DOI 10.37882/2223–2966.2020.05.05

ЦИФРОВОЙ ИДЕНТИФИКАТОР ВЕБ-ОБОЗРЕВАТЕЛЯ НА ОСНОВЕ АНАЛИЗА ВРЕМЕНИ ИСПОЛНЕНИЯ JAVASCRIPT КОДА

DIGITAL WEB BROWSER IDENTIFIER BASED ON JAVASCRIPT CODE PERFORMANCE TIME ANALYSIS

V. Bessoltsev
R. Gilmullin

Summary. The digital identifier of a web browser as a means of identifying an Internet user.

Keywords: authentication, web browser, JavaScript, google v8 web engine.

Бессольцев Виталий Евгеньевич
К.т.н., преподаватель, Военно-космическая академия имени А. Ф. Можайского (г. Санкт-Петербург)
v.bessoltsev@gmail.com
Гильмуллин Рустам Менауирович
Старший инженер, Военно-космическая академия имени А. Ф. Можайского (г. Санкт-Петербург)
grustam@bk.ru

Аннотация. Цифровой идентификатор веб-обозревателя как средство идентификации пользователя сети интернет.

Ключевые слова: идентификация, веб-обозреватель, JavaScript, веб-движок google v8.

Введение

Современный мир, его глобальная экономика, социальные взаимодействия, научные исследования немыслимы без использования сети интернет. Для осуществления этих и многих других видов деятельности пользователи сети интернет как правило используют специальное программное обеспечение, такое как веб-обозреватель. Последние десять лет показали, что наряду с легитимной деятельностью в интернете активно стали появляться ресурсы пропагандирующие, продвигающие противоправные действия, такие как размещение информации экстремистского характера, информации порнографического характера, размещение персональных данных граждан Российской Федерации, клевета, торговля оружием и наркотическими веществами и многое другое.

Государство гарантирует гражданам защиту от указанных выше противоправных действий и принимает меры по ограничению доступа к незаконно распространяемой информации в целях защиты основ конституционного строя, нравственности, здоровья, прав, обеспечения обороноспособности страны, безопасности государства. В связи с тем, что большинство интернет ресурсов, занимающихся противоправной деятельностью, размещены на серверах за пределами Российской Федерации, то в целях защиты граждан применяется блокирование доступа к указанным ресурсам (рис. 1).

Тем не менее многие пользователи (потенциальные нарушители действующего законодательства Российской Федерации) ищут пути обхода блокировок. На данный момент существуют следующие методы:

1. Использование анонимных сетей, имеющих выход за пределы Российской Федерации, таких как Tor, VPN, P2P и т.д.;
2. Использование специализированных расширений для веб-обозревателей, шифрующих и перенаправляющих трафик через каскады прокси серверов, находящихся за пределами Российской Федерации;
3. Обход блокировки провайдера посредством смены DNS сервера;
4. Использование сайтов посредников (анонимайзеров).

Использование всех вышеперечисленных методов позволяет успешно обходить блокировки, выставленные Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций (Роскомнадзор), что несет угрозы обществу, описанные выше.

Методы деанонимизации пользователей сети интернет

Идентификация пользователей сети интернет является важной государственной задачей, для этого применя-

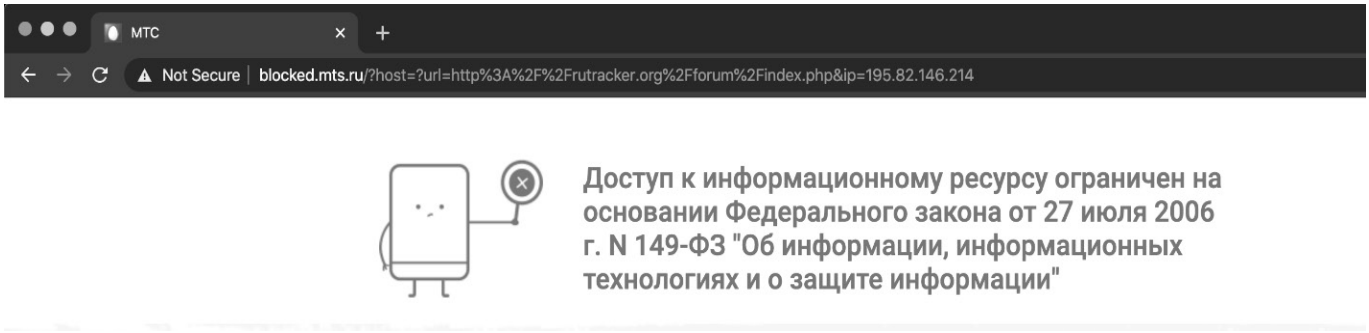

Рис. 1. Пример блокирования доступа к ресурсу, занимающимся противоправной деятельностью

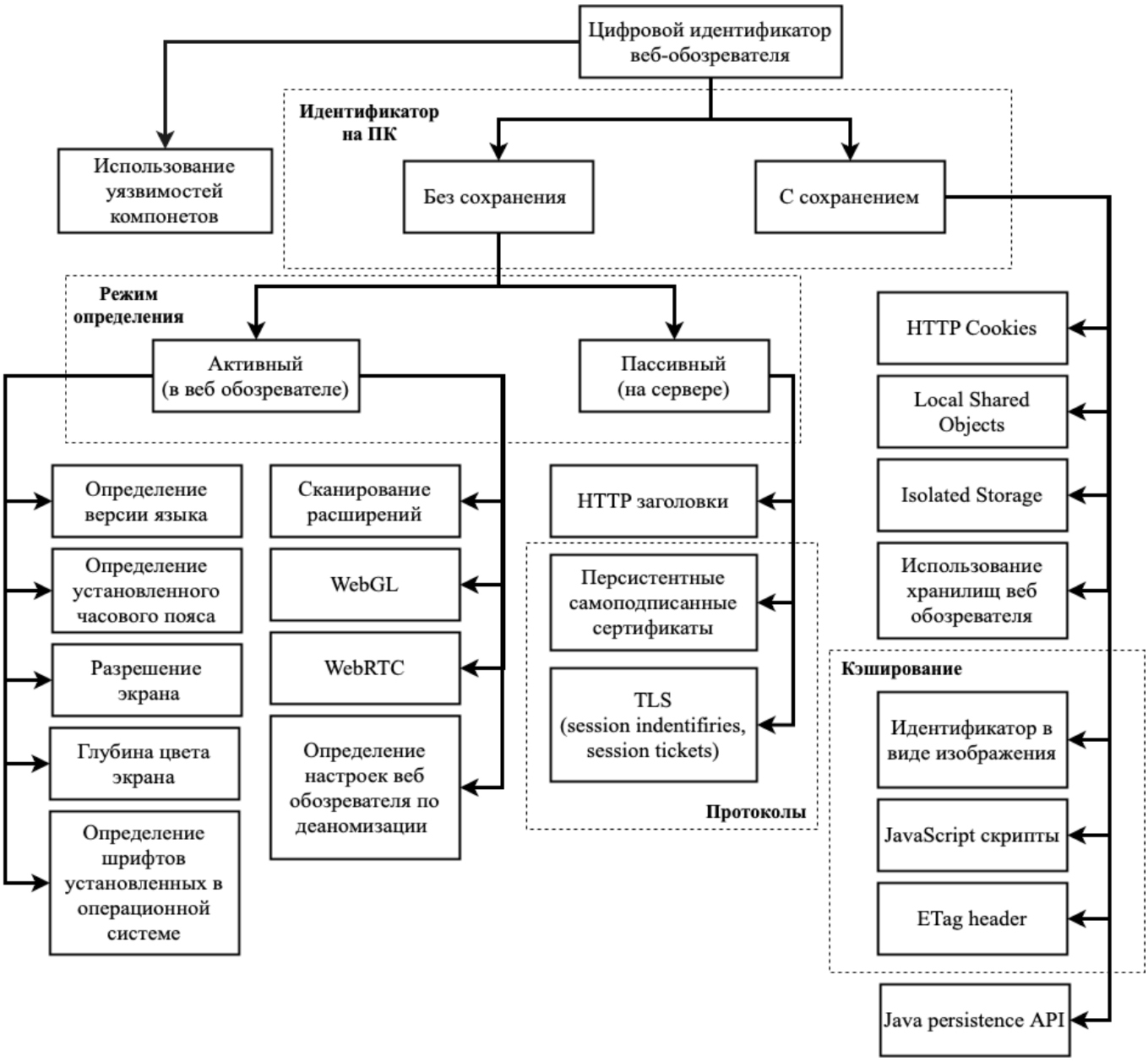

Рис. 2. Цифровой идентификатор веб-обозревателя

Таблица 1. Пример JavaScript функций

№ п/п	JavaScript функция	Описание	Противодействие
1	console.log(navigator.language);	Определение версии языка операционной системы	Активно применяется
2	x = new Date(); console.log(x.getTimezoneOffset() / 60);	Определение установленного часового пояса	Активно применяется
3	console.log(screen.width); console.log(screen.height);	Определение разрешения экрана	Активно применяется
4	console.log(screen.colorDepth);	Определение глубины цвета экрана	Активно применяется

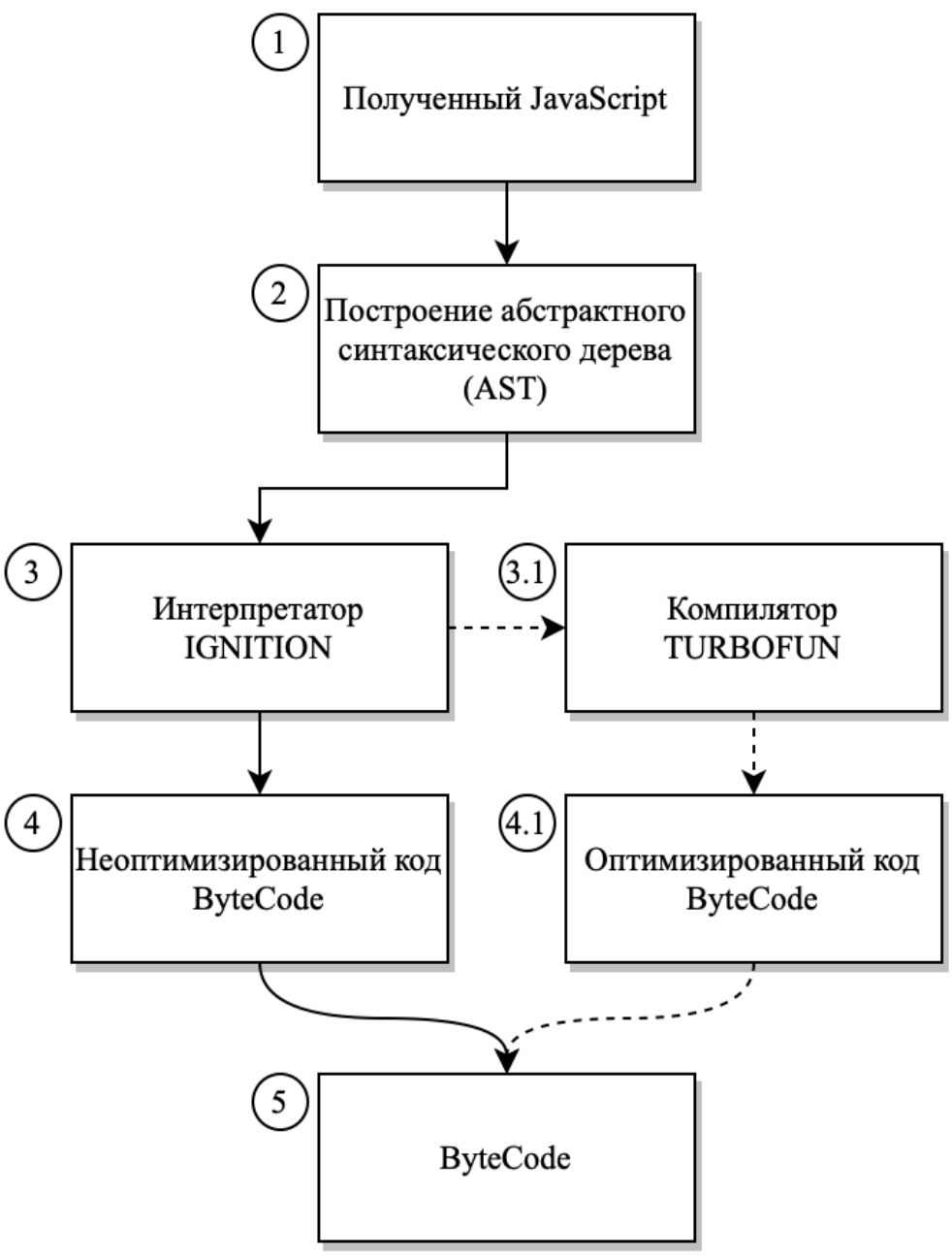

Рис. 3. Схема работы JavaScript движка Google V8

Таблица 2. Код для тестирования времени исполнения

```
1  for (var x = 0; x < countIteration; ++x) {
2  var start = new Date().getTime();
3  var someArray = new Array();
4  for (var i = 0; i < 1500; ++i) {
5  var el = document.createElement("div");
6  el.id = "testDiv";
7  el.style = "width: 100px; height: 100px; background-color: red;";
8  el.innerHTML = x;
9  document.body.append(el);
10 document.body.removeChild(el);
11 }
12 var end = new Date().getTime();
13 var time = end — start;
14 if (typeof (arr[x + 1])!= "undefined") {
15 arr[x + 1].push(time);
16 }
17 else {
18 arr[x + 1] = [x, time];
19 }
20 }
```

ется множество технических решений, такие как: глубокое инспектирование пакетов (Deep Packet Inspection), создание цифровых идентификаторов (ЦИ) (пользователя, вею-обозревателя) и т.д.

Вследствие того, что для посещения (с целью получения информации различного характера) ресурсов сети интернет, пользователи используют веб-обозреватели, то особое внимание необходимо уделить составлению цифрового идентификатора веб-обозревателя. ЦИ веб-обозревателя возможно получить посредством анализа параметров, указанных на рисунке 2, однако одновременное развитие средств противодействия получению ЦИ сформировало проблемную ситуацию, а именно выявление устойчивого к модификациям признака, идентифицирующего абонентский терминал среди множества потребителей контента интернет ресурсов.

Признаковое пространство (ПП) параметров веб-обозревателя формируется посредством выполнения JavaScript кода, а именно посредством вызова конкретных функций (таблица 1), которым в современных веб-обозревателях оказывается активное противодействие в виде оповещения пользователя об их использовании или принудительной отмене их выполнения.

Противодействие формированию ПП и как следствие созданию ЦИ ставит задачу расширения ПП получаемых параметров, признаков работы оборудования абонентских терминалов посредством веб-обозревателя.

Исполнение javascript в веб-обозревателях

JavaScript является неотъемлемой частью современных веб-приложений, позволяющий осуществлять как интерактивное взаимодействие с пользователем веб-ресурса, так и построение полноценного веб-приложения по архитектуре single page application (SPA). Блокирование работы (равно как отключение) JavaScript ведет к «выпадению» пользователя из современного веб-пространства, другими словами, это ведет к блокированию возможности использования веб-приложений.

Современные веб-обозреватели (на основе веб-обозревателя Chromium) для интерпретации JavaScript кода используют движок Google V8.

Для расширения ПП с целью идентификации веб-обозревателя необходим анализ особенностей функционирования внутренних компонентов JavaScript движка Google V8 (далее V8).

Алгоритм работы V8 (текущей версии 80+) схематично показан на рисунке 3, где отображены основные компоненты и их связь между собой.

Рассмотрим его:

этап 1. Происходит получение JavaScript кода (как с удаленного веб-ресурса, так и с локального);

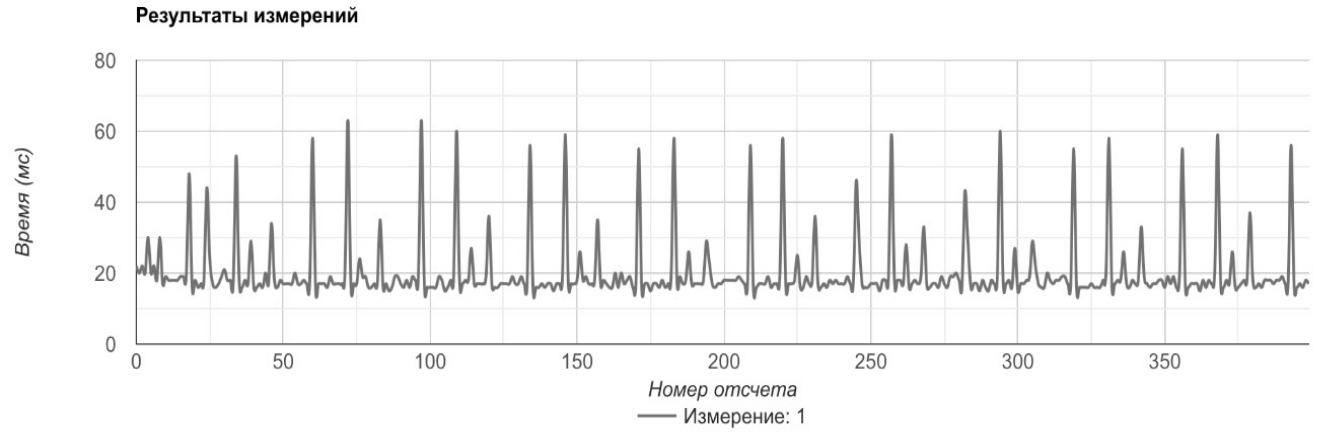

Рис. 4. Зависимость времени исполнение кода от номера отсчета
(Веб-обозреватель: chromium 71.0.3560.0, V8: 7.1.163)

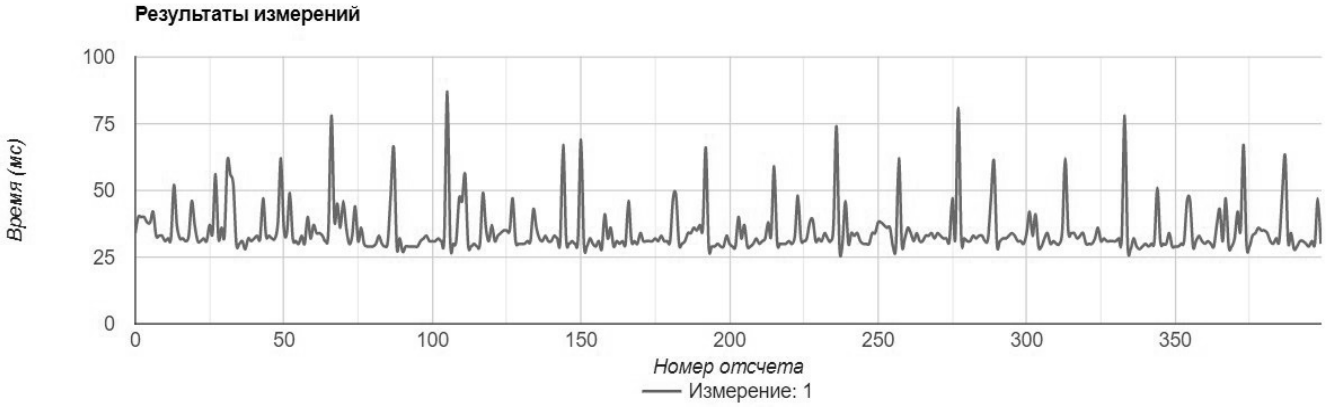

Рис. 5. Зависимость времени исполнение кода от номера отсчета
(Веб-обозреватель: chromium 62.0.3165.0, V8: 6.2.2)

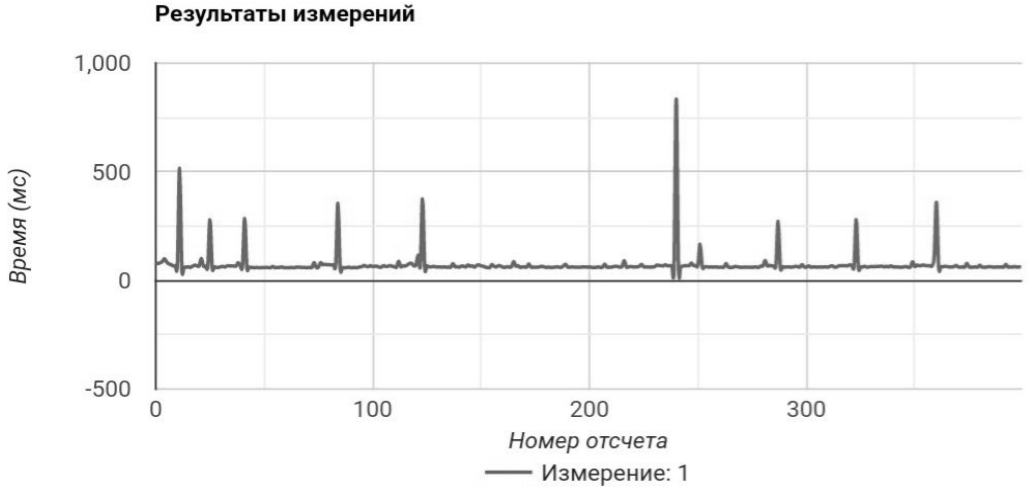

Рис. 6. Зависимость времени исполнение кода от номера отсчета
(Веб-обозреватель: chromium 80.0.3987.149, V8: 8.0.426.27)

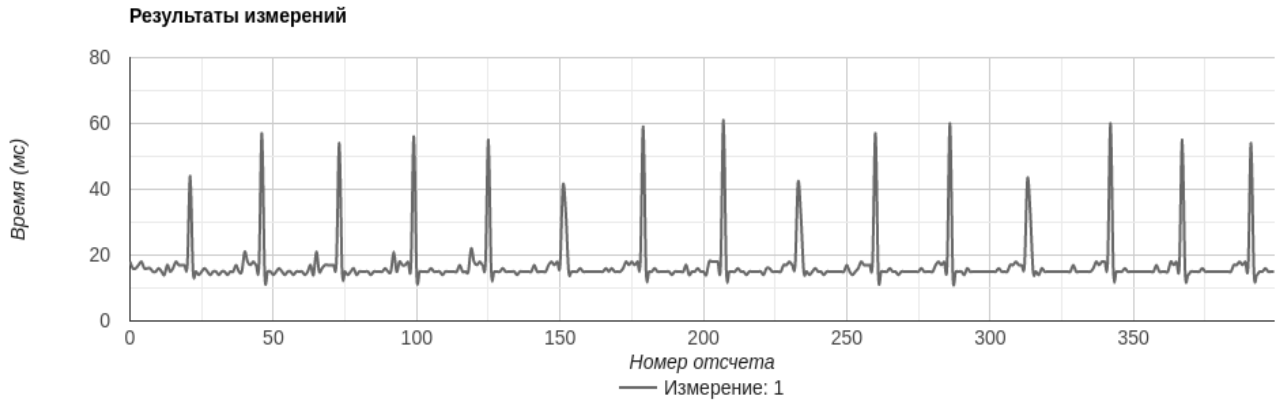

Рис. 7. Зависимость времени исполнение кода от номера отсчета
(Веб-обозреватель: chromium 80.0.3987.163, V8: 8.0.426.30)

этап 2. Для упрощения дальнейшей работы интерпретатора Ignition и компилятора Turbofun в модуле 2 происходит преобразование полученного JavaScript кода в абстрактное синтаксическое дерево (AST);

этап 3. После окончания работы этапа 2, AST передается интерпретатору Ignition, где на его основе генерируется байт код. В случае если интерпретатор выявляет в коде закономерности, которые по его алгоритму необходимо проверить на возможность оптимизации (например, циклическое повторение кода), то байт код для оптимизации передается в компилятор текущего времени исполнения Turbofun. В противном случае Ignition выдает неоптимизированный байт код, для его последующего преобразования в машинный код под текущую архитектуру системы.

Алгоритм работы V8 с момента своего появления до текущей реализации v.12.04 претерпел значительные изменения, что сказалось на значительном приросте производительности в интерпретации JavaScript кода. Для поиска закономерностей работы V8 различных версий был написан JavaScript код, приведенный в таблице 2.

Анализ времени исполнения JavaScript кода (рисунок 4–7) показал наличие закономерностей в необычно длительном исполнении кода, по-разному проявляющихся в различных версиях V8.

Вывод

Исследование особенности исполнения JavaScript кода в различных версиях V8 позволит выявить уникальные признаки работы веб-обозревателя, что в свою очередь позволит идентифицировать текущую версию JavaScript движка V8 и как следствие расширить признаковое пространство получаемых параметров функционирования веб-обозревателя.

ЛИТЕРАТУРА

1. Об изменчивых методах объекта Math в JavaScript. [Электронный ресурс] URL: https://habr.com/ru/company/ruvds/blog/489826/ (дата обращения 20.03.2020).
2. Федеральная служба по надзору в сфере связи, информационных технологий и массовых коммуникаций (Роскомнадзор) — Новости. [Электронный ресурс] URL: https://rkn.gov.ru/treatments/p459/p463/) (дата обращения 16.02.2020).
3. Федеральный закон от 27.06.2006 г «Об информации, информационных технологиях и о защите информации».
4. Идентификация абонентов анонимных компьютерных сетей посредством выявления уникальных параметров веб-обозревателя / К. В. Сазонов и др. // Труды Военно-космической академии имени А. Ф. Можайского.— СПб.: ВКА им. А. Ф. Можайского, 2018.— Вып. 665.— С. 99–111.
5. Approaches to optimizing v8 JavaScript engine. [Электронный ресурс] URL: https://cyberleninka.ru/article/n/approaches-to-optimizing-v8-javascript-engine/viewer (дата обращения: 1.04.2020).

Журнал «Современная наука: актуальные проблемы теории и практики»