электронный журнал
МОЛОДЕЖНЫЙ НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК
Издатель ФГБОУ ВПО "МГТУ им. Н.Э. Баумана". Эл No. ФС77-51038.

УДК 004.023

Обзор способов распознавания посетителей веб ресурсов на основе получаемой или оставляемой ими информации

Цатурян Т. Ш., *студент*
кафедра «Системы обработки информации и управления»
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана

Научный руководитель: Гапанюк Ю. Е., к.т.н., доцент
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана
Gapyu@bmstu.ru

Введение.

На сегодняшний день из 7 миллиардов жителей земли 40 процентов (2.7 миллиарда) подключено к интернету [1]. Когда вы попадаете на веб ресурс, вам кажется, что вероятность определения составляет 1 на 2.7 млрд., что является относительно очень маленьким числом. Это не так. Документы, опубликованные Эдвардом Сноуденом, указывают на утрату частной жизни и что имея достаточное количество средств, современное оборудование и технологии, можно осуществлять тотальное слежение за телекоммуникациями, персональными данными и активностью во всемирной сети [2]. Несмотря на то, что текущее внимание к анонимности достаточно повышено в связи с публикациями от Сноудена, проблема защита персональных данных является актуальным вопросом.

Исследования, проводимые различными университетами и компаниями, показывают, что персональные данные могут появляться там, где их не ожидали. Например, любое физическое устройство – уникально. Вы не найдете и двух одинаковых деталей: размеры, шероховатость поверхности, цвет и т.д. являются неповторимыми. Это влияет и на переносимую или представляемую информацию.

Камера вашего фотоаппарата была, очень возможно, выпущена на конвейере в размере минимум десятки тысяч копий. Но линза, установленная в фотоаппарате, имеет уникальные характеристики. Свет, преломляясь и внутренне отражаясь, создает “картинку” на матрице и, соответственно, изображение в памяти. Данная картинка имеет разницу всего в пару пикселей от оригинала и незаметна обычным глазом. Компьютер же отличает и создает неповторимый отпечаток вашей камеры.

В данной статье рассматривается тема обнаружения пользователя на веб

ресурсах без каких-либо дополнительных средств. Локально установленные шпионы, логи и записи маршрутизаторов, вирусы, взлом браузеров и иные дополнительные противоправные действия не используются. То есть в итоге мы имеем, расположенный где-то, сайт (сервер), ресурсы этого сервера и не более того.

Типы данных и идентификационная фраза.

Для осуществления распознавания, компьютерной программе нужны данные. Чем больше – тем лучше (тем точнее результат). Главные критерии - чтобы данные были полезные (т.е. имеющие смысл), были интерпретируемы, могли храниться, могли быть обработаны и являлись не очевидными (не могли быть выведены из существующих данных). Последнее особенно важно, потому что в противном случае данные являются подтверждающими, но непригодными для использования в задачи распознавания.

Например, знание о том, что браузер использует протокол HTTP для доступа к ресурсам, очевидно, является неким примером тривиальности и не несет пользы. Гораздо более интересно – какие именно заголовки отсылает браузер при запросе (Request Headers). Они отличаются, и они могут быть полезны.

Все данные (вектор значений) делятся на 2 большие группы:

- Установленные нами (скриптами сайта)
- Существующие до нас (до входа на сайт) и не имеющие отношения к деятельности нашего сайта

Принцип в установленных нами значениях лежит в генерировании сложной, мало повторимой и трудно подделываемой фразы, которая сохраняется на браузере пользователя. В дальнейшем, сайт, получая доступ к хранилищу, ищет фразу и при совпадении идентифицирует клиента. Малая повторяемость влияет на частоту и ложность срабатывания. Защита от подделки препятствует выдачи браузера за чужого, путем подделки чужой фразы.

Существующие до нас данные должны обладать теми же характеристиками, что и установленные нами. Тут существуют 2 проблемы – потенциально высокая повторяемость и возможная подделка. Например, в среднем браузер Google Chrome установлен на 46.02 процентов компьютеров и это миллиарды браузеров.

Обычно, используются установленные нами значения, чем уже существующие. Но в общем случае структура получения идентификационной фразы (ключа) выглядит как на рис. 1. Выходящие данные (идентификационная фраза) – строка символов, точно определяющая выбранный браузер и никакой иной. В 90 процентов сайтов

идентификационная фраза совпадает, например, с cookies (см. далее), но это необязательно.

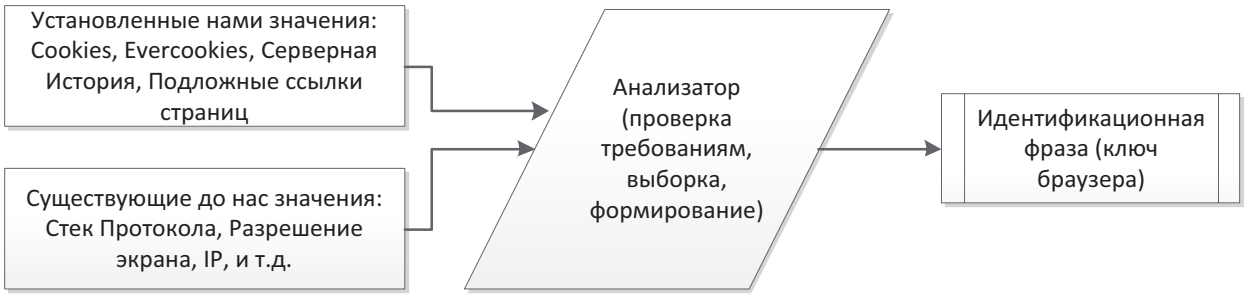


Рис. 1. Комбинирование данных для получения идентификационной фразы (ключа).

Основное требование к идентификационной фразе – полнота (точно характеризующая браузер) и максимальная сжатость (возможность использования её для ключа к СУБД, например).

СХЕМЫ ПОЛУЧЕНИЯ ДАННЫХ.

Cookies :

Ку́ки (от англ. cookie — печенье) — небольшой фрагмент данных, отправленный веб-сервером и хранимый на компьютере пользователя. Веб-клиент (обычно веб-браузер) всякий раз при попытке открыть страницу соответствующего сайта пересылает этот фрагмент данных веб-серверу в виде HTTP-запроса. [4]. Применяется для сохранения данных на стороне пользователя, на практике обычно используется для:

- аутентификации пользователя;
- хранения персональных предпочтений и настроек пользователя;
- отслеживания состояния сессии доступа пользователя;
- ведения статистики о пользователях. [4].

Преимуществом является простота установки. Для установки не требуется поддержки JavaScript в браузере, т.к. это может быть спокойно осуществлено средствами языка php. Недостатками являются широкая известность и распространенность, поэтому существует огромное кол-во инструментов для модифицирования cookies или их очистки. От изменения есть простое и достаточно эффективное решение – сохранять на сервере все выданные cookies, а в качестве cookie использовать длинную и специально подготовленную строку. В идеале – со сложным обратным преобразованием (например, md5). А для правильного изменения требуется расшифровать строку, что может быть трудной задачей.

Evercookies

Это идея cookies, с возможностью оставлять данные в местах, куда позволяет сохранить браузер. Повсеместное распространение программ для очистки и модификации cookies привели к поиску мест сохранение устанавливаемой фразы, остальной принцип действия остается таким же.

Были определены следующие места сохранения:

- Стандартные HTTP Cookies (как дубль).
- Local Shared Objects (Flash Cookies)
- Silverlight Isolated Storage
- cookies в RGB цветах PNGs
- Размещение cookies в Web History
- Размещение cookies в HTTP ETags
- Размещение cookies в Web cache
- window.name
- Internet Explorer userData storage
- HTML5 Session Storage
- HTML5 Local Storage
- HTML5 Global Storage
- HTML5 Database Storage

IP

IP (Сетевой адрес компьютера) известен на сервере при обращении к нему браузера. Это последний хост, перед нашим сервером. Его невозможно подделать. Использование IP для определения пользователя имеет очень существенные недостатки, и не будет работать (или будет работать плохо) для машин, которые имеют внутренний IP адрес, а выходят через шлюз с одним внешним ip. В данном случае будет определяться именно внешний IP или IP шлюза. Использование Tor или прокси серверов аналогично изменит IP и сделает привязку бессмысленной.

Несмотря на недостатки, IP привязка широко используется, например, для целей безопасности: при обращении с незнакомого IP, система может запросить дополнительное подтверждение.

Цифровой отпечаток браузера

Цифровой отпечаток браузера – это набор данных, которые мы можем получить, описывающий браузер. Примерами являются: Разрешение экрана, User Agent, Заголовки HTTP_ACCEPT, Данные об установленных плагинах, Установленные Шрифты.

Собрав их воедино, мы получаем цифровой отпечаток браузера, который может быть очень популярным или очень редким, стоит лишь пользователю установить редкий шрифт или добавить мало популярные плагины.

Цифровой отпечаток – существующие до нас характеристики. Они могут меняться (например, при добавлении шрифта), но в целом, набор является более-менее фиксированным.

Дата и время посещения

Вероятность посещения того же самого ресурса в короткий срок (1-2 дня), выше чем при длинном промежутке времени. (Зависит от ресурса).

Пример и применение: Если неизвестные пользователи с редким отпечатком браузера зашли в достаточно короткий промежуток времени несколько раз, то, скорее всего, пользователь один.

Использование времени может быть эффективными при точном понимании специфики обращений к ресурсу.

Кэш файлы

Интересный механизм, который работает следующим образом: формируется специальный файл, который помечается как кэшируемый, чтобы браузер его внес в кэш. При повторном доступе на сайт, браузер не будет считывать файл, потому что он есть в кэше, что и будет служить признаком повторного появления.

Еще один вариант - страница может содержать ссылку <script type="text/javascript" src="example.js">. С загрузкой страницы загружается и example.js. Затем скрипт кэшируется и не требует загрузки при повторном посещении страницы. В результате, если скрипт содержит такое значение, как id=3243242, этот идентификатор остаётся в силе и может быть использован другим сценарием javascript или другой страницей, запрашивающей этот скрипт.

Это явный пример данных, устанавливаемый нами. По сути, он работает как cookie (evercookie), только технология доступа кардинально отличается.

Подложные ссылки страниц

Механизм был детально описан мною в бакалаврской работе [5]. Способ рассчитан на:

a. Сохранение страницы в закладках, в текстовом файле и т.д.

b. Передачи адреса страницы в виде ссылки

Идея в том, что мы создаем псевдоним страницы и отображаем его пользователю. Пользователь работает с псевдонимом, и он не знает реальное название страницы, но скрипт, работающий на сервере, знает реальное название страницы, название псевдонима и уникальный номер пользователя. При повторном открытии псевдонима, не составляет труда узнать уникальный номер пользователя и восстановить информацию.

Данные из ранее посещенных страниц.

При обращении к странице некоторые браузеры при отправке запроса указывают адрес предыдущей страницы в поле HTTP_REFERER.

Его можно считать на сервере командой $prev_page = ($_SERVER['HTTP_REFERER']);

Способ активно используется seo-мастерами при раскрутке сайта и анализе его эффективности, т.к. это значение указывает на предыдущую страницу, откуда пришел пользователь. Иногда эта информация может быть полезна: Например, на сайте vk.com (популярная социальная сеть ВКонтакте) если пользователь на главной странице, то в адресной строке будет отображаться его уникальный id (или ник). При переходе с этого сайта, мы сможем получить id пользователя.

“Биометрические данные”

Существуют предпосылки распознавания пользователя на основе его уникальных человеческих характеристик, таких как (в случае с компьютером) ошибки при наборе слов, фигуры движения мыши, скорость нажатия клавиш, скорость пробелов, скорость набора и иное. Существует программа для установки, которая будет блокировать систему и просить ввода пароля для авторизации, если предположит, что за рабочим местом сидит иной человек. Мне не удалось обнаружить какие-то разработки в данной области или что-то рабочее.

Я самостоятельно пытался сделать небольшое исследование в этой области, но пришел к выводу, что:

1) Для реализации этой схемы в web требуется: быстрый канал (скорость доступа к сайту) без задержек и перебоев, колоссальные требования к объемам БД в

которой будут храниться нажатия клавиш, задержки и прочее.

2) Уверенность, что данные методы вообще работают. Набирая текст на другой клавиатуре или переместив её в неудобное положение, мы рискуем оказаться новым для системы, т.к. текущие параметры не будут совпадать со старыми.

СРАВНЕНИЯ И ПРИМЕНИМОСТЬ

Рассмотрим преимущества, недостатки и возможные методы обхода типов авторизаций.

Таблица 1

Сравнение

Схема	**Требования для работы**	**Плюсы**	**Минусы**	**Комментарий**
Cookie	Браузер должен принимать cookie	Работает при выключенном js и при отсутствии flash	Широко распространенная технология и как следствие существует огромное кол-во программ для удаления	Пользователи могут изменять cookie, поэтому рекомендуется их генерировать максимально зашифрованным способом
Evercookie – набор различных cookie	Для каждого метода свой. Для некоторых нужен js, для некоторых flash, некоторые работают без дополнительного ПО.	Огромное число мест хранения и более низкие шансы для полного удаления	Для некоторых методов требуется flash или javascript	

IP	Нет	Работает всегда (IP всегда можно получить)	Ввиду огромного кол-во аномайзеров, tor, прокси-серверов неточный метод определения.	
Цифровой отпечаток браузера	JavaScript и Flash	В случае уникальности какого-то параметра, отпечаток может быть достаточно точным	Требуется JavaScript и Flash. Если рабочие станции идентичные, то определить точно какую-нибудь маловероятно.	
Дата и время посещения	Нет	Легко получить.	Не несет особой информации	Используется как дополнительный метод
Кэш файлы	Нет	Не требует дополнительных настроек, разрешения на запуск или ПО.	Неточен, не работает после очистки кэша, при переключении на иной браузер и т.д.	
Подложные ссылки страниц	Нет	Работает практически всегда	Не работает, если заход не осуществляется по ссылке	
Данные из ранее посещенных страниц	JavaScript		Работает очень редко и данные малополезны	Используется как дополнительный метод

Биометрические данные	В общем случае достаточно JavaScript	Неизвестно	Не известно. Как минимум, большой объем БД, вычислительных ресурсов и широкий канал.	

МАТЕМАТИЧЕСКИЙ РАСЧЕТ НЕОБХОДИМОГО ОБЪЕМА ДАННЫХ ДЛЯ МЕТОДА ЦИФОРОВОГО ОТПЕЧАТКА.

На основании данных, полученных Electronic Frontier Foundation, приводим математическую базу. Допустим, у нас есть алгоритм получения идентификационной фразы (ключа), основанной на одном или группе методов описанных выше.

Примем F такое, что при появлении x новых браузеров (новых версий), вероятностная функция P имеет следующий вид.

$$P(f_n),\ n \in [0, 1, .., N] \quad (1)$$

Положим названием уникальность (или сюрприз) выход следующей функции:

$$I\big(F(x) = f_n\big) = -\log_2\big(P(f_n)\big), \quad (2)$$

Уникальность I, измеряется в битах (т.к. мы прологарифмировали уникальность отпечатка по основанию 2).

Энтропия распределения P(fn) – это ожидаемое значение уникальности по всем браузерам:

$$H(F) = -\sum_{n=0}^{N} P(f_n)\log_2\big(P(f_n)\big) \quad (3)$$

Уникальность (или удивление) имеет следующий физический смысл: это кол-во информации об объекте, над которым применятся функция цифрового отпечатка, где каждый бит вдвое уменьшает кол-во вариантов (т.к. мы применили логарифм). Если веб сайт регулярно посещают X различных браузеров, то конкретный браузер $x \in X$, будет точно распознан, если

$$I\big(F(x)\big) \gtrapprox log_2|X| \quad (4)$$

Обобщенный алгоритм действий:

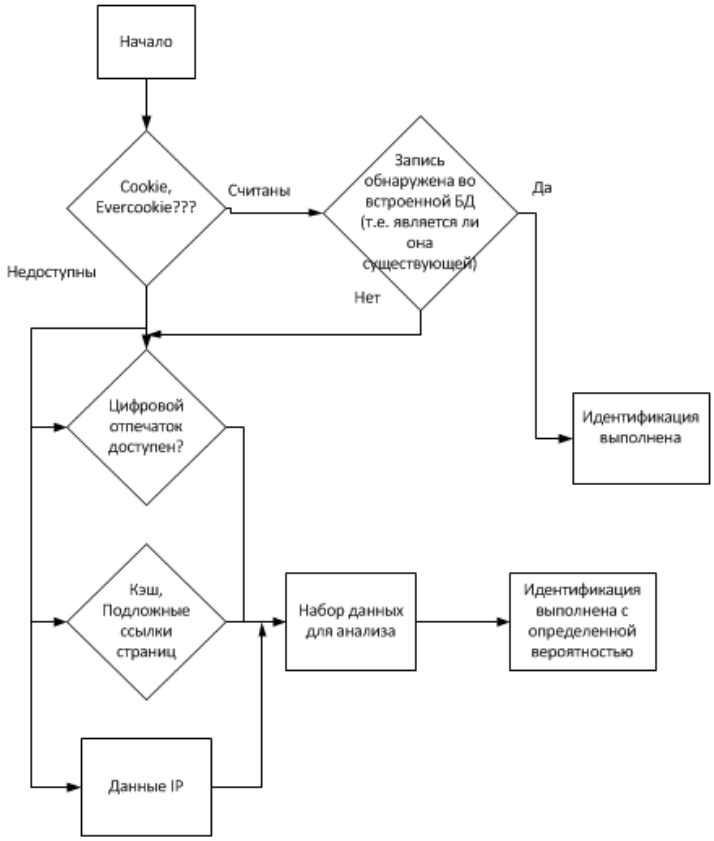


Рис. 2. Общий алгоритм распознавания.

ВЫВОДЫ:

Предложенные технологии слежения не требуют специального оборудования или разрешений, доступных узкому кругу лиц. На любом сервере (даже 5-6 летней давности) могут быть развернуты описанные методы и система “получай, анализируй и сохраняй все” работает.

При написании сравнительного анализа, я построил и запрограммировал

большинство описанных методов. Время, требуемое для написания системы, оценочно составляет около 1-2 месяцев при работе в несколько человек. В расчет не входит проведение исследований и построение блоков анализа.

Пользователям стоит предпринимать дополнительные меры для обеспечения невозможности прослеживания и относиться серьезно к публикуемой ими информации в интернете.

Список литературы

1. ICT Facts and Figures // Itu.int: ICT Data and Statistics Division Telecommunication Development Bureau International Telecommunication Union. URL.http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2013.pdf (дата обращения: 05.09.2013).
2. Касперский: После разоблачений Сноудена утрачено понятие о тайне частной жизни // rosbalt.ru: Росблалт. URL.http://www.rosbalt.ru/moscow/2013/09/05/1172105.html (дата обращения: 05.09.2013).
3. How Unique Is Your Web Browser? // Dl.acm.org: Electronic Frontier Foundation URL.http://dl.acm.org/citation.cfm?id=1881152 (дата обращения: 05.09.2013)
4. HTTP cookie // ru.wikipedia.org: Википедия. URL. http://ru.wikipedia.org/wiki/HTTP_cookie (дата обращения: 05.09.2013).
5. Бакалаврская работа: Система Сбора Идентификационной Информации Посетителей Сайта. МГТУ им. Н.Э. Баумана 2013.