Мануал: Настройка реальных конфигов с нуля. Работа с логами: с Советы и фишки при работе с логами и использованию антидетекта

Получение базовой информации с лога о системе

В логе самая базовая информация о системе содержится в файле System.txt, либо Information.log. Screenshot: https://prnt.sc/lx4rp1.На скриншоте я выделил параметры, которые нужны нам для настройки системы.

1. **Windows** – Данный параметр содержит информацию о версии Windows и разрядности системы (32-х битная или 64-х битная, 64-х битная встречает намного чаще). Чаще всего будут вам встречаться логи Windows 7, Windows 10 , реже – Windows 8, 8.1, XP. Данный параметр нам будет нужен для настройки «navigator.UserAgent», и некоторых производных.
2. **Display Resolution** – Данный параметр содержит информацию о разрешении экрана пользователя. Необходим для настройки всех параметров, связанных с разрешением экрана и размером окна браузера и проивозодных параметров.
3. **Display Language, Keyboard Languages** - Данные параметры содержат информацию о языке/языках системы. Нужны для настройки параметров «navigator. Language», «navigator. Languages» и HTTP_ACCEPT_LANGUAGE.
4. **CPU Count** - Данный параметр содержит информацию о количестве количество потоков процессора. Нужен для настройки параметра «navigator.hardwareConcurrency»
5. **RAM** - Данный параметр содержит информацию о количестве оперативной памяти. Необходим для настройки «navigator.deviceMemory»
6. **Videocard** - Данный параметр содержит информацию о видеокарте системы. Необходим для настройки WebGL. Обращаю внимание, что система может содержать две видеокарты: одну дискретная, а другая интегрированная. Такое используется обычно на ноутбуках. И какая из них запускается для браузера на 100% неизвестно. Во-первых, пользователь может вручную задать, какая именно видеокарта будет использоваться, во-вторых, к примеру, может быть так: если ноутбук на зарядке – используется дискретная видеокарта, если же от батареи – то интегрированная. Поэтому в ноутбуках на 100% на этот параметр полагаться не стоит.
7. **[Network] Параметры** берем почти все, кроме Geo (Latitude and Longitude); Данная информация пригодятся вам для более грамотного подбора Socks/SSH-туннеля. ZIP в моем логе нет, но пробить его сложности не составляет. Для этого нужно лишь пробить IP адрес по базе MaxMind, либо найти домашний адрес пользователя в автозаполнении браузера, либо на почте или в шопе. Подбирать IP желательно не только как можно ближе по ZIP-адресу, но и по возможности с той же маской IP и того же интернет провайдера.

Следующим нашим шагом будет определить тип браузера и браузеры для создания конфигурации. Бывает, что владельцы ПК используют несколько браузеров, а не только один. Поэтому при необходимости лучше создавать две сессии в сфере, т.е. две конфигурации, а не грузить cookie в одну. Для этого смотрим нужным нам сайты с логинами и паролями в файле «passwords.txt» параметр «Soft» Screenshot: https://prnt.sc/lx5ofi, а также файлы в папке «Cookies» на наличие нужных сайтов (файлы в этой папке поделены на браузеры; возможен вариант, что файлы Cookies могут лежать в общей папки. Все зависит от того, с какого стиллера лог). Пример: http://prntscr.com/lx5oag

В моем случае в логе только один браузер Google Chrome, поэтому я помечаю себе только 1 браузер. Переходим к более интересной информации, которая не лежит на поверхности.

Определяем, имеется ли FLASH в системе и его версию, определяем версию браузера (если возможно)

Связь с автором: Jabber: wirl@exploit.im

Для этого заходим в файл System.txt, либо Information.log и в разделе установленные программы [Software] ищем «Adobe Flash Player». Если нашли, то помечаем, что Flash есть, записываем его версию. Существует два вида программы Adobe Flash Player: Adobe Flash Player ** NPAPI – под браузер Firefox. Adobe Flash Player ** PPAPI– под браузер Opera/Chrome. Screenshot: http://prntscr.com/lx5ztv

Следом на этом же скриншоте мы видим версию Google Chrome, если нет, то пробуем ее найти в файле по запросу «Google Chrome». Также помечаем себе версию. Тип браузера и его версия будет нужна нам для настройки параметра «navigator.UserAgent», ну и в исключительных случаях для того, чтобы отключить подмену Canvas . Браузер Mozilla Firefox ищем по запросу «Firefox», найти мы должны примерно такое «Mozilla Firefox 64.0 (x64 en-US) [64.0]». В названии браузера Firefox содержится битность программы (32 или 64 bit), что также пригодится в настройке «navigator.UserAgent». Браузер Opera ищем по запросу «Opera», найти мы должны примерно такое «Opera Stable 57.0.3098.106 [57.0.3098.106]».

По разным причинам версию браузера не всегда удается определить, одна из которых – браузер может быть Portable, т.е. не установлен в системе. Браузер IE видно не будет, т.к. он уже изначально в Windows, с Edge в Win 10 такая же шляпа.

Наличие Flash и его версия понадобится нам для того, что добавить его в плагины и, при необходимости, включить его физическую версию в антидетекте.

Определяем у пользователя стационарный компьютер (Desktop) или Ноутбук(Laptop)

Определить это можно, используя различные варианты.

1. **По скриншоту экрана в логе.** На скриншоте экрана мы ищем то, что свойственно ноутбуку на панели задач в правом нижем углу, либо на рабочем столе то, что свойственно ноутбуку (значки программ для ноутбука и.т.п.).

На панели задач можно найти Значок батареи, значок подключения Wi-Fi. Покажу теперь это на примерах.

Примеры: http://prntscr.com/lx86z7

https://prnt.sc/lx871y

2. **По информации о процессоре в системе.** Для этого заходим в файл System.txt, либо Information.log и смотрим параметр «Processors» Screenshot: https://prnt.sc/lx88az

Копируем значение и гуглим информацию о процессоре. Вот пример информации по этому процессору с сайта Intel, который показывает нам, что у пользователя стационарный компьютер. Screenshot: https://prnt.sc/lx89jp

Пример информации о процессоре для ноутбука. Screenshot: http://prntscr.com/lx8g8y

Ну и еще один вариант - поискать в процессах или установленных программах в файле System.txt, либо Information.log процессы/программы, которые относятся к ноутбуку. Например, это процессы, в которых фигурирует ключевое слово «Bluetooth», программы, характерные определенному производителю ноутбуков (ASUS, DELL, MSI, ACER и др.)

Примеры процессов: «Intel (R) Wireless Bluetooth (R)», «Dell Touchpad».

Знать несколько вариантов необходимо, потому что иногда скриншота может и не быть, или скриншот получается определённой области без панели задач, бывает панель задач скрыта.

Связь с автором: Jabber: wirl@exploit.im

Панель задач: определяем положение панели задач на экране, размер значков и скрывается ли панель задач (если возможно)

Первый вопрос, который приходит в голову: «Нахрена это нужно?». Отвечу: это нужно для того, чтобы выставить размеры экрана; размеры окна браузера и размеры рабочей области браузера в полноэкранном режиме работы браузера (параметры «window.innerWidth», «window.innerHeight», «window.outerHeight», «window.outerWidth»).

Конечно, не в каждом логе будет такая возможность посмотреть и на все 100% понять. Иногда скриншота может не быть, бывает скриншот не полной области экрана.

Сейчас я покажу, как правильно оценить данные параметры. Screenshot: https://prnt.sc/lxy3x0

Данные примеры сделаны на ОС Windows 7. При желании вы сами можете потом посмотреть и поиграться с данными настройками на любой ОС Windows.

1) **Положение панели задач**. Бывает: горизонтальное и вертикальное. У большинства пользователей положение стоит по умолчанию: горизонтальное.
2) **Размер значков панели задач.** Бывает два размера значков: большие и маленькие. По умолчанию стоит большой размер значков. Большие значки установлены у большинства пользователей. На Windows 7 особенность есть: если значки маленькие, то значок кнопки «пуск» выступает за области панели задач. Иногда не всегда даже по скриншоту можно понять размер значков, советую также обращать внимание на Display Resolution в логе; одно дело скриншот размера экрана «1024 x 768», другое дело «2560 x 1440»
3) **Скрытая панель задач.** По умолчанию у большинства пользователей панель задач не скрыта. Скрытая панель задач не означает, что ее вообще нет. Она лишь не отображается на экране, но при наведении курсора мышки появляется. Если у вас в логе полный скриншот экрана и там нет панели задач, то она как раз скрыта.
4) Если на скриншоте у владельца ПК открыт нужный вам тип браузер, это также пометьте, пригодится в настройке. Скриншоты с открытым браузером встречаются довольно таки часто.

Сеть пользователя: определяем примерный роутер и его модель (по возможности)

Иногда по логу можно определить бренд роутера пользователя или его примерную модель. Это может быть необходимо для более точной настройки WebRTC, а точнее – Local IP Address.

Для этого нужно посмотреть в логе в файле с логинами/паролями или в файле, где хранится история браузера, популярные маски IP адресов роутеров. Вот ссылка на таблицу брендов самых популярных роутеров и локальных ip адресов по умолчанию:

https://docs.google.com/spreadsheets/d/1GySRwS_QAmvPSJEDxYcsGnz_7Vu_mtj0nn_RvY4wgl4/edit?usp=sharing

Самые популярные маски для поиска в логе: «192.168.», «10.0.», «10.1.», «10.90.». Самые популярные бренды я выделил в таблице светло синим.

Если там еще будет указан логин и пароль, можно попробовать вот тут посмотреть по брендам стандартные связки логин/пароль: https://192-168-1-1ip.mobi/default-router-passwords-list/

Вот на примере https://prnt.sc/ly3sww можно предположить, что у пользователя ПК роутер D-Link. Но это не 100%, так как еще несколько роутеров имеют такую же связку.

Куда более точную информацию иногда нам может показать файл истории браузера. Вот пример: https://prnt.sc/ly41tw

Связь с автором: Jabber: wirl@exploit.im

В истории браузера мы видим Local IP Address и плюс еще заголовок страницы, что дает нам огромный плюс в определении роутера. Если загуглить «B593s-931», то можно определить что это название роутера «HUAWEI B593s-931». Еще один пример: https://prnt.sc/ly49nx

Если загуглить «userRpm/DdnsAddRpm.htm», то можно увидеть, что роутер относится к TP-Link TL-WR741N / ND, либо к TL-WR841N или некоторым другим.

Помимо Local IP Address WebRTC, информация будет полезна, если кто-то меняет MAC адрес, так как «начало» MAC-адреса у каждого производителя свое.

Плагины браузера: определяем популярные плагины, которые установлены в браузере.

Плагины в любых программах – это дополнения, которые позволяют расширить её возможности. Большинство популярных браузеров имеют возможность устанавливать плагины, которые позволяют расширить его возможности. Например, это может быть плагин Flash от Adobe, возможность читать PDF страницы в браузере; в Chrome данный плагин уже идет по умолчанию; возможность запускать какие-либо Аудио/Видео кодеки.

С каждым новым выходом обновлений, увеличивается количество новых функций и вариаций поддерживаемого контента, поэтому плагины постепенно теряют свою актуальность. В итоге в браузерах Chrome, Firefox, Opera, Edge остались только встроенные плагины и один добавляемый: Adobe Flash Player. Поэтому с поиском плагинов больше актуально для браузера Internet Explorer, либо для старых версий Firefox (ДО 52 версии), Chrome, Opera.

Самые популярные плагины: Flash, Java, Microsoft Office, Adobe PDF Reader, Windows Media Player, Real Video/Audio.

В начале статьи мы уже определяли, имеется ли Flash в системе. Так вот Flash Player также является плагином в браузере. Поэтому если Flash есть, то и в некоторых типах браузера он будет в плагинах. Помечаем себе, если он имеется.

Искать остальные плагины будем так же в файле System.txt, либо Information.log в разделе установленные программы [Software].

Плагин QuickTime находим по запросу «QuickTime», примерное название плагина: «QuickTime 7 [7.79.80.95]»

Плагин Silverlight находим по запросу «Microsoft Silverlight», примерное название плагина: «Microsoft Silverlight [5.1.50907.0]»

Плагин Java находим по запросу «Java», примерное название плагина: «Java 8 Update 191 [8.0.1910.12]»

Плагин RealPlayer находим по запросу «RealPlayer», примерное название плагина: «RealPlayer [18.1.15.]»

Плагин Adobe Acrobat (для чтения PDF файлов) находим по запросу «Adobe Acrobat Reader DC», в итоге будет – что то вроде «Adobe Acrobat Reader DC [19.010.20064.]»

Есть еще множество других различных плагинов, это был лишь пример популярных плагинов. Список продолжать можно очень долго.

На этом сбор информации по логу закончен. По итогу у нас собрана вот такая информация:

Windows: Windows 10 Home [x64]

Display Resolution: 1920x1080

Связь с автором: Jabber: wirl@exploit.im

Display Language: en-US

Keyboard Languages: English (United States)

CPU Count: 4

RAM: 8139 MB

VideoCard: NVIDIA GeForce GTX 970

[Network]

IP: 38.104.174.234

Country: United States (US)

City: Pleasant View (California)

ZIP: 93260

ISP: Cogent Communications (Txox Communications)

--

Browser: Google Chrome ver. 68.0.3440.106

Flash: имеется, ver. 30.0.0.154

--

PC: Ноутбук(Laptop)

--

[Панель задач]

Положение: Горизонтальное

Размер значков: Большой

Скрытая панель задач: Нет

Есть ли браузер в скриншоте: ДА

--

Router: ~TP-Link TL-WR741N or TL-WR841N

[Плагины браузера]

Adobe Flash Player

RealPlayer

Adobe Acrobat

Связь с автором: Jabber: wirl@exploit.im

Конечно данный пример имеет уж слишком много информации. На практике, ее может быть меньше.

Мануал по настройке реальных конфигов с нуля с использованием антидетекта.

Переходим к самому интересному разделу данной статьи.

Основа всех основ – UserAgent

UserAgent является основой в создании конфига. Как строительство дома начинается с фундамента, так и создание конфига с начинается с UserAgent (сокращенно UA). Начнем с теории. Разберемся, что такое UA.

UserAgent - это свойство (параметр), который содержит свойства, по которым и происходит определение — какой браузер, какая операционная система, какой версии, и какое специфичное ПО стоит у пользователя.

В конфигах любого Антидетекта данный параметр находится в navigator.UserAgent и в HTTP_USER_AGENT.

Примечание: navigator.UserAgent и HTTP_USER_AGENT всегда совпадают, но есть исключение: браузеры Internet Explorer. Очень часто в данных браузерах в navigator.UserAgent находится информации о программном обеспечении пользователя.

Пример:

HTTP_USER_AGENT: «Mozilla/5.0 (Windows NT 6.3; WOW64; Trident/7.0; rv:11.0) like Gecko»

navigator UserAgent: «Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; Media Center PC 6.0; rv:11.0) like Gecko»

Разбираем на практике, как составлять UA самых популярных браузеров в Windows. Начнем с самого простого – Mozilla Firefox. Структура UserAgent:

Mozilla/5.0 (<**Windows version**>; <**bit tags**>; rv: <***Firefox version***>) Gecko/20100101 Firefox/<***Firefox version***>

Выше я выделил те параметры, которые нужно знать для создания реального UA.

<**Windows version**> - Версии операционной системы. Варианты:

Windows NT 6.0 – Windows Vista, Windows Server 2008.

Windows NT 6.1 – Windows 7, Windows Server 2008 R2.

Windows NT 6.2 – Windows 8, Windows Server 2012.

Windows NT 6.3 – Windows 8.1, Windows Server 2012 R2.

Windows NT 10 – Windows 10, Windows Server 2016&2019.

Данный параметр есть во всех UA на Windows. **Примечание**: на браузерах Edge он статичен, т.е. не меняется, т.к. браузер заточен как раз только под Windows 10.

<**bit tags**> - «битность» системы. Думаю, все знают и все встречались с тем, что существуют две 32-битные системы Windows и 64-битные. Как раз браузер и передает возможные вариации:

Win64; x64 – данное значение передается, если система 64-битная.

Пустое значение (ничего не передается) если система 32-битная. Пример UA: Mozilla/5.0 (Windows NT 6.1; rv:60.0) Gecko/20100101 Firefox/60.0

Связь с автором: Jabber: wirl@exploit.im

WOW64 – данное значение передается когда 32-битное приложение браузера запущено на 64-битной системе.

<Firefox version> – данное значение показывает версию вашего браузера Firefox. Примечание: передается значение лишь с одной цифрой после точки, даже если версия браузера «63.0.3», то в UA будет передаваться лишь «63.0». Вот ссылка список всех актуальных версий Firefox: https://www.mozilla.org/en-US/firefox/releases/

Комбинируя данные значения, мы получаем различные UserAgent's. Не забывайте, что значение «rv:» и «Firefox/» обязательно должны совпадать.

Примеры:

Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0 – UserAgent Windows 10 [64bit] с браузером Firefox 64.

Mozilla/5.0 (Windows NT 6.1; rv:52.0) Gecko/20100101 Firefox/52.0 0 – UserAgent Windows 7 [32 bit] с браузером Firefox версии либо 52.0.1, либо 52.0.2

Mozilla/5.0 (Windows NT 6.1; WOW64; rv:43.0) Gecko/20100101 Firefox/43.0 – UserAgent Windows 7 [64 bit] с браузером Firefox, который рассчитан на 32-bit системы версией 43.0.1, либо 43.0.2, либо 43.0.3, либо 43.0.4

Переходим к браузеру Google Chrome.

Структура UserAgent Google Chrome:

Mozilla/5.0 (<**Windows version**>; <**bit tags**>) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/<***Chrome version***> Safari/537.36

Несмотря на то, что UA Chrome кажется сложнее, на самом деле он даже чуть проще, т.к. верси. хрома не нужно дублировать два раза.

<Windows version> и **<bit tags>** абсолютно такие же значения, как в Firefox.

<Chrome version> – данное значение показывает версию вашего браузера Chrome. Вот ссылка список актуальных версий Chrome: https://filehippo.com/download_google_chrome/history/

Пример: Chrome/71.0.3578.98

71.0.3578 – это **версия браузера.**

98 – **Build**. Он показывает, какое количество фиксов различных багов, улучшений было в данной версии.

Примеры:

Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36 – UserAgent Windows 8.1 [64 bit] с браузером Google Chrome версии 71.0.3578 с build 98.

Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.110 Safari/537.36 – UserAgent Windows 10 [64 bit] с браузером Google Chrome версии 70.0.3538 с build 110.

Связь с автором: Jabber: wirl@exploit.im

Mozilla/5.0 (Windows NT 10.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.110 Safari/537.36 36 – UserAgent Windows 10 [32 bit] с браузером Google Chrome версии 70.0.3538 с build 110.

Переходим к Opera.

Mozilla/5.0 (<**Windows version**>; <**bit tags**>) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/<***Chrome version***> Safari/537.36 OPR/<***Opera version***>

Браузер Opera реализован на движке WebKit и V8 в оболочке Chromium, поэтому и в UA имеется «Chrome/<***Chrome version***>» , можно сказать UserAgent не сильно таки отличается.

<**Windows version**> и <**bit tags**> и <***Chrome version***> абсолютно все одинаково, как я описывал выше. Единственный момент с версиями chrome, но об этом ниже.

<***Opera version***> – данное значение показывает версию вашего браузера Opera. Вот ссылка на список актуальный версий Opera: https://blogs.opera.com/desktop/

Нас интересуют больше всего «Stable update», «beta update, developer update, initial release» - в меньшей степени.

Пример: OPR/56.0.3051.116

56.0 – **версия** браузера

3051 – **Build** браузера

116 – **Patch** браузера.

Уточняю в чем особенность Chrome. У определенной версии Opera, определенная версия Chrome. Нельзя от балды написать версию Chrome или наоборот. Эти два значения должны быть согласованы.

Вот таблица, накидал в качестве примеров 11 версий браузера Opera.

https://docs.google.com/spreadsheets/d/1OglvdCpkWxr0GztpQ3Nzi3Ij0Ep4oEZxdfZn-PVwdqU/edit?usp=sharing

Примеры:

Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/68.0.3440.106 Safari/537.36 OPR/55.0.2994.44 - UserAgent Windows 10 [64 bit] с браузером Opera версии 55.0

Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/68.0.3440.106 Safari/537.36 OPR/55.0.2994.44 - UserAgent Windows 8 [64 bit] с браузером Opera 32-Bit версии 55.0

Переходим к браузеру Edge.

Структура UserAgent Edge:

Mozilla/5.0 (Windows NT 10.0; <**bit tags**>) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/<***Chrome version***> Safari/537.36 Edge/<***Edge version***>

<**bit tags**> и <***Chrome version***> абсолютно одинаково, как я писал выше.

Связь с автором: Jabber: wirl@exploit.im

<Edge version> - данное значение показывает версию вашего браузера Edge. Также как и Opera , у определенной версии Edge определенная версия Chrome.

Вот ссылка на актуальные версии Edge: https://en.wikipedia.org/wiki/Microsoft_Edge

Примечание: Нам нужны значения «EdgeHTML version» а не «Version».

Пример: Edge/17.17134

17 –EdgeHTML Версия

17134 – Window Build.

Таблица с примерами версий Edge Chrome

https://docs.google.com/spreadsheets/d/1QkUj5f0oPIUGU6aGyZSS9DNUpGCaywv9W50y-tvSVPM/edit?usp=sharing

Примеры:

Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.140 Safari/537.36 Edge/17.17134 - UserAgent Windows 10 [64 bit] с браузером Edge версии 17.

Mozilla/5.0 (Windows NT 10.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.140 Safari/537.36 Edge/17.17134 - UserAgent Windows 10 [32 bit] с браузером Edge версии 17.

На этом тему UA завершим, можно еще много чего сказать про существующие UA, так как я разобрал лишь самые основные браузеры и самые популярные и простые варианты. Если вообще данная статья зайдет, то раскрою более подробно про более сложные вариации UserAgent от разных типов браузера; про мобильные UserAgent и новые типы браузеров.

Иные варианты, где можно взять UA:
1) Реальные устройства.

2) https://developers.whatismybrowser.com/useragents/explore/

Множество различных UA по типу браузера, по ОС, мобильные UA и.т.д. Очень много вариантов выборки. Минусы в том, что и всякого «шлака» много, самых новых версий не так уж много; есть параметр популярности UA, но я бы не советовал на него ориентироваться.

3) Конфигшопы. Собственно в конфигшопах можно спокойно увидеть данный параметр и без покупки конфига. Вариант очень удобный, ведь можно сделать выборку по нужным параметрам и в конфигшопах самые актуальные UA. В некоторых можно просто спокойно зарегистрироваться. Ссылки сюда кидать не буду, кому будет очень нужно – пишите в ЛС или по контактам.

Пройдемся по простым параметрам настройки конфига в Linken Sphere (Extended session settings).

Navigator.vendor – данный параметр показывает имя поставщика браузера. В наших типах браузера Значение пустое, либо «Google Inc.». Параметр статичен, т.е. не меняется. Значения в наших типах браузера:

Firefox – пустое

Edge– пустое

Связь с автором: Jabber: wirl@exploit.im

Chrome - Google Inc.

Opera- Google Inc. [/QUOTE]

Navigator.ProductSub – данный параметр показывает Build номер браузера. Параметр статичен, т.е. не меняется. Значения в наших типах браузера:

Firefox – 20100101

Edge– 20030107

Chrome - 20030107

Opera- 20030107

Navigator.hardwareConcurrency – данный параметр показывает количество потоков процессора, а не количество физических ядер процессора, как полагают многие. Параметр не зависит от типа браузеров, которые мы рассматриваем. Популярные значения данного параметра: «2», «4», «8», «12».

Для лучшего понимаю рассмотрю новый процессор на ноутбуках: Intel Core i7-8750H.Это 6 ядерный процессор, но имеет он 12 потоков, следовательно параметре будет стоять значение «12» а не «6». Иногда количество потоков соответствует количеству ядер. По названию процессора можно всегда посмотреть в интернете данные значения. Насчет информации в логах – там как раз указана информация о количестве потоков, поэтому можно смело ставить данный параметр, но на всякий случай перепроверяйте (параметры: # of Cores и # of Threads)

Navigator.MaxTouchPoints – данный параметр показывает максимальное количество одновременных сенсорных нажатий, которые поддерживает устройство, т.е. если устройство имеет несколько сенсорных экранов с разными максимальными значение, то показывается максимальное значение. Параметр не зависит от типа браузеров, которые мы рассматриваем. Вообще обычно говорят, что этот параметр больше актуален для мобильных конфигов и это так, но не совсем.

Собственно обычный стационарный компьютер или ноутбук с подключенной мышкой и клавиатурой будет показывать значение «0». Чаще всего такое значение параметра.

Но встречаются сенсорные мониторы в ноутбуках, сенсорные мониторы для стационарных ПК. Поэтому в таком случае, значение параметра обычно «1» или «2». Поэтому при настройке наших типов конфига допустимо ставить данные значения.

По информации с лога не определить в 95% какой именно ноутбук или какой именно дисплей, поэтому лучше ставить значение по умолчанию «0».

Navigator.Platform - данный параметр показывает платформу, на которой работает браузер. В рамках наших типов браузеров и ОС могут быть два значения: «Win32» и «Win64». Но даже если Windows 64-bit и софт браузера 64 bit, все равно используется значение «Win32». Поэтому ставим только это значение.

Navigator.doNotTrack – данная технология позволяет включить или отключить запрет на отслеживание сайтами , различными системами. Самые популярные используемые значения: «Null» - пользователь не устанавливал данный параметр, следовательно, она не включена. Чаще всего используется этот параметр. «1», «true» - пользователь включил данную функцию, «0»,

«false» - пользователь выключил данную функцию. В конфигах можно использовать все три значения, предпочтительнее «null» либо «0».

Насчет подмены без антидетектов, в Google есть с картинками инструкция к каждому типу браузера как включить/выключить данную технологию.

Navigator.gamepads – данная технология показывает подключенные геймпады и их свойства (джойстики как на Xbox и Playstation). По значениям в сфере: «True» - функция включена, «False» - функция выключена.

Особенность: даже если в системе нет подключенных геймпадов эта функция включена. Так что в основном используем значение T»rue» для наших типов браузера вне зависимости от версии ОС. , Даже на большинстве мобильных браузеров функция также включена.

Navigator.battery – данная технология показывает информацию о состоянии батареи (идет ли зарядка, уровень зарядки в %, кол-во времени для полной зарядки/разрядки и.т.д.). По значениям в Linken Sphere: «True» - используется данная функция без подмен, «False», «Fake» - идентично True, только параметры самой батареи подменяются.

По использованию: в Edge, Firefox (после 52 версии) ставим только «False», в Chrome и Opera ставим либо «true», либо «fake».

Данная функция, как можно подумать, относится не только к ноутбукам. На стационарных компьютерах функция Battery включена. Разница в том, что параметры будут статичны, как будто это ноутбук на зарядке при 100%.

Информация по настройке данного параметра с лога: если вы определили, что пользователь ПК с ноутбука можно включить «fake», если же на вашей системе, где установлена Linken Sphere, статичные параметры Battery и у пользователя лога стационарный компьютер, то имеет смысл включить «True».

На реальных системах, если у вас имеется ноутбук, очень просто данный параметр изменять, нужно лишь разряжать/заряжать батарею. Тогда значения данной функции будут меняться.

Navigator.webdriver. Webdriver в браузере представляет собой программную библиотеку (драйвер), позволяющая другим программам осуществлять взаимодействие с браузером; управлять браузером. Данная технология появилась в браузере не так давно, поэтому является экспериментальной, информации по ней не так уже много. Технологию webdriver поддерживают все основные типы браузеров последних версий. Основные значения свойстваWebdriver : «true», «false» и «undefined» (не определено). По использованию в Linken Sphere: если делаем конфиги старых версий браузера (ниже 63 Chrome и 50 Opera , то используем значение «undefined»). В остальных случаях допускается использовать значение «true», и «false». Но с учетом специфики данной технологии и то, как она реализована в браузерах, советую в 95% случаев использовать «false».

Navigator.Online - данный параметр показывает статус браузера. Варианты параметра: «True», «False», «1», «0». Тут и ежу понятно, что значение должно быть «True» или «1». В сфере специально установлена возможность поставить только эти параметры.

Navigator.deviceMemory - данный параметр показывает объем оперативной памяти в ГБ. Значения: 0,25 – 256 МБ оперативной памяти, 0,5 – 512 МБ, 1 – 1024 МБ и.т.д. до значения 8. Если оперативной памяти больше чем 8 ГБ(12 ГБ, 16 ГБ,32 ГБ, 64 ГБ), то значение все равно будет «8». Если вы настраиваете конфиг браузера Firefox, то ставим значение «False», .т.к. там нет данного

параметра. Если вы настраиваете конфиг Chrome, Edge, Opera, то выставляем данный параметр (Работает в Chrome с 63 версии, Opera 50 и в Edge 17 версии). Самые популярные значения «2»,«4», «8».

Incognito - параметр показывает включен или выключен режим инкогнито (приватный режим) в браузере. Для работы ставим только «False».

Режим инкогнито в браузере, это когда не сохраняется история посещений в браузере, cookie файлы, автозаполнение и др. Отличный вариант для школьников смотреть порнуху, чтобы мамка или батя не спалили))

Настройка языка в конфиге

За язык в конфиге в антидетектах отвечают три параметра. Два в navigator (language, languages) один в заголовках браузера (HTTP_ACCEPT_LANGUAGE)

Navigator.language – данный параметр показывает язык интерфейса браузера (т.е. грубо говоря какой язык вашего браузера, а не системы, такой и будет отображать в этом параметре.) Пример: «en-US», «en-GB», «ru-RU» и.т.д. Этот параметр составляется так:

[Name of Language]-[Country codes]

Name of Language – ниже ссылка на список всех языков и обозначений: http://www.loc.gov/standards/iso639-2/php/code_list.php (брать значение из «ISO 639-1 Code»)

Country codes – ниже ссылка на список всех языков и обозначений:

https://www.iso.org/obp/ui/#search/code/ (брать значение из «Alpha-2code»)

Navigator.languages – данный параметр показывает предпочитаемые языки пользователя и берется из HTTP_ACCEPT_LANGUAGE

Пример: «en-US,en,ru-RU,ru», «de-DE,de,en-US,en»

Параметр составляется так для КАЖДОГО языка (каждый язык разделяется через запятую без пробела):

[Name of Language]-[Country codes], [Name of Language]

HTTP_ACCEPT_LANGUAGE - данный параметр показывает предпочтительные языки, который может понять пользователь (язык системы, язык интерфейса браузера) и «предпочтительность» языка.

Пример: «ru-RU,ru;q=0.9,en-US,en;q=0.7»

Параметр составляется так для КАЖДОГО языка (каждый язык разделяется через запятую без пробела):

[Name of Language]-[Country codes], [Name of Language]; q=[quality values]

quality values – значение «предпочтительности» языка. Может иметь значение от 0,1 до 0,9. Чем выше, тем язык предпочтительнее. Советую ставить для основного языка от 0,6 до 0,9, для второго от 0,4 до 0,7.

В сфере для настройки параметров языка нужно лишь настроить HTTP_ACCEPT_LANGUAGE

Связь с автором: Jabber: wirl@exploit.im

(https://prnt.sc/lypoyp). Самый простой способ изменить язык и без АД, это просто поменять язык в вашем браузере. В логе также имеется информация о языке пользователя и о языках раскладки клавиатуры.

Настройка параметров экрана

Переходим к настройкам параметров, которые относятся к экрану системы пользователя. Сильно в теорию вдаваться не буду, попробую очень просто на практике объяснить данные параметры.

Для начала, посмотрим вам основные параметры экрана в Linken Sphere наглядно на скриншоте:

http://prntscr.com/lz6cwt

Screen.width| device-width – данные параметры показывают ширину экрана в пикселях. Конечно, по некоторым тонкостям это разные параметры, но в рамках нашей статьи я их объединил, так как значения будут совпадать.

Screen.height| device-height – данные параметры показывают высоту экрана в пикселях. Объединил по той же причине.

device-width и **device-height** в сфере настраивать в общих настройках (Physical screen size).

Screen.width и **Screen.height** настраивать в настройке экрана сессии (НЕ в Extended settings)

Screen.availWidth – данный параметр показывает ширину экрана в пикселях, которую может занять браузер. На скриншоте у нас панель задач находится внизу, следовательно, связана не с шириной, а с высотой; браузер может занять полную длину. Поэтому Screen.availWidth= Screen.width| device-width

Screen.availHeight– данный параметр показывает высоту экрана в пикселях, которую может занять браузер. На скриншоте для того чтобы браузеру занять полную высоту, «мешает» панель задач, поэтому этот параметр будет рассчитан как Screen.width МИНУС высота панели задач.

Разберем примеры из первой части статьи, когда мы там смотрели панель задач.

Теперь еще более подробно и на примере. Возьмем экран Full HD 1920x1080. Если панель задач по умолчанию (внизу, с большими значками как на скриншоте) то ее высота будет 40 px. При таких значениях «Screen.availWidth» будет 1920, а «Screen.availHeight» 1040 px (1080-40 =1040)

Если значки в панели задач будут маленькие, то высота панели задач будет 30 px, а значение «Screen.availHeight» будет 1050 px

Если панель задач будет скрыта, то значение «Screen.availHeight» будет 1080 px.

Точно такая же история будет, если панель задач будет размещена не внизу, а вверху.

Далее, панель задач может быть размещена справа или слева и тогда будет меняться параметр «Screen.availWidth». По умолчанию он будет 1858 (1920 МИНУС ширина панели задач 62 px). Если значки будут маленькие, то при таком размещении панели задач в ширине панели ничего не меняется, и значение будет 1858; если панель задач будет скрыта, то значение будет 1920.

Вот собственно для чего мы смотрели на панель задач со скриншота экрана в логе.

Screen.availTop - показывает первую верхнюю (вертикальную) координату экрана пикселя, которая не занята панелью задач.

Связь с автором: Jabber: wirl@exploit.im

Screen.availLeft - показывает первую верхнюю (горизонтальную) координату экрана пикселя, которая не занята панелью задач.

Если панель задач размещена внизу или справа, данные параметры будет иметь значение «0». Исключение: если имеется второй монитор, то параметр «Screen.availLeft» может быть отрицательным и даже положительным.

Если панель задач размещена вверху или слева, то данные параметры будут иметь значения в зависимости от того, большие или маленькие значки. Если панель задач будет при этом скрыта, то данные параметры будут иметь значение «0».

В остальном: если панель задач слева по умолчанию, то «Screen.availLeft» будет иметь значение 62 px, если значки панели задач будут маленькие, то также 62 px (так как при боковом ее размещении ширина не меняется)

Если панель задач будет вверху, то «Screen.availLeft» будет иметь значение 40 px, если значки панели задач будут маленькие, то значение будет 30 px.

Проще говоря, Screen.availTop показывает высоту панели задач, если она размещена вверху, Screen.availLeft показывает ширину панели задач, если она размещена слева.

Зная размещение панели задач про скриншоту экрана в логе, мы можем рассчитать данные параметры.

Приведенные выше значения актуальны для Full HD экрана 1920 px на 1080 px.

Во вложениях темы (в самом низу) я прикрепил самый простой, но очень удобный чекер для расчета параметров экрана и окна браузера. Я не программист и не увлекаюсь этим, поэтому чекер правильно и стабильно работает только на Chromium браузерах (Chrome, Opera). Поэтому на Firefox данный чекер использовать не стоит. Надеюсь, найдется человек, который знает jQuery и адаптирует этот простой чекер и под Firefox.

Бонусом ссылка на таблицу, где я указал самые популярные значения разрешения экрана по ОС и рассчитал параметры для разных разрешений браузера:

https://docs.google.com/spreadsheets/d/12KM12QLMdwdmBKDuxlM-uh31WZNMQ6hdlIz_QnipAVg/edit?usp=sharing

Screen.colorDepth и **Screen.pixelDepth** – данные параметры показывают качество цветопередачи. Значения данных параметров совпадают. Возможные значение «24» и «32». В рамках нашей статьи ставим только «24». Значение «32» имеют такие устройства как Iphone, Ipad и.т.д.

Screen.orientation – данный параметр показывает информацию об ориентации экрана. Проще всего это объяснить скриншотом. https://prnt.sc/lz7j8w

Используем в рамках ПК только параметр «landscape-primary»; остальные параметры для мобильный устройств, планшетов и.т.д.

Screen.angle – данный параметр показывает угол поворота экрана. «landscape-primary» значение 0;

«portrait-primary» значение 90; «landscape-secondary» значение 180; « portrait-secondary » значение 270

Настройка параметров окна браузера

Связь с автором: Jabber: wirl@exploit.im

Для начала, посмотрим все основные параметры окна браузера в Linken Sphere наглядно на скриншоте для более лучшего понимания (скриншот честно украл и доработал):

https://prnt.sc/lz7r9g

Рассматривать настройку будем с двух вариантов:

1) **Полноэкранный режим**, когда мы разворачиваем в браузер на весь экран.
2) **Оконный режим**, когда браузер занимает лишь какую-то часть экрана. На скриншоте для примера как раз показан именно этот вариант.

Window.outerWidth – данный параметр показывает ширину окна вашего браузера, включая полосу прокрутки, панель инструментов и.т.д.

Window.outerHeight – данный параметр показывает высоту окна вашего браузера, включая панель инструментов, URL строку, вкладки браузера, область загрузки и.т.д.

На скриншоте выше отлично продемонстрированы эти параметры, и то чем они отличаются от других. Если браузер у нас в полноэкранном режиме, то мы можем указать точные значения. Если же мы работаем в оконном режиме, то значений может быть огромное множество, главное чтобы значения были «согласованы» с другими параметрами (innerWidth, client.Width, innerHeight, clientHeight, screenLeft, screenTop, screenX,screenY) . Самый лучший и самый простой вариант получить значения для оконного режима, это использовать скрипт, что я прикрепил к теме.

В полноэкранном режиме данные параметры соответствуют параметрам «availWidth» и «availHeight»

Window.innerWidth и body.clientWidth – данные параметры показывают значение ширины рабочей области браузера, проще говоря всю ширину в пикселях, на которую у вас прогружаются сайты, исключая ширину скролла, панели задач, если она размещена справа и других элементов, которые сужают данную ширину. Объединил данные параметры, так как в рамках нашей статьи они будут совпадать.

Window.innerHeight и body.clientHeight - данные параметры показывают значение высоты рабочей области браузера, проще говоря всю высоту в пикселях, на которую у вас прогружаются сайты, исключая высоту горизонтального скролла, высоту области вкладок, высоту URL строки в браузере и других элементов, которые уменьшают данную ширину. Объединил данные параметры, так как в рамках нашей статьи они будут совпадать.

Данные параметры по сравнению с остальными параметрами самые динамичные и непредсказуемые. Даже в полноэкранном режиме помимо outer.Width/Height влияет куча других окон.

Например, в **Google Chrome** влияет параметр настроек внешнего вида браузера («Показывать панель закладок»), отображается ли панель загруженных файлов в браузере (Пример: http://prntscr.com/lzd3r5) и др.

В **Firefox** влияют настройки в разделе «Customize» (https://support.mozilla.org/en-US/kb/customize-firefox-controls-buttons-and-toolbars?redirectlocale=en-US&redirectslug=Navigation+Toolbar+items). Если конкретно, то параметры Toolbars (Меню, закладки, Заголовок), параметр «Density».

И так в каждом браузере разные настройки влияют на данные значения.

Связь с автором: Jabber: wirl@exploit.im

В оконном режиме, помимо этих параметров, влияют еще и параметры «screenLeft, screenTop, screenX,screenY, outerWidth/Heght).

В любом режиме влияет параметр devicePixelRatio, но о нем подробнее ниже. Опять же, самый лучший и самый простой вариант получить значения – это использовать скрипт.

В таблице я дам варианты настройки для разных разрешений экрана в полноэкранном режиме при настройках браузера по умолчанию.

https://docs.google.com/spreadsheets/d/12KM12QLMdwdmBKDuxlM-uh31WZNMQ6hdlIz_QnipAVg/edit?usp=sharing

window.dexicePixelRatio – данный параметр показывает отношение размера физического пикселя к логическому. Проще говоря, в рамках рассмотрения наших типов браузера, это параметр масштаба страницы. По умолчанию он 100% и параметр равен «1». Если мы увеличиваем масштаб страницы или уменьшаем, то данный параметр меняется. Увеличили масштаб страницы на 125%, параметр изменился на «1.25», уменьшили страницу до 90% , параметр изменился на «0.9».

Уточнения: изменения данного параметра влияют на параметры «Window.innerWidth, body.clientWidth, Window.innerHeight, body.clientHeight) что в полноэкранном режиме работы, что в оконном режиме работы.

Для естественного изменения параметра нужно использовать шаг увеличения или уменьшения как в реальном браузере. Пример:

Браузер Firefox значения масштаба:

«50%», «60%», «70%», «80%», «90%», «100%»,«110%», «120%»,«130%», «140%» и.т.д. (Шаг 10%)

Браузер Chrome значения масштаба:
«33%»,«50%»,«67%»,«75%»,«80%»,«90%»,«100%»,«125%»,,«150%»,,«175%»,,«200%»,,«250%», «250%» и.т.д. (Шаг динамический).

И так для каждого браузера.

Еще одна тонкость со значениями данного параметра. Возьмем браузер Chrome:

100%- значение параметра «1»; 110% значение параметра не «1.1», а «1.100000023841858»; 125% значение параметра «1.25». Т.е. не всегда значение может быть точь-в-точь; в разных браузерах по разному

Последняя тонкость: размер рабочего окна, уменьшается или увеличивается НЕ РОВНО на значение devicePixelRatio. Т.е. если мы увеличиваем масштаб на 25%, это не значение что высота рабочей области браузера уменьшится РОВНО на 25%. Значения в процентном соотношении будут иные.

window.screenLeft и **window.screenX** – данные параметры показывают в пикселях, насколько окно браузера в оконном режиме сдвинуто вправо от первого пикселя.

window.screenTop и **window.screenY**– данные параметры показывают в пикселях, насколько окно браузера в оконном режиме сдвинуто вниз от первого пикселя.

Связь с автором: Jabber: wirl@exploit.im

На скриншоте я наглядно показал эти параметры. Объединил данные параметры, т.к. они совпадают в рамках нашей статьи. В браузерах Chrome, Opera, Edge используются все эти параметры. В браузерах Mozilla Firefox используются только эти параметры: ScreenX и ScreenY.

Если в браузере полноэкранный режим и панель управления находится внизу или справа, то данные параметры равны «0».

Если используется полноэкранный режим в браузере и панель управления находится слева или сверху, то значения данных параметров равны ширине или длине панели управления.

Если используется оконный режим работы браузера, то параметры будут зависеть от того, насколько сдвинуты они от левого первого пикселя экрана и верхнего первого пикселя экрана. Для расчета данных параметров лучше всего использовать скрипт. Данные параметры не имеют прямой зависимости с параметрами Outer.Width/Hegiht, innerWidthHeight, т.е. правило «Ширина Экрана= screenLeft/ screenX+Outer.Width» НЕ РАБОТАЕТ, так как параметров, отвечающих за правую и нижнюю сторону экрана нет, а, следовательно, значение «outer.Width» при значении screenLeft/ screenX 50 px может быть как и 600 px, так и 500 px, так и 900 px –все зависит от того, насколько мы «растянем» окно браузера по ширине. Данное правило относится также и к высоте экрана.

window.pageXOffset – данный параметр показывает насколько страница прокручена вправо (по вертикали в пикселях) с помощью полосы прокрутки относительно левого верхнего окна.

window.pageYOffset – данный параметр показывает насколько страница прокручена вниз (по горизонтали в пикселях) с помощью полосы прокрутки относительно левого верхнего окна.

Для более лучшего понимания, посмотрите скриншот.

В полноэкранном режиме параметр **window.pageYOffset** динамический, потому что почти на любом крупном популярном сайте, мы прокручиваем страницу вниз, редко когда сайт полностью умещается в рабочее окно, главная страница поиска google не в счет :) Поэтому данный параметр лучше всего просто НЕ подменять.

В полноэкранном режиме параметр **window.pageXOffset** в большинстве своем равен «0», так как сайты адаптированы под разные разрешения экрана, а скролл в бок – дико неудобный. Но если у нас оконный режим браузера, то он также может быть, в зависимости от сайта и размера окна браузера.

Поэтому задавать постоянные значения данным параметрам нет никакого смысла. Как по мне, если уж и брать подмену, то единственно-возможный смысл, делать его случайным в пределах каких-либо значений.

Настройка плагинов в конфиге

Про плагины я подробно рассказал в 1 разделе статьи. В новых версиях Chrome, Firefox, Opera, Edge остались только встроенные плагины и 1 плагин, который можно установить - Adobe Flash Player. Существует два вида программы Adobe Flash Player: Adobe Flash Player ** NPAPI – под браузер Firefox. Adobe Flash Player ** PPAPI– под браузер Opera/Chrome.

Сейчас мы подробно разберем, как настраивать плагины и какие вариации можно сделать.

Firefox имеет два встроенных плагина по умолчанию «Widevine Content Decryption Module» и « OpenH264 Video Codec provided », но данные плагины не показываются при запросе.

Связь с автором: Jabber: wirl@exploit.im

В Firefox один лишь плагин, который можно добавить – это Flash. **Тонкости**: при установке в системе Flash, по умолчанию стоит настройка «Ask to Activate», при такой настройке Flash показывается лишь при запросе сайта; плагин в чекерах не светится; если стоит параметр «Always Activate», то физический Flash и плагин светится. Поэтому устанавливая Flash в систему, мы можем и без антидетекта уникализировать данный отпечаток.

С антидетектом у нас возможны два варианта: либо добавляем плагин Flash'а, либо нет. Если добавляем, то у нас есть различные вариации в виде версий Flash. Это дает нам возможность в разных конфигурациях, делать разный плагин Flash, а не добавлять один и тот же. Ниже будет ссылка на таблицу под тип браузера Firefox как выглядит настройка Flash в плагине, а также список различных версий. Напоминаю, что в сфере плагины настраиваются в «Extended session settings».

https://docs.google.com/spreadsheets/d/1BPCD97WmsiSsHoFDZ3MJjtbbvtqBMgpJfXpF4SGnjc0/edit?usp=sharing

Google Chrome по умолчанию имеет 4 плагина, некоторые из них можно включить/отключить; единственный плагин, который можно добавить –это также Flash.

Параметры плагинов по умолчанию статичны; они не меняются. У плагина Flash есть же параметры, которые меняются в зависимости от версии плагина и в зависимости битности системы: 32-bit; 64-bit. Подробнее о плагинах по умолчанию:

Chrome PDF Plugin и **Chrome PDF Viewer** – эти плагины отвечают за PDF документы в Chrome и позволяют, например, открыть PDF прямо в Chrome онлайн, без скачивания файла на компьютер. Данные плагины связаны; так что вы либо добавляете оба плагина в конфигурацию, либо ни один из них. Включить/выключить в обычном браузере можно в Расширенных настройках--> Настройки контента --> PDF документы.

Widevine Content Decryption Module – данный плагин отвечает за запрет копирования аудио и видео контента правообладателем. С 57 версии Chrome плагин нельзя отключить. Но при этом я не раз встречал в системах и конфигах, что данный плагин не светился, хотя версии Chrome были одни из самых последних.

Native Client – плагин отвечает за запуск некоторых онлайн игр и приложений. Отключить нельзя, поэтому данный плагин добавляем 100%.

Вот табличка для настройки плагинов в сфере для типа браузера Google Chrome и вариации настроек с Flash:

https://docs.google.com/spreadsheets/d/1BPCD97WmsiSsHoFDZ3MJjtbbvtqBMgpJfXpF4SGnjc0/edit?usp=sharing

В браузере Opera все идентично Chrome, кроме некоторых тонкостей.

1) Различаются названия плагинов отвечающие за PDF. Вместо «Chrome PDF Plugin» значение «Chromium PDF Plugin»; вместо «Chrome PDF Viewer» значение «Chromium PDF Viewer».
2) Нет плагина Native Client.
3) Плагин «News feed handler». Отвечает за фиды, т.е. за получение контента с сайта прямо в браузер с помощью протокола RSS .По умолчанию активирован. Поэтому добавляем данный плагин.

Связь с автором: Jabber: wirl@exploit.im

Вот табличка для настройки плагинов в сфере для типа браузера Opera и вариации с Flash:

https://docs.google.com/spreadsheets/d/1BPCD97WmsiSsHoFDZ3MJjtbbvtqBMgpJfXpF4SGnjc0/edit?usp=sharing

Настройка списка шрифтов

Все антидетекты на рынке позволяют подменить отпечаток шрифтов. В конфигах большинства антидетектов содержится список шрифтов. Сфера позволяет удобно редактировать список шрифтов в конфиге или создавать с нуля, загружая названия шрифтов из файла.

В самой системе без антидетекта можно очень просто редактировать список шрифтов. Для этого нужно зайти в панель управления -->оформление и персонализация-->шрифты.

Там можно добавить новые шрифты, предварительно скачав их, удалить имеющиеся шрифты. Совершая такие манипуляции, мы изменяем наш список.

В каждой системе за счет установленных различных программ и прочих факторов, список шрифтов и количество шрифтов будет разное. Но есть базовые шрифты для каждой версии ОС Windows.

Список базовых шрифтов и их стилей для Windows 7

https://docs.microsoft.com/en-us/typography/fonts/windows_7_font_list

Список базовых шрифтов и их стилей для Windows 8

https://docs.microsoft.com/en-us/typography/fonts/windows_8_font_list

Список базовых шрифтов и их стилей для Windows 10

https://docs.microsoft.com/en-us/typography/fonts/windows_10_font_list

От этих базовых шрифтов можно отталкиваться при создании своего списка. Некоторые уточнения: все ссылки семейства шрифтов кликабельны. Внутри можно найти информацию, какие ОС Windows и программы используют данное семейство шрифтов. Не нужно в списке указывать все стили шрифта, можно указывать только семейство шрифтов. Проверяйте семейства шрифтов на сайте; например семейство «Wingdings» на самом деле содержит 3 шрифта.

Вот отличный список для создания своего списка шрифта. В нем указано большое количество шрифтов и какие ОС Windows и программы их используют.

https://docs.microsoft.com/en-us/typography/font-list/

Настройка подмены WebRTC и .MediaDevices.enumerateDevices

.MediaDevices.enumerateDevices – данная функция позволяет получить список всех устройств (аудиоустройства и видеоустройства системы, USB-камеры, микрофоны и.т.д.). Можно получить deviceID данных устройств, названия устройства и тип устройства.

Функция в Linken Sphere имеет настройки: **«True»** - функция включена, но параметры не подменяется. **«False»** - функция выключена, **«Fake»** - функция включена; параметры подменяются.

В наших типах браузеров используем только параметр «Fake».

Связь с автором: Jabber: wirl@exploit.im

Перейдем к настройке WebRTC. Используем подмену на всех типах браузера, что мы разбираем с вами сегодня. Рассмотрим некоторые тонкости.

1)**Галочка IPv6**. Данную галочку включать, если на вашей системе идет утечка ipv6. Проверить можно здесь: https://browserleaks.com/ip (пункт «IPv6 Address»)

2) **Внешний (Публичный) IP** в WebRTC. Тут все просто: внешний IP совпадает с IP вашего носка или туннеля. Но при работе с логами я встречал еще вот такой необычный подход. Суть его заключается в том, что внешний IP ставится IP системы пользователя. Да, при этом, чекеры будут показывать, что это неправильно, но такой подход имеет место.

3) **Внутренний (Локальный) IP в WebRTC**. Тут тоже вроде как все просто: есть диапазоны локальных IP, которые могут использоваться.

10.0.0.0 — 10.255.255.255

172.16.0.0 — 172.31.255.255

192.168.0.0 — 192.168.255.255

Но опять же есть тонкости. Напомню вот эту табличку:

https://docs.google.com/spreadsheets/d/1GySRwS_QAmvPSJEDxYcsGnz_7Vu_mtj0nn_RvY4wgl4/edit?usp=sharing

Так вот, есть столбец Default Local IP. Это локальный IP роутера по умолчанию, по которому как раз можно попасть в его настройки. Поэтому эти IP лучше не ставить при настройке конфига.

Следующая фишка заключается в работе с логами и локальным IP. В первой части мы пытались узнать примерный бренд роутера, а в идеале его модель. Так вот в некоторых случаях, мы можем предположить примерный локальный IP адрес.

Вообще откуда берется этот локальный адрес в WebRTC в вашей системе? Большинство роутеров имеют в настройках DHCH-сервер. DHCH-сервер назначает каждому устройству, которое подключается к роутеру, локальный IP. Обычно настройки параметров DHCH примерно такие, в зависимости от бренда и модели роутера: Начальный IP, конечный IP и время, на которое выдается IP адрес. Возьмем, к примеру, что у роутера такие настройки:

Начальный IP: 192.168.0.2

Конечный IP: 192.168.0.100

Время: 1440 мин (24 часа)

Мы подключаем к роутеру наш ноутбук, DHCH-сервер дает ему локальный IP: 192.168.0.2 на 24 часа; Мы подключаем наш мобильный телефон, DHCH-сервер дает ему локальный IP: 192.168.0.3 на 24 часа; мы подключаем к роутеру наш холодильник с Wi-Fi, DHCH-сервер дает ему локальный IP: 192.168.0.4 на 24 часа и.т.д. Допустим, прошло 12 часов, свет вырубили, и роутер перезагрузился; и первый к нашему роутеру подключился холодильник. Теперь ему DHCH-сервер дает ему локальный IP: 192.168.0.2 на 24 часа; потом мобильник подключился - ему DHCH-сервер дает ему локальный IP: 192.168.0.3 на 24 часа; потом уже ноутбук подключился - ему DHCH-сервер дает ему локальный IP: 192.168.0.4 на 24 часа.

Связь с автором: Jabber: wirl@exploit.im

Таким образом, на данном примере видно, что локальный IP является динамическим параметром и может меняться в тех пределах, которые указаны в настройке DHCH-сервера в роутере.

Зная бренд роутера и примерную модель, можно посмотреть этот диапазон IP,и выставить в логе примерный локальный IP. Опять же, на примере выше, у владельца роутер D-Link; мы определили начальный и конечный IP. У владельца, скорее всего, имеется помимо компьютера еще 2-4 устройства, которые подключаются к роутеру (например, телефон и телевизор). Следовательно, спокойно ставим локальный IP «192.168.0.2» или «192.168.0.3» или «192.168.0.4» или «192.168.0.5». В интернете можно найти эмуляторы большинства популярных роутеров и в настройках посмотреть базовый диапазон IP; в таблице я также добавил к некоторым моделям начальный и конечный IP.

Настройка подмен в конфиге.

Хоть и это не относится к настройке реального конфига, но по некоторым особенностям я пройдусь. Про все отпечатки, которые подменяет сфера, можно найти кучу разной информации, поэтому 10 раз одно и тоже я описывать не буду.

Насчет использования подмен: советую в любом из наших типов браузера использовать **все подмены**, но с некоторыми тонкостями.

1) **Enable Flash -** включает Flash. Включать flash без необходимости не рекомендует каждый автор антидетекта, так как это является дополнительной вариации вас задетектить. Насчет использования Flash могут посоветовать такие варианты, неважно создаем ли мы конфиг для работы с логом или конфиг какого-либо из типов браузера:
   А) Добавить Flash в плагины конфига, но при этом физический flash(enable flash) оставить выключенным. Тут получается интересная ситуация, по плагинам он у нас как есть, но при этом физической версии нет.
   Б) Добавить Flash в плагины конфига и включить физический flash(enable flash). Минусы этого варианта я описал выше.

Еще один момент, в некоторых антидетектах можно настроить параметры Flash, поэтому если есть такие настройки, и вы решили использовать Flash, обязательно не забываем их настраивать (такие параметры как ОС, язык, разрешение экрана, версия Flash и другие)

2) **Подмена Canvas**. Включаем данную подмену, но сейчас я напишу варианты, когда данную подмену можно попробовать отключить в наших типах браузера.

Их всего две: А) Когда создаваемый конфиг имеет такой же тип браузера как и антидетект, т.е. Linken Sphere написан на базе Chromium, следовательно, если вы создаете конфигурацию Chrome, то допускается как вариант отключить подмену. Второй вариант чуть похуже: это когда браузер сделан в оболочке Chromium. В нашем случае это тип браузера: Opera.

3) **Подмена AudioFingerprint.** Подмену само собой обязательно включаем. Но у audio также есть параметры (http://prntscr.com/lyqeto). Некоторые из них можно изменить в системе, поэтому ниже будет информация для размышления: как вариант помимо отпечатка их также подменять. Например, первый параметр на скрине (ac-sampleRate) изменить очень просто: для этого нужно в настройках вашего воспроизводящего устройства по умолчанию поменять Default Format (https://prnt.sc/lyqgop).

Связь с автором: Jabber: wirl@exploit.im

Советы, Фишки, лайфхаки с использованием антидетекта Linken Sphere и при работе с логами.

1. Установка Linken Sphere: Виртуальная машина или Основа?

Ставить на Виртуальную машину или на Основную? Также очень популярный вопрос. Опять же, можете найти для себя лучший вариант.

Linken Sphere на Основной Машине

Плюсы:

1) Удобство работы
2) Не требует больших ресурсов ОЗУ, меньше нагружает компьютер по сравнению с использованием виртуалки, например на Win 10 x64, особенно если для безопасности отключен файл подкачки и PC не сильно мощный.
3) Если вдруг существует в мире какой-либо детект или детект виртуалки, о котором никто еще не знает, то, это будет, несомненно, плюсом, по сравнению с использованием антидетекта на виртуальной машине.

Объясню подробнее, что я имел ввиду: Почти любой антидетект в мире, если не подменяет какой-либо параметр, то он, скорее всего, берется с вашей системы, либо просто отключен.

4) Безопасность. Не берусь утверждать на все 100%, но с точки зрения безопасности, в цепочки анонимности, возможно, данный вариант хуже, чем использовать Linken Sphere на Виртуальной машине или сервере.

Linken Sphere на Виртуальной машине

Тут все же наоборот, и минусы становятся плюсом, а плюсы – минусами.

3 вариант использования: некоторые используют Linken Sphere на выделенном сервере, что тоже, по своему, интересный вариант, который имеет некоторые плюсы с тех двух вариантов выше.

2. Какой тип конфигов лучше использовать для вбива. «Хорошие» варианты использования конфигов для разных типов ОС.

Собственно с учетом того, что данный Антидетект написан на исходниках движка Chromuim, идеально использовать конфигурации с **<u>браузером Chrome</u>** и браузеры на основе платформы Chromium.

Если Linken Sphere стоит у вас на ОС Windows, то «хорошие» варианты конфигов:

1) Win XP, 7,8, 10 + Chrome
2) Win XP, 7,8, 10 + Opera
3) MAC OS + Chrome
4) Win 10 + Edge (Совсем недавно Microsoft анонсировали замену движка на Chromium)

Мобильные варианты:

1) Android +Chrome
2) Windows Phone + IE (Internet Explorer)
3) Iphone, Ipad+ Chrome

Если Linken Sphere стоит у вас на ОС MAC X, то «хорошие» варианты конфигов:

1) MAC OS + Chrome
2) MAC OS + Safari
3) Win + Chrome

Мобильные варианты:

1) Iphone, Ipad + Chrome
2) Iphone, Ipad + Safari

Конечно, можно использовать любые конфиги на любых ОС, но эти предпочтительнее в силу того, что ОС и/или платформы совпадают.

3. Лайфхак: Использование «Нестандартных» конфигов при вбивах

Как хорошую и необычную альтернативу тех, вариаций, что я написал выше, может быть использование «нестандартных» конфигов. В моем понимании, нестандартные конфиги, это те системы, которые не распространены в целом и которые редко используются для вбивов. Для сферы и некоторых других АД, критерием еще может быть то, что данные конфиги не найти в конфигшопах. Приведу примеры таких конфигов: Xbox One, PlayStation 4, Blackberry, PlayBook, Kindle и др. Сложно, конечно, представить холдеров, вбивающих с Playstation 4 или PlayBook, но тем не менее данные варианты имеют место быть в некоторых темах и как одним из факторов «нестандартного» вбива.

Связь с автором: Jabber: wirl@exploit.im

Как получить данные конфиги для сферы? Тут вариант только один – сделать самому. Прочитав полностью мануал, вам будет более менее понятно как делать конфиги. Проблема лишь в том, что где взять все данные(UserAgent, WebGL, WebRTC, Window.Screen, Window.Navigator и.т.д) по этим устройствам? Тут все очень просто) Либо посмотреть на реально устройстве по всем нужным чекерам, либо взять с конфигов другого антидетекта.

4. Использование инструмента «Web Emulator»

Web Emulator – это инструмент в Linken Sphere, который позволяет в автоматическом режиме посещать список сайтов, имитируя поведения человека. Этот инструмент полезен тем, что автоматизирует процесс получения cookies, тем самым снижая наши временные затраты на рутинную работу, т.е. вводишь список сайтов, включаешь эмулятор, и вауля, у нас уже браузер cookies различных сайтов.

На практике данный инструмент очень полезен, т.к. антифрод системы шопов вполне могут собирать и анализировать ваши cookie файлы. Тем самым используя правильно данный инструмент, мы будем более похожи на обычного пользователя.

По настройкам данного инструмента (Скриншот: https://prnt.sc/jkvy3p)

Ставим галочки на Disable popups(отключаем всплывающие окна) и Enable alert after complete (Включаем оповещения после завершения работы Web Emulator). **MaxVisited Page** – это то, сколько максимум страниц на каждом сайте будет открыто. Тут уже каждый сам решает сколько ставить, я бы рекомендовал от **3-4** до **12-30**. Max time on page, min – я бы рекомендовал ставить от **30 секунд** до **2 минут**.

Start delay – данный пункт отвечает за задержку (в минутах) перед началом включения самого эмулятора. Тут уже на ваше усмотрение.

Каждый сайт нужно указывать с новой строки и с http[s]://.

Насчет списка сайтов. Я бы рекомендовал каждому составить свой список сайтов для обхода в зависимости от страны вашего вбива (в моем случае это USA). В свой список я бы собрал таких сайтов штук 30-40, для того чтобы была возможность чередовать различные сайты, а не каждый раз обходить одни и те же. Например, ТОП сайтов по ALEXA RANK: (https://www.alexa.com/topsites). Там вы можете выбрать ТОП 500 сайтов по различным странам, узнать среднюю глубину просмотра страниц, среднюю длительность нахождения пользователей на сайте за последние 3 месяца.

5. Детект Social Media Login

http://www.tomanthony.co.uk/tools/detect-social-network-logins/

Вот обычный паблик пример демонстрации того, что сайты спокойно могут увидеть залогинены ли вы в популярных социальных сервисах. Поэтому для того чтобы больше быть похожим на реального ПК пользователя, нужно делать аккаунты в популярных сетях и логинится в них непосредственно перед вашей работой (ну либо покупать готовые аккаунты). Самые популярные сервисы: Facebook, Twitter, Gmail, Youtube, Google+, Instagram, Pinterest, Battle Net, Xbox, PSN, Tumblr и др.

Данное правило относится также и к логам. Смотрим в аккаунтах, какие популярные социальные сервисы есть у нашего пользователя, авторизуемся в них (если автоматически не попадаем с помощью наших кук) и только потом переходим на нужные нам сайты.

Теперь перейдем к работе с логами.

По всем параметрам все логи можно разделить по степени важности (по убыванию важности):

1. Лог, который имеет набор стандартных сервисов в комплекте с БА и криптобиржами.
2. Лог, который имеет набор стандартных сервисов таких как Paypal , Amazon , различные шопы
3. Лог, который имеет пару сервисов/не интересные небольшие шопы; cookie файлы данного лога не вызывают у нас особого восхищения.
4. Бесполезный лог

Для новичков в первую очередь будет совет не брезговать логами 2 и 3 категории, относится к ним с такой же серьезностью как и к логам к первой категории для того чтобы получить опыт, набить свою руку, технически научиться правильно использовать лог и выбранный инструмент для его обработки.

На форумах часто раздают отработанные логи разных категорий; данный вариант также является хорошим вариантом старта работы с логами.

Связь с автором: Jabber: wirl@exploit.im

Определить степень важности можно по Логинам/Паролям лога, авто заполнениям, да иногда даже по заставке на рабочем столе можно отличить лог школьника-задрота от хорошего лога.

С опытом, в зависимости от ваших навыков, финансового положения вы будете сами определять на раз-два, с каким логом работать кропотливо, с каким не очень, а какой вообще стоит выкинуть и не тратить свое время.

Отработка логов в Linken Sphere очень удобна тем, что, используя разные вкладки, можно отрабатывать несколько логов одновременно, что также экономит время.

Отработка лога может быть комплексной, либо отработка на один конкретный запрос.

Часто бывает, что новички отрабатывают логи всего лишь на 1 запрос, например на Paypal, а если уж отработать не получается, то расстраиваются и выкидывают лог. Это плохой подход к работе, так как с ним не получишь нормального профита и знаний; если у вас много времени и мало опыта, отрабатывайте лог по полной, набивайте руку.

Перейдем к интересным советам, лайфхакам при работе с логами.

1. Определите ваш «вектор атаки». Проще говоря, вам нужно понять, где у холдера есть деньги , какими средствами оплаты он пользуется чаще всего. Посмотреть популярные средства оплаты холдера можно в Amazon, Paypal, Ebay и.т.д) . После того как вы определили, где деньги , нужно попытаться узнать сколько денег хранится у холдера, если возможно. Для этого нужно попасть в онлайн-банкинг, глянуть стейтманы и.т.д
2. Сервисы операторов, такие как AT&T, Verizon и другие могут иметь функцию блокировки сим карты, утери телефона и прочих полезных штук, которые могут усложнить восстановление доступа холдера, а то и полной утери.
3. Всегда ставьте на почте холдера в спам-фильтры сообщения от нежелательных сервисов и шопов, с которых может прийти сообщения. Перенаправляйте нужные сообщения через фильтры на свою почту.
4. Проверяйте все облачные хранилища (Google drive, icloud, onedrive, dropbox и др.) Есть большая вероятность найти там Photo ID, Drive License, Credit Cards, Wallet Seed , 2FA и прочую полезную инфу.
5. Живите жизнью холдера. Собрав все информацию на холдера , почитайте его Facebook и другие соц. сети, посмотрите, куда он едет завтра , когда его нет дома , когда он в зале , когда кушает, с кем трахается , это поможет составить психологический портрет для психологического воздействия, а так же дать возможность выбрать наилучшее время атаки

К примеру, можно срочно вызвать холдера на работу, прислать венок с угрозами (Сказав, что это от мафии, страховая все вернет, не рыпайся пока мы тебя не обчистим). Важно понять суть данных мыслей, а фантазия в вариантах использования полезной информации может быть безгранична.

6. Оставляйте за собой бекдоры. Ставьте свои секретные вопросы на почты, подвязывайте свои 2FA , телефоны, резервные почты. Тогда с большой вероятностью холдер не сможет быстро и безболезненно вернуть свой аккаунт, а даже если вернет, то может не заметить ваш бекдор, которым вы можете опять воспользоваться.
7. Пароли. Множество холдеров пользуются однообразными паролями, поэтому, даже если в логе нет нужного сервиса, можно подобрать самому путем перебора
8. Активность. По письмам на почте можно определить самые популярные и самые свежие по дате сервисы, которые использует холдер. Из них выбрать нужные -те, которые вы умеете отрабатывать. Данные сервисы будут более лояльны по фроду, так как холдер часто пользуется ими, а, следовательно, они имеют более свежие cookie по сравнению с другими сервисами.

Связь с автором: Jabber: wirl@exploit.im

Связь с автором: Jabber: wirl@exploit.im