, б) траффик данных от клиента к входному узлу, траффик TCP подтверждения доставки от сервера к выходному узлу, в) траффик TCP подтверждения доставки от входного узла к клиенту и траффик данных от выходного узла к серверу, г) траффик TCP подтверждения доставки от входного узла к клиенту, траффик TCP подтверждения доставки от сервера к выходному узлу. При анализе исследуются поля TCP заголовков в наблюдаемом траффике для выявления номера TCP последовательности и номер TCP подтверждения доставки. Далее, вычисляется корреляция между этими полями.

Второй компонент – анализ натуральных перебоев. Путь между клиентом и входным узлом изменяется во времени из-за физической топологии и политик АС. Такие изменения увеличивают со временем вероятность попадания пользовательских цепей в коррумпированные АС.

Третий компонент – атаки BGP похищения и BGP прослушивания. Подробно данные атаки были описаны в работе [33]. Атака BGP прослушивания позволяет коррумпированной АС стать на пути перед входным узлом, т.е. после перехвата, траффик будет возвращаться обратно через входной узел. Такой перехват сохранит соединение и позволит АС произвести ассиметричный анализ траффика. Кроме того, такая атака позволяет коррумпированной АС деанонимизировать пользователя посещающего какой-то конкретный сайт. АС видит траффик на клиенте и может запустить BGP прослушивания атаку против выходного узла.

Результаты экспериментов, приведенные в работе [35] показывают, что атака успешна в 90% случаев.

2.2.3 DoS атаки

Обычные DoS атаки не требуют глубокого знания Tor и могут быть осуществлены с использованием простых, известных техник [36]. Кроме того, детальный анализ обычных DoS атак на Tor можно найти в работе [37]. В 2014 году вышла работа [38], посвященная атаке, позволяющей израсходовать всю доступную память узла, однако, из-за кооперации авторов с разработчиками Tor, атака более не работоспособна, поэтому в настоящей статье не рассматривается.

Атака packet spinning [39] заставляет пользователя выбирать коррумпированные узлы, через вывод из строя легитимных узлов. Атакующий строит циклические цепи через сеть Tor и посылает большой объем данных через этот путь, чтобы легитимные узлы были максимально загружены. Атакующий запускает другой набор коррумпированных узлов, которые когда-нибудь будут выбраны пользователями, потому что атакующий перегрузил все легитимные. Атака будет успешной, только если инициатор выбирает только коррумпированные узлы для своих цепей, в результате чего деанонимизация становится тривиальной.

Атака перегрузки с использованием длинных путей [40] основана на следующих свойствах Tor: а) маршрутизаторы Tor не вставляют искусственные задержки между запросами б) IP-адреса всех узлов Tor публично известны и доступны.

Атака предполагает, что атакующий контролирует выходной узел. Узел используется для вставки JavaScript кода в HTML ответ на запрос. Код заставляет браузер посылать HTTP запросы каждую секунду и в ответ на каждый запрос, выходной узел

посылает пустой ответ, который отвергается браузером. Атакующий записывает интервалы времени периодических запросов, производимых браузером. Т.к. запросы маленькие, возникает задержка, равная примерно разнице во времени доставки сигнала кодом.

У владельца коррумпированного выходного узла теперь стоит задача вызвать перегрузку узлов, подозреваемых в участии в цепи. Предполагаем, что все узлы являются подозреваемыми и в самом простом случае, атакующий будет итеративно, через все узлы проверять является ли данный узел входным узлом цепи.

Для каждого узла X, атакующий конструирует длинную цепь, которая с повторениями включает X в цепь. Из-за того, что Tor сбрасывает цепь при попытке расширения цепи через предыдущий узел, нужно использовать два или более промежуточных узла для замыкания цепи на X.

Как только цепь становится достаточно длинной (авторы статьи [40] предлагают 24 узла), атакующий использует цепь для передачи данных. Цепь длины m позволит атакующему с величиной пропускной способности p уменьшить пропускную способность сети Tor на величину m * p. Узлу X придется участвовать в m/3 дополнительных цепях, что позволит атакующему встраивать большие задержки в конкретный узел.

Если узел X не относится к компрометирующей цепи, измеримые задержки не вызовут заметных изменений во время выполнения атаки. Если X – входной узел, атакующий будет просматривать внедренный шаблон задержки и выявит искомую цель.

CellFlood DoS атака [41] вместо создания большого количества запросов, на обработку которых требуется мало вычислительных ресурсов, как в обычной DoS атаке, атака использует несколько тяжелых запросов создания цепи, которые быстро генерируются атакующим с минимальным количеством ресурсов, однако будут требовать большое количество вычислительных ресурсов от узла, на который идет атака.

Из-за криптографических операций, совершаемых во время расширения цепи, обработка CREATE сообщения занимает в 4 раза больше времени, чем его генерация [42], из-за криптографических операций, основанных на паре открытый-закрытый ключа, совершаемых во время расширения цепи. Это может быть использования всех доступных ресурсов атакуемого узла.

Благодаря архитектуре Tor, узел, на который посылается огромное количество CREATE сообщений, не теряет возможность пересылать сообщения типа RELAY_DATA. Узел, получающий CREATE сообщения быстрее, чем его процессор может обработать, отвечает на них, посылая DESTROY сообщения в ответ. Следовательно, узел, находящийся под атакой, будет отклонять запросы от легитимных узлов. Если атака выполняется стратегически, возможна перегрузка большой части сети и выявление цепи, проходящей через конкретные узлы.

Если атакующий заинтересован в том, чтобы исключить узел или набор узлов из сети Tor, ему выгоднее использовать поток сообщений CREATE, чем обычную и более дорогую DoS атаку.

Описание некоторых методов не вошло в настоящую статью из-за большой степени схожести с уже рассмотренными. Другие методы в рамках тематики Tor описывать не имеет особого смысла. К числу таких атака относятся Sybil атаки [43], которые в рамках Tor представляют собой ни что иное, как обладание большим количеством коррумпированных узлов, составляющих ощутимую долю узлов Tor. В работе [44] описана комбинированная атака анализа траффика, использующаяся совместно с Sybil атакой. В работах [45, 46] можно найти атаки анализа траффика с использованием исторических данных о TCP. Ресурс [47] содержит исчерпывающую библиографическую подборку по атакам и защите анонимных систем.

2.3 Классификация атак

Детальный обзор, приведенный выше, позволяет с уверенностью вывести итоговую классификацию атак (см. таблицу 1) на приватность пользователя в сети Tor.

№	Ресурсы	Атаки
1	Коррумпированный входной узел	• Website fingerprinting атака
2	Коррумпированные входной и выходной узлы	• Анализ трафика • Тайминг атаки • Circuit fingerprinting атака • Атака с пометкой ячеек (тэггинг атака)
3	Коррумпированный выходной узел	• Сниффинг перехваченного трафика (если пользователь не использует протокол https)
4	Коррумпированный входной, выходной узел и внешний сервер	• Тайминг атака с использованием браузера и внедрения JavaScript кода • Атака анализа траффика с использованием браузера и внедрения JavaScript кода
5	Автономная система	• Атаки BGP похищения • Атаки BGP прослушивания • RAPTOR атака
6	Большое количество любых коррумпированных узлов	• Packet spinning атака • Атака перегрузки с использованием длинных путей • CellFlood DoS атака

Таблица 1. Классификация атак по используемым ресурсам

3 Структура интеллектуальной системы для деанонимизации пользователей с использованием облака

Ключевыми компонентами системы являются компоненты для сбора и анализа данных [50].

Бурное развитие облачных вычислений и методов машинного обучения позволяет спроектировать недорогую систему для решения задачи деанонимизации пользователя. Структура предлагаемой интеллектуальной системы представлена на рисунке 2.

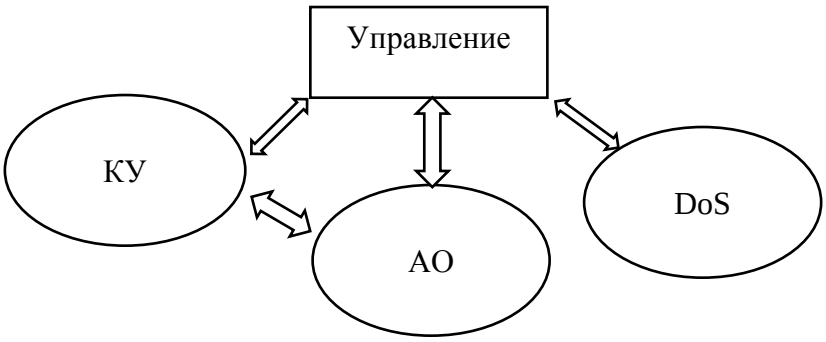

Рисунок 2. Структура интеллектуальной системы. Управление – блок, контролирующий все компоненты системы. КУ – коррумпированные узлы (под контролем атакующего). АО – аналитическое облако. DoS – узлы, предназначенные для DoS атаки.

Блок «КУ» – компонент системы, состоящий из набора коррумпированных узлов. Это такие узлы, на которых атакующий может просматривать траффик. Данные узлы будут использоваться для перехвата пользовательского траффика и его передачи в аналитическое облако.

Блок «АО» – аналитическое облако, включает в себя следующие компоненты: база данных, классификатор и паук. База данных используется для хранения накопленного траффика для обучения и копии последних версий ресурсов из списка атакующего. Классификатор – блок машинного обучения, отвечающий за определение ресурса по траффику, отправленному к нему на вход. Паук же занимается скальпированием ресурсов из списка и обновлением траффика в базе, он постоянно проверяет актуальность собранных данных.

Блок «DoS» содержит в себе узлы Tor, которые предназначены для DoS атаки на другие узлы внутри сети (случай, если нужно заставить систему сменить входной узел для конкретного пользователя). Кроме того, данный блок в любой момент может быть использован в качестве КУ, поскольку узлы могут использоваться в обычном режиме.

Блок «Управление» предназначен для синхронизации и управления всеми компонентами системы.

Описанная интеллектуальная система позволяет применять метод деанонимизации пользователей на практике, в полностью автономном режиме. От оператора требуется лишь заниматься пополнением списка ресурсов и исправлением ошибок, о которых сигнализирует блок «Управление». Блок «АО» проектируется с учетом необходимости обработки больших объемов данных. Одним из возможных решений является использование платформы Hadoop.

Приведем алгоритм использования системы. Первый шаг – инициализация, оператор загружает в блок «Управление» список ресурсов, затем блок «АО», получив команду, передает управление блокам «КУ» и «DoS», которые собирают данные и записывают их в базу. После того, как сбор завершен – классификатор обучается. Сразу же после завершения обучения система готова к использованию. Блок «КУ» перехватывает траффик и отправляет его в классификатор для дальнейшего анализа. Паук же постоянно мониторит ресурсы из списка для обновления базы траффика. Если база траффика обновляется, классификатор должен переобучиться, поскольку от актуальности данных сильно зависит точность классификации.

Следующим этапом реализации системы является усовершенствование «Оракула» и проведение исследований, направленных на построение более точной модели классификатора.

4 Схема «дешевой» системы деанонимизации

Для решения задачи деанонимизации в предлагаемой системе реализовано два модуля. Первый модуль используется для сбора трафика через Tor. Второй модуль используется для применения техник машинного обучения.

4.1 Сбор данных

Так как трафик может собираться как на коррумпированном входном узле, так и на клиентской стороне, можно для сбора данных использовать локальную машину или входной узел. Для корректного функционирования программы, на используемой машине устанавливается следующее программное обеспечение:

- Tor – свободное программное обеспечение для осуществления анонимных коммуникаций
- Torsocks – свободное программное обеспечение, позволяющие использовать любое приложение через сеть Tor.
- Wget – программа для получения контента от веб-серверов.

- Tshark – свободный анализатор пакетов. Используется при анализе сетевых взаимодействий.
- Mozilla Firefox или Tor Browser – открытый веб-браузер.

Любая программа в списке может быть заменена на специализированную библиотеку. Самое просто решение – использовать предложенное выше программное обеспечение. Нам необходим полный контроль над конструированием цепей Tor, поскольку нужно посещать ресурсы через собственный входной узел. Для этого используется библиотека Stem для языка программирования Python, которая свободно распространяется через интернет. Stem попросту является питоновским контроллером для Tor.

Stem используется для конструирования цепей Tor через собственный коррумпированный входной узел. Без конструирования цепей Tor через собственный коррумпированный входной узел точность классификатора может стать очень низкой. Это связано с тем, что на входных узлах могут быть различные версии Tor.

Вместо использования Stem можно модифицировать конфигурации Tor на локальной машине, использующейся для посещения Web-сайтов. При этом важно использовать один и тот же входной узел как для сбора данных, так и для деанонимизации пользователей.

Tshark используется в качестве основного пакета как для анализа пакетов траффика, так и для извлечения TLS записей из данных. Tshark может быть заменен любой библиотекой, поддерживающей перехват TCP пакетов.

Существует два способа автоматизации процесса сбора данных: использование wget через torsocks, или Mozilla Firefox. В случае использования wget, атакующий просто запускает загрузку страницы из командной строки. При использовании Mozilla Firefox автоматизация работы с браузером достигается двумя путями: 1) запуск из командной строки и ожидание окончания загрузки страницы, 2) использование Selenium Webdriver.

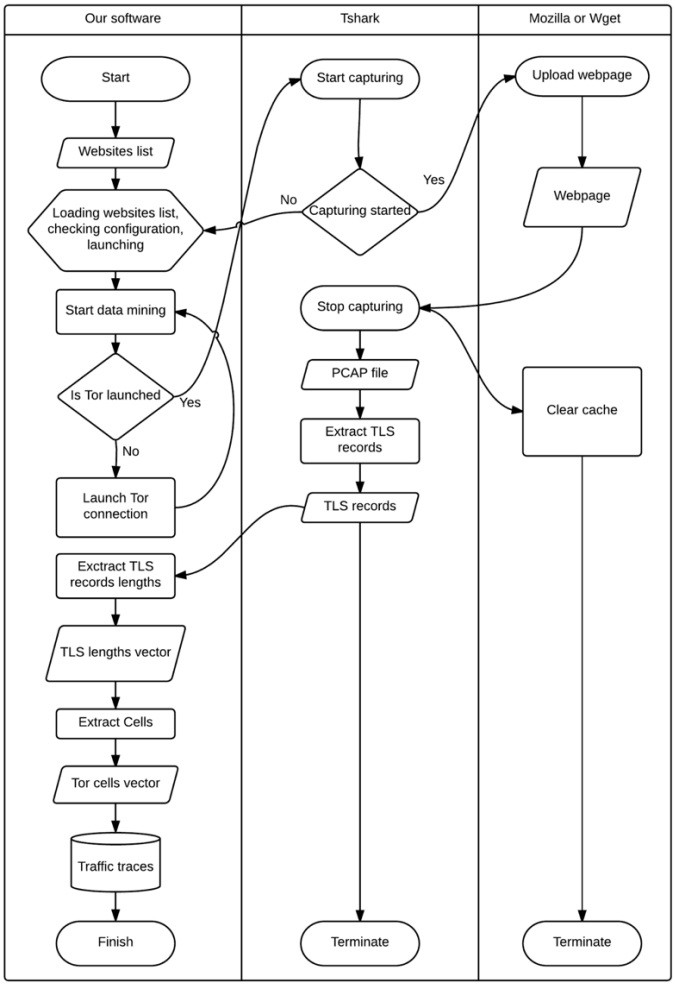

Рисунок 3. Процесс извлечения данных

4.2 Извлечение признаков

Извлечение признаков осуществляется на уровне ячеек Tor. Сначала извлекаются TLS записи из TCP пакетов с помощью программы Tshark. Далее, длина записи, умножается на -1, если она исходящая.

Получившийся массив длин TLS записей должен быть трансформирован в ячейки Tor. Каждое число массива делится на 512 и добавляется в вектор ячеек количество -1 или 1, равное результату деления. Например, если элемент массива равен 2048, результирующий вектор ячеек будет [1,1,1,1]. В случае, если элемент массива равен -3072 результирующий вектор ячеек будет [-1,-1,-1,-1,-1].

Такие векторы ячеек используются в качестве признаков, а названия веб-страниц используются в качестве лейблов. Заметим, что вообще говоря, такие массивы имеют разные длины. Для упрощения процесса, в конец входных векторов добавляются нули, поскольку большинство алгоритмов машинного обучения требуют того, чтобы входные векторы имели одинаковые длины.

Процесс извлечения данных для использования в классификаторе представлен на рисунке 3.

4.3 Модуль машинного обучения

Для машинного обучения используется пакет sklearn для языка программирования Python. На основе полученных векторов ячеек строится модель машинного обучения, которая используется для классификации нового траффика.

5 Экспериментальный комплекс

Представленная схема системы деанонимизации реализована на языках Java и Python. UML диаграмма классов программы экспериментального решения представлена на рисунке 4. Цель эксперимента – показать, что возможно деанонимизировать небольшую долю пользователей Tor в реальном мире, даже если не используются самые мощные методы деанонимизации.

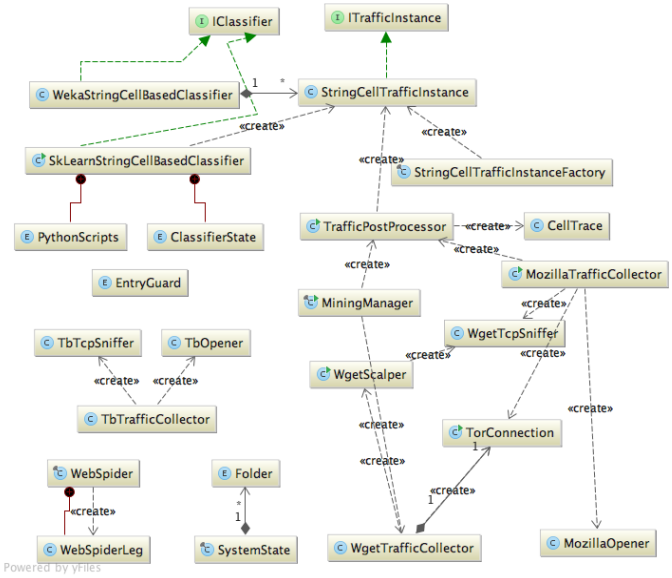

Рисунок 4. UML диаграмма классов программы

5.1 Экспериментальная среда

Представим следующую ситуацию: группа террористов пытается получить доступ к нелегальному контенту из маленькой комнаты в общежитии. Список ресурсов для мониторинга представлен компанией Group-IB. В нашем эксперименте было три пользователя, играющих роль террористов. Каждый из них посещал ресурсы из списка, руководствуясь следующими правилами: запросы из браузера делались только через одну вкладку, а время для чтения странички было как минимум 5 секунд. Такое предположение является реалистичным [49]. Введение таких ограничений позволяет нам упростить процесс разделения последовательностей пакетов и извлечения трафика.

5.2 Сбор данных

Перед деанонимизацией пользователей, был сделан подготовительный шаг, в рамках которого собрано 80 примеров траффика из списка ресурсов, предоставленного компанией Group-IB. Такое маленькое количество примеров траффика является достаточным, потому что более крупные наборы данных не увеличивают точность классификатора при том же количестве сайтов. В эксперименте использовано 7 ресурсов, относящихся к наркотикам, оружию и экстремистскому контенту.

Пользователи повторяли процесс чтения и загрузки веб-страницы 5 раз для каждой страницы из листа. После этого, собранные последовательности траффика загружались и проходили предварительную обработку. Было использовано разделение пакетов, основанное на временных промежутках, как описано в работе [49]. После этого шага, данные стали пригодными для использования в классификаторе.

5.3 Модель машинного обучения

Машины опорных векторов (SVM) являются моделями обучения с учителем, с ассоциированными алгоритмами обучения, которые используются для классификации или регрессии. Модель SVM представляет примеры как точки в пространстве, сгруппированные так, что примеры из разных категорий разделяются границей. Новые примеры затем классифицируются в зависимости от попадания в конкретную область, ограниченную границей.

В работе использовался NuSVC алгоритм со стандартными гиперпараметрами из библиотеки sklearn. NuSVC является Nu-Support vector классификаций, основанной на машине опорных векторов. Этот алгоритм использует параметр для контролирования количества опорных векторов, где параметр является верхней границей доли тренировочных ошибок и нижней границей доли опорных векторов.

5.4 Результаты эксперимента

В качестве метрик были использованы следующие величины:

- True positives (tp) – равно количеству попаданий
- False positives (fp) – равно количеству корректных отвержений
- False negatives (fn) – ошибки второго рода
- Precision – tp / (tp + fp). Интуитивно – способность классификатора избежать определения негативного примера с позитивным лейблом.
- Recall – tp / (tp + fn). Интуитивно – способность классификатора найти все положительные примеры (наилучшее – 1, наихудшее – 0)
- F1-score – взвешенное среднее precision и recall (лучшее значение – 1, наихудшее – 0) = 2 * (precision * recall) / (precision + recall).

Классификатор был проскорен с использованием стандартной функции из библиотеки sklearn. По этическим причинам, описанным в документе по этическому исследованию Tor, мы анонимизировали сайты, используемые в эксперименте. Подробную оценку классификатора смотреть в таблице 2.

Website	Precision	Recall	F1-score
Site_1	1.00	1.00	1.00
Site_2	0.80	0.80	0.80
Site_3	0.80	0.80	0.80
Site_4	0.50	0.40	0.44
Site_5	1.00	1.00	1.00
Site_6	0.38	0.60	0.46
Site_7	0.67	0.40	0.50
Avg/total	0.73	0.71	0.72

Таблица 2. Результаты скоринга классификатора

В целом, точность классификации следующая: 0.714

Результаты не являются впечатляющими, в сравнении с самыми современными техниками деанонимизации [51], но они показывают, что мы можем деанонимизировать пользователей с помощью достаточно простых программ и при этом достигать приличной точности.

6 Выводы

Были изучены самые распространенные анонимные сети и методы деанонимизации пользователей в этих сетях, была построена классификация методов деанонимизации на основе ресурсов, использующихся атаками, была построена структура интеллектуальной системы деанонимизации пользователей.

Была построена программа для деанонимизации пользователей Tor, был проведен эксперимент, подтверждающий деанонимизацию пользователей Tor и позволяющий определить точность деанонимизации.

Показано, что атакующий без супер-современных методов машинного обучения может использовать website fingerprinting атаку. Если атакующий имеет достаточно опыта и технической компетентности, то он может построить такую систему и использовать ее для деанонимизации пользователей. Более того, предложенное решение будет работать лучше всего, если атакующий будет перехватывать Wi-Fi траффик или какую-либо другую локальную сеть, потому что для него будет проще найти трафик, относящийся к Tor и собрать его.

7 Дальнейшая работа

Проблему «Оракула» предполагается решить на основе облачного приложения, использующего рекуррентные нейронные сетей и исследовать применимость этого приложения для WF атаки.

Основная цель решения проблемы «Оракула» - найти достаточно точный разделяющий алгоритм, который позволит использовать WF атаки даже при интернет-серфинге с несколькими вкладками.

Термины и сокращения

BGP (англ. Border Gateway Protocol) — это основной протокол динамической маршрутизации, который используется в Интернете. Маршрутизаторы, использующие протокол BGP, обмениваются информацией о доступности сетей.
BGP Hijack (BGP похищение) – незаконное поглощение группы IP адресов.
BGP Interception (BGP прослушивание) – незаконное прослушивание группы IP адресов.
CREATE сообщение – сообщение управляющего типа, посылается для установления новой цепи через сеть.
DESTROY сообщение – сообщение управляющего типа, посылается для разрыва существующей цепи.
DoS (англ. Denial of Service) – хакерская атака на вычислительную систему с целью довести ее до отказа, то есть, создание таких условий, при которых легальные пользователи системы не могут получить доступ к предоставляемым системным ресурсам (серверам), либо этот доступ затруднен.
HTTP (англ. HyperText Transfer Protocol) - протокол прикладного уровня передачи данных (изначально — в виде гипертекстовых документов в формате HTML, в настоящий момент используется для передачи произвольных данных).
ID - уникальный идентификатор.
Iframe – html контейнер, содержание которого игнорируется браузерами, не поддерживающими данный тег.
IP-адрес – (англ. Internet Protocol Address) уникальный сетевой адрес узла в компьютерной сети, построенной по протоколу IP.
ISP (англ. Internet Service Provider) – организация, предоставляющая пользователям доступ к сети Интернет и связанные с этим услуги.

JavaScript – прототипно-ориентированный сценарный язык программирования.
RAPTOR (англ. Routing Attacks on Privacy in Tor) – маршрутизационные атаки на приватность в Tor.
RELAY_DATA сообщение – сообщение управляющего типа, посылается для выполнения команды к ретранслированию данных.
Sybil атака – атака на безопасность компьютерной системы, где репутационная система подрывается с помощью нелегитимных сущностей в пиринговых сетях.
Tagging attack – атака на приватность в сети Tor, при которой помечается какая-то ячейка и, затем, происходит ее поиск на другом конце соединения.
TCP – (англ. Transmission Control Protocol), протокол управления передачей) — один из основных протоколов передачи данных Интернета, предназначенный для управления передачей данных в сетях и подсетях TCP/IP.
Timing атака – атака по сторонним канал, в которой атакующий пытается скомпрометировать систему с помощью анализа времени, затрачиваемого на исполнение операций
Tor (англ. The Onion Router) – анонимная сеть и открытое программное обеспечение, позволяющее сохранять пользователям свою анонимность.
WF (website fingerprinting) – класс пассивных атак, позволяющий определить посещенный пользователем портал или скрытую службу.
Автономная система (АС) – это система IP-сетей и маршрутизаторов, управляемых одним или несколькими операторами, имеющими единую политику маршрутизации с интернетом.
Конвейерная обработка HTTP – технология, которая позволяет передавать на сервер сразу несколько запросов в одном соединении, не ожидая соответствующих ответов.
Коррумпированная автономная система – автономная система, контролируемая наблюдателем.
Коррумпированный сервер – конечная точка назначения, контролируемая наблюдателем.
Коррумпированный узел – узел, траффик которого может модифицировать и просматривать атакующий.
ЛП – луковый прокси.
Мета-данные сети – субканальная информация об используемых данных.
Миксы Чаума - устройства для передачи и хранения, принимающее какое-то количество сообщений фиксированной длины от нескольких источников, совершающее криптографическую трансформацию сообщений и, затем, передающее сообщение к следующему пункту назначения в случайном порядке.
Скрытая служба – портал/сайт, доступный только внутри сети Tor.
УМПС - узел с минимальной пропускной способностью
ЦП - целевой поток

Список литературы

1. Dingledine R., Mathewson N., Syverson P. Tor: The Second-Generation Onion Router [Electronic resource] // Tor project [Official website]. URL: https://svn.torproject.org/svn/projects/design-paper/tor-design.pdf (accessed: 8.09.2015).
2. TorMetrics [Electronic resource] // Tor project [Official website]. URL: https://metrics.torproject.org (accessed: 28.09.2015).
3. The Russian government hired people to hack the Tor browser, but they failed and now they're quitting [Electronic resource] // Meduza [Official website]. URL: https://meduza.io/en/news/2015/09/09/the-russian-government-hired-people-hack-the-tor-browser-but-they-failed-and-now-they-re-quitting (accessed: 28.09.2015).

4. The NSA's Been Trying to Hack into Tor's Anonymous Internet For Years [Electronic resource] // Gizmodo [Official website]. URL: http://gizmodo.com/the-nsas-been-trying-to-hack-into-tors-anonymous-inte-1441153819 (accessed: 28.09.2015).
5. Закупка №0373100088714000008 [Электронный ресурс] // Государственные закупки [Официальный сайт]. URL: http://zakupki.gov.ru/epz/order/notice/zkk44/view/common-info.html?regNumber=0373100088714000008 (дата обращения: 2.10.2015).
6. The Rise & Fall of Silk Road [Electronic resource] // Wired [Official website]. URL: http://www.wired.com/2015/04/silk-road-1/ (accessed: 28.09.2015).
7. Danezis G. Statistical disclosure attacks [Electronic resource] // Freehaven [Official website]. URL: http://freehaven.net/anonbib/cache/statistical-disclosure.pdf (accessed: 1.10.2015).
8. Kedogan D., Agrawal D., Penz S. Limits Of Anonymity in Open Environment [Electronic resource] // Springer Link [Official website]. URL: http://link.springer.com/chapter/10.1007%2F3-540-36415-3_4 (accessed: 1.10.2015).
9. Danezis G. Statistical disclosure attacks [Electronic resource] // Freehaven [Official website]. URL: http://freehaven.net/anonbib/cache/statistical-disclosure.pdf (accessed: 1.10.2015).
10. Mathewson N., Dingledine R. Practical traffic analysis: Extending and resisting statistical disclosure [Electronic resource] // Freehaven [Official website]. URL: http://freehaven.net/doc/e2e-traffic/e2e-traffic.pdf (accessed: 1.10.2015).
11. Agrawal D., Kesdogan D., Penz S. Probabilistic Treatment of Mixes to Hamper Traffic Analysis [Electronic resource] // Freehaven [Official website]. URL: http://freehaven.net/anonbib/cache/agrawal03.pdf (accessed: 1.10.2015).
12. Danezis G. The Traffic Analysis of Continuous-Time Mixes [Electronic resource] // Freehaven [Official website]. URL: http://freehaven.net/anonbib/cache/danezis:pet2004.pdf (accessed: 1.10.2015).
13. Zhu Y., Fu X., Graham B., Bettati R., Zhao W. On flow correlation attacks and countermeasures in mix networks [Electronic resource] // Freehaven [Official website]. URL: http://freehaven.net/anonbib/cache/flow-correlation04.pdf (accessed: 1.10.2015).
14. Levine B.N., Reiter M.K., Wang C., Wright M.K. Timing attacks in low-latency mix-based systems [Electronic resource] // Freehaven [Official website]. URL: http://freehaven.net/anonbib/cache/flow-correlation04.pdf (accessed: 1.10.2015).
15. Johnson A., Wacek C., Jansen R., Sherr M., Syverson P. Users Get Routed: Traffic Correlation on Tor by Realistic Adversaries [Electronic resource] // Aaron Michael Johnson [Official website]. URL: http://www.ohmygodel.com/publications/usersrouted-ccs13.pdf (accessed: 2.10.2015).
16. Murdoch S.J., Danezis G. Low-Cost Traffic Analysis of Tor [Electronic resource] // UCL-CS [Official website]. URL: http://sec.cs.ucl.ac.uk/users/smurdoch/papers/oakland05torta.pdf (accessed: 1.10.2015).
17. Bauer K., McCoy D., Grunwald D., Kohno T., Sicker D. Low-Resource Routing Attacks Against Tor [Electronic resource] // University of Washington [Official website]. URL: https://homes.cs.washington.edu/~yoshi/papers/Tor/wpes25-bauer.pdf (accessed: 1.10.2015).
18. Mittal P., Khurshid A., Juen J., Caesar M., Borisov M. Stealthy Traffic Analysis of Low-Latency Anonymous Communication Using Throughput Fingerprinting [Electronic resource] // Princeton University [Official website]. URL: http://www.princeton.edu/~pmittal/publications/throughput-fingerprinting-ccs11.pdf (accessed: 1.10.2015).
19. Cai X., Zhang X., Joshi B., Johnson R. Touching from a Distance: Website Fingerprinting Attacks and Defenses [Electronic resource] // Stony Brook University

[Official website]. URL: http://www3.cs.stonybrook.edu/~xcai/fp.pdf (accessed: 1.10.2015).
20. Wang T., Cai X., Nithyanand R., Johnson R., Goldberg I. Effective Attacks and Provable Defenses for Website Fingerprinting [Electronic resource] // Centre for Applied Cryptographic Research The University of Waterloo [Official website]. URL: http://cacr.uwaterloo.ca/techreports/2014/cacr2014-05.pdf (accessed: 1.10.2015).
21. Cai X., Nithyanand R., Wang T., Johnson R., Goldberg I. A Systematic Approach to Developing and Evaluating Website Fingerprinting Defences [Electronic resource] // Freehaven [Official website]. URL: http://freehaven.net/anonbib/cache/ccs2014-fingerprinting.pdf (accessed: 1.10.2015).
22. X.Gu, M.Yang, J.Luo, "A Novel Website Fingerprinting Attack Against Multi-Tab Browsing Behavior," in Computer Supported Cooperative Work in Design (CSWD), 2015 [Electronic resource] // IEEE Explore [Official website]. URL: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=7230964&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D7230964 (accessed: 3.04.2016)
23. Panchenko, F. Lanze, A. Zinnen, M. Henze, J. Pannekamp, K. Wehrle, T. Engel, "Website Fingerprinting at Internet Scale," [Electronic resource] // Comsys [Official website]. URL: https://www.comsys.rwth-aachen.de/fileadmin/papers/2016/2016-panchenko-ndss-fingerprinting.pdf (accessed: 1.04.2016).
24. Panchenko A., Niessen L., Zinnen A., Engel A. Website fingerprinting in onion routing based anonymization networks [Electronic resource] // Freehaven [Official website]. URL: http://freehaven.net/anonbib/cache/wpes11-panchenko.pdf (accessed: 1.10.2015).
25. Experimental Defense for Website Traffic Fingerprinting [Electronic resource] // Tor project [Official website]. URL: https://blog.torproject.org/blog/experimental-defense-website-traffic-fingerprinting (accessed: 1.10.2015).
26. Wang. T., Goldberg I. Improved Website Fingerprinting on Tor [Electronic resource] // Freehaven [Official website]. URL: http://freehaven.net/anonbib/cache/wpes13-fingerprinting.pdf (accessed: 1.10.2015).
27. Goldschlag D., Reed M., Syverson P. Onion Routing for Anonymous and Private Internet Connections. January 28, 1999 [Electronic resource] // Onion Routing [Official website]. URL: http://www.onion-router.net/Publications/CACM-1999.pdf (accessed: 8.09.2015).
28. Chaum D. Untraceable Electronic Mail, Return Addressed, and Digital Pseudonyms [Electronic resource] // Free Haven [Official website]. URL: http://www.freehaven.net/anonbib/cache/chaum-mix.pdf (accessed: 28.09.2015).
29. Wiangsripanawan R., Susilo W., Safavi-Naini R. Design principles for low latency anonymous network systems secure against timing attacks [Electronic resource] // ACM Digital Library [Official website]. URL: http://dl.acm.org/citation.cfm?id=1274553 (accessed: 28.09.2015)
30. Kwon A., AlSabah M., Lazar D., Dacler M., Devadas S. Circuit Fingerprinting Attacks: Passive Deanonymization of Tor Hidden Services [Electronic resource] // USENIX [Official website]. URL: https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/kwon (accessed: 2.10.2015).
31. Abbot T., Lai K., Lieberman M., Price E. Browser-Based Attacks on Tor [Electronic resource] // Privacy Enhancing Technologies [Official website]. URL: https://www.petsymposium.org/2007/papers/PET2007_preproc_Browser_based.pdf (accessed: 1.10.2015).
32. Vanbever L., Li O., Rexford J., Mittal P. Anonymity on QuickSand: Using BGP to Compromise Tor [Electronic resource] // ACM SIGCOMM [Official website]. URL: http://conferences.sigcomm.org/hotnets/2014/papers/hotnets-XIII-final80.pdf (accessed: 1.10.2015).

33. Ballani H., Francis P., Zhang X. A study of prefix hijacking and interception in the Internet [Electronic resource] // ACM Digital Library [Official website]. URL: http://dl.acm.org/citation.cfm?id=1282411 (accessed: 1.10.2015).
34. Pries R., Yu W., Fu X., Zhao W. A New Replay Attack Against Anonymous Communication Networks [Electronic resource] // IEEE Xplore [Official website]. URL: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4533341&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D4533341 (accessed: 2.10.2015).
35. Sun Y., Edmundson A., Vanbever L., Li O., Rexford J., Chiang M., Mittal P. RAPTOR: Routing Attack on Privacy in Tor [Electronic resource] // USENIX [Official website]. URL: https://www.usenix.org/system/files/conference/usenixsecurity15/sec15-paper-sun.pdf (accessed: 5.10.2015).
36. Low orbit ion cannon. [Electronic resource] // Wikipedia [Official website]. URL: http://en.wikipedia.org/wiki/Low_Orbit_Ion_Cannon (accessed: 1.10.2015).
37. Borisov N., Mittal P., Danezis G., Tabriz P. Denial of Service or Denial of Security? How Attacks on Reliability can Compromise Anonymity [Electronic resource] // Princeton University [Official website]. URL: http://www.princeton.edu/~pmittal/publications/dos-ccs07.pdf (accessed: 1.10.2015).
38. Jansen R., Tschorsch F., Johnson A., Scheuermann B. The Sniper Attack: Anonymously Deanonymizing and Disabling the Tor Network [Electronic resource] // Center for High Assurance Computer Systems [Official website]. URL: http://www.nrl.navy.mil/itd/chacs/biblio/sniper-attack-anonymously-deanonymizing-and-disabling-tor-network (accessed: 1.10.2015).
39. Pappas V., Athanasopoulos E., Ioannidis S., Markatos E.P. Compromising Anonymity Using Packet Spinning [Electronic resource] // FORTH-ICS [Official website]. URL: http://www.ics.forth.gr/dcs/Activities/papers/torspin.isc08.pdf (accessed: 2.10.2015).
40. Evans N.S., Dingledine R., Grothoff C. A practical congestion attack on tor using long paths [Electronic resource] // ACM Digital Library [Official website]. URL: http://dl.acm.org/citation.cfm?id=1855771 (accessed: 2.10.2015).
41. Barbera M.V., Kemerlis V.P., Pappas V., Keromytis A.D. CellFlood: Attacking Tor Onion Routers on the Cheap [Electronic resource] // Springer Link [Official website]. URL: http://link.springer.com/chapter/10.1007%2F978-3-642-40203-6_37 (accessed: 1.10.2015).
42. How fast is the RSA algorithm [Electronic resource] // RSA Laboratories [Official website]. URL: http://www.rsa.com/rsalabs/node.asp?id=2212 (accessed: 1.10.2015).
43. Douceur J.R. The Sybil Attack [Electronic resource] // Freehaven [Official website]. URL: http://freehaven.net/anonbib/cache/sybil.pdf (accessed: 2.10.2015).
44. Chakravarty S., Barbera M.V., Portokalidis G., Polychronakis M., Keromytis A.D. On the Effectiveness of Traffic Analysis Against Anonymity Nerworks Using Flow Records [Electronic resource] // Columbia University [Official website]. URL: https://mice.cs.columbia.edu/getTechreport.php?techreportID=1545 (accessed: 2.10.2015).
45. Gilad Y., Herzberg A. Spying in the Dark: TCP and Tor Traffic Analysis [Electronic resource] // Freenet [Official website]. URL: http://freehaven.net/anonbib/cache/tcp-tor-pets12.pdf (accessed: 2.10.2015).
46. Chakravarty S., Barbera M.V., Portokalidis G., Polychronakis M., Keromytis A.D. On the Effectiveness of Traffic Analysis Against Anonymity Nerworks Using Flow Records [Electronic resource] // Columbia University [Official website]. URL: https://mice.cs.columbia.edu/getTechreport.php?techreportID=1545 (accessed: 2.10.2015).
47. Selected Papers in Anonimity [Electronic resource] // Freehaven [Official website]. URL: http://freehaven.net/anonbib/ (accessed: 2.10.2015).

48. Авдошин С.М., Лазаренко А.В. Технология анонимных сетей. Журнал «Информационные технологии», том 22, №4, стр. 284-291.
49. T. Wang, “Website Fingerprinting: Attacks and Defenses”, PhD Thesis, 2015 [Электронный ресурс] // Университет Ватерлоо [Официальный сайт]. URL: https://uwspace.uwaterloo.ca/bitstream/handle/10012/10123/Wang_Tao.pdf?sequence=3 (дата обращения: 3.04.2016).
50. А.В. Лазаренко. Структура интеллектуальной системы деанонимизации пользователей сети Tor. Материалы межвузовской научно-технической конференции студентов, аспирантов и молодых специалистов им. Е.В. Арменского, 2016 год, стр. 81-83.
51. Авдошин С.М., Лазаренко А.В. Методы деанонимизации пользователей Tor. Журнал «Информационные технологии», том 22, №5, стр. 362-372.

Приложение 1

Рисунок 5. Диплом за лучшую работу на конференции Е.В. Арменского

Приложение 2

Рисунок 6. Диплом победителя отборочного этапа «УМНИК»