https://doi.org/10.53656/math2024-5-1-ana

Science in Education
Научно-методически статии

ПОДХОД ЗА ОТКРИВАНЕ НА ПРОПУСКИ В СИГУРНОСТТА НА УЕББРАУЗЪРИ В ОПЕРАЦИОННИ СИСТЕМИ ЗА МОБИЛНИ УСТРОЙСТВА

Стоян Мечев

Висше военноморско училище „Н. Й. Вапцаров“ – Варна (България)

Резюме. В настоящото изследване се демонстрира, че е възможен достъп до сензорите на мобилен телефон чрез уеббраузър без знанието на потребителя. По този начин мобилните телефони са уязвими за атаки по странични канали[1]. Разработен е софтуер, който проверява достъпа до сензорите на смартфона и генерира протокол с резултати и препоръки.

След проучване на научните публикации по темата е направен технически експеримент, чрез който се доказва, че може да се осъществи достъп до различни сензори на изследваното устройство, без потребителят да е известен за това. Опитните резултати показват, че браузърите Chrome и Samsung Internet, работещи под Android, са уязвими за атаки по странични канали. Използвани са методите сравнителен анализ, обобщение и технически експеримент.

Ключови думи: Android; iOS; сигурност; уязвимост; браузър; уеббраузър; атака по странични канали

1. Въведение

През второто десетилетие на XXI век употребата на мобилни устройства стана широка практика, което ги превърна в неотделима част от ежедневния живот на хората по целия свят. Например по данни на Statista за 2022 г. има над 4,7 милиарда потребители на смартфони[2].

От една страна, използването на такова количество мобилни устройства предоставя огромен брой възможности за иновации и комуникация, но от друга страна, създава и множество предизвикателства в областта на сигурността.

Заплахите в онлайн средата, свързани с мобилните устройства, станаха по-сложни и разпространени. Част от тях са свързани с достъп до сензорите на мобилните устройства чрез JavaScript код, изпълняван в мобилен браузър.

В изследването са използвани методите сравнителен анализ, обобщение и технически експеримент.

2. Актуално състояние на проблема

В литературния обзор са включени предимно статии, индексирани в Scopus или Web of Science, публикувани през последните шест години, за да се повишат надеждността и актуалността на резултатите, получени в изследването. Стремежът на автора е да се спазят препоръките на Jennex (Jennex 2015).

Въпреки че изследването на Mehrnezhad и колектив излиза извън времевото ограничение на литературния обзор, за отбелязване е, че те за първи път показват как сигурността на потребителите може да бъде компрометирана чрез JavaScript код, изпълнен в браузър за мобилен телефон. Изследователите използват факта, че в съответствие със спецификациите на W3C мобилните уеббраузъри разрешават JavaScript код в уебстраница да получава достъп до данни от сензорите за движение и ориентация без разрешението на потребителя. Авторите разработват собствен код – TouchSignatures, който е в състояние да разграничи действията на потребителя при докосване (т.е. докосване, превъртане, задържане и мащабиране) и въвежданите ПИН кодове, което позволява на отдалечен уебсайт да научи дейностите на потребителя от страна на клиента. Атаката е приложима както за Android, така и за iOS (Mehrnezhad et al. 2016).

Mehrnezhad и Toreini изследват сензорите за мобилни устройства и политиките за разрешения, които Android, iOS и мобилните браузъри предлагат за тях. Фокусът на изследването е насочен към повишаване на осъзнатостта на потребителите относно рисковете за сигурността, които могат да възникнат при разрешаване на достъп до мобилните сензори. Резултатите показват, че познаването на сензорите не дава бързо отражение върху поведението на потребителите. Въпреки това натрупването на знания в тази област има силно влияние върху възприятията на потребителите в това отношение (Mehrnezhad & Toreini 2019).

Diamantaris и колектив проучват възможността за атаки на Android чрез WebAPI на HTML 5. В изследването се използва динамична система за анализ на приложения в реално време. Проучени са 183 571 от най-популярните уебсайтове през периода март – септември 2018 г.

Установени са 5313 уникални домейна, които достъпват поне едно мобилно WebAPI повикване; 35,89% от тях също така водят до достъп до сензори от страна на скриптове от трети лица, които идват от 11 домейна на второ ниво. Авторите предлагат таксономия на атаките през WebAPI (Diamantaris et al. 2020).

Изследвани са следните въпроси:

• До кои сензори на мобилен телефон имат достъп браузърите?

• Кой браузър е по-сигурен по отношение на достъпа до сензорите на мобилния телефон?

• Коя операционна система за мобилни устройства е по-сигурна по отношение на браузърите?

3. Ограничения на изследването

3.1. Изследвани операционни системи за мобилни устройства (ОСМУ)

Изследването на всички ОСМУ е нецелесъобразно, тъй като по данни на Statcounter[3] в края на 2021 г. пазарните дялове на ОС Android и iOS са били съответно 70,01% и 29,24%, или общо 99,25% от мобилните устройства на пазара са използвали една от двете операционни системи.

Версии на ОС Android. Като се отчетат хронологията на внедряване на различните версии на ОС Android (Raphael 2023) и пазарните им дялове в края на 2021 г., може да се направи заключението, че изследването на уязвимости за версии, по-ранни от Android 5.0 Lollipop, всъщност не са продуктивни за конкретизираната цел на изследването, тъй като общият пазарен дял на всички по-ранни версии е под 2%.

Версии на iOS. След прилагане на същите ограничения, а именно хронология на внедряване на iOS (Moreau 2021) и пазарен дял (фиг. 2), може да се направи заключението, че не е целесъобразно да се изследват версии на iOS, по-ранни от 10.0.

3.2. Изследвани устройства

Изследването е ограничено до смартфони, защото при тях има най-голям брой сензори, и второ, това е общо и за двата вида операционни системи.

3.3. Изследвани браузъри

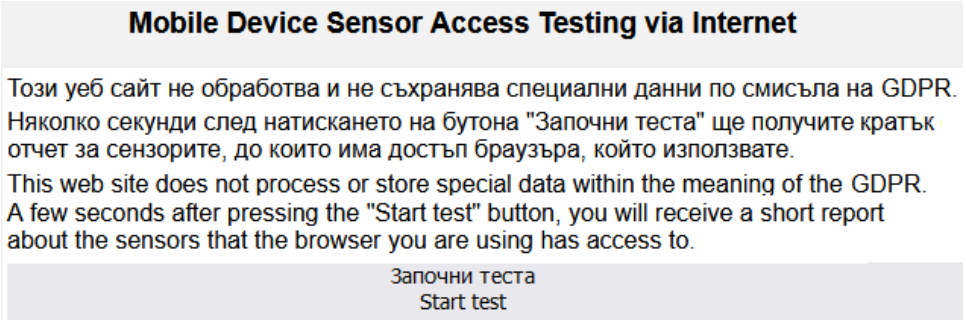
В това изследване са включени браузърите Chrome, Safari и Samsung Internet на база на техните пазарни дялове. По данни на Statcounter за края на 2023 г.[4], те са съответно: Chrome – 64,93%, Safari – 24,71%, и Samsung Internet – 4,48%. Трите приложения обхващат общо 94,12% от пазарните дялове на всички мобилни браузъри. В изследването са включени и данни за FireFox, тъй като използва различна машина за оформление на HTML.

4. Методология

Съгласно препоръките на World Wide Web Consortium (W3C), съвременните мобилни браузъри трябва да поддържат интерфейси за следни-

те сензори: API за универсален сензор, акселерометър, сензор за геолокация, сензор за близост, жироскоп, сензор за ориентация и сензор за осветеност[5].

За провеждането на експеримента е създаден уебсайт, достъпен на адрес: https://www.learning-is.fun/mdsati (фиг. 1). На сайта е присвоено работно название: Mobile Device Sensors Access Testing via Internet (MDSATI). Преведено на български език: „*Тестване на достъпа до сензорите на мобилно устройство през интернет*“.



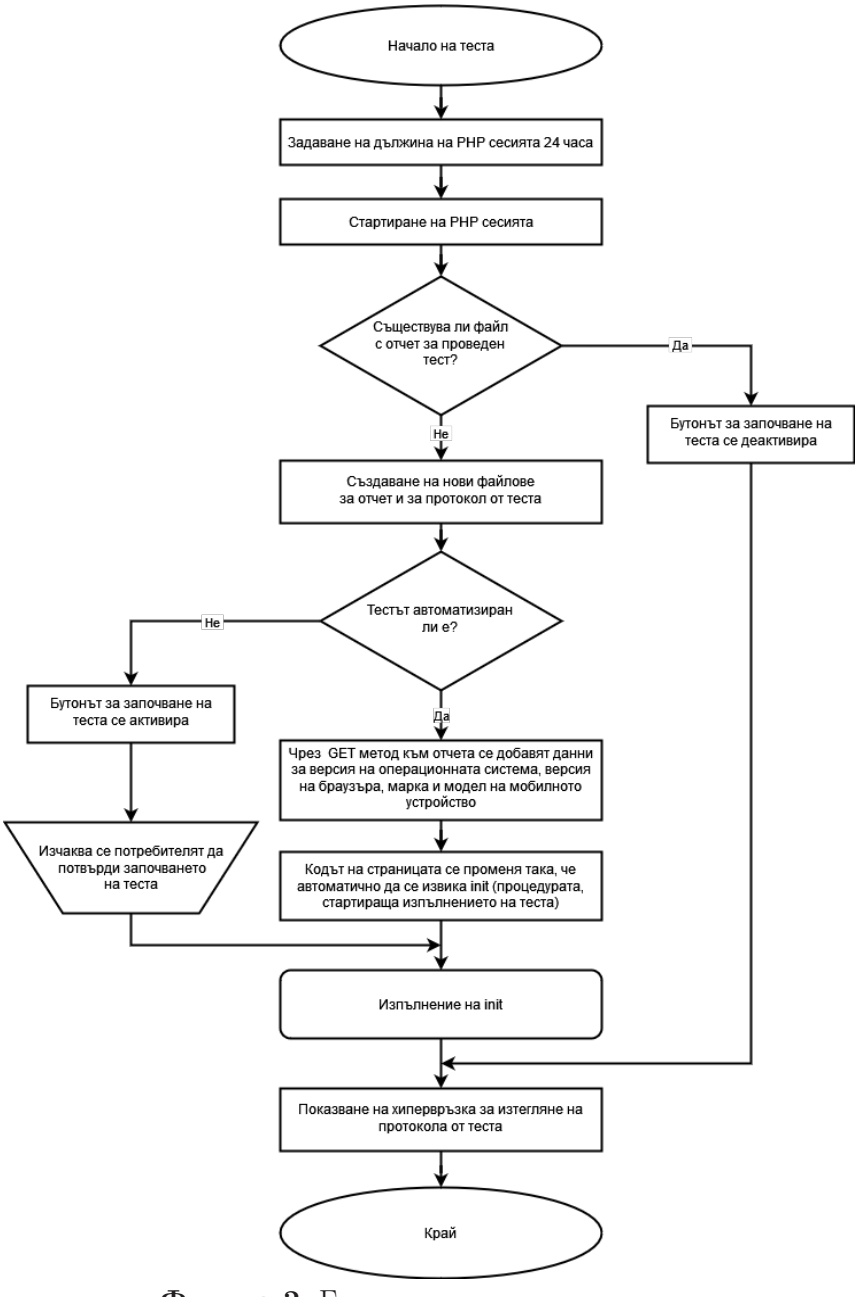
Фигура 1. Изглед от началната страница на уебсайта, от който се провежда експериментът

Началната страница на сайта (index.php) съдържа PHP скрипт, който предотвратява многократното изпълнение на един и същи тест. За целта се стартира PHP сесия с продължителност 24 часа и се проверява за наличието на файл с резултати. Ако съществува файл с резултати, останалата част от страницата се модифицира така, че да съдържа съобщение, че тестът вече е изпълнен, и активна хипервръзка към протокол с резултатите от теста. Бутонът за стартиране на теста е деактивиран.

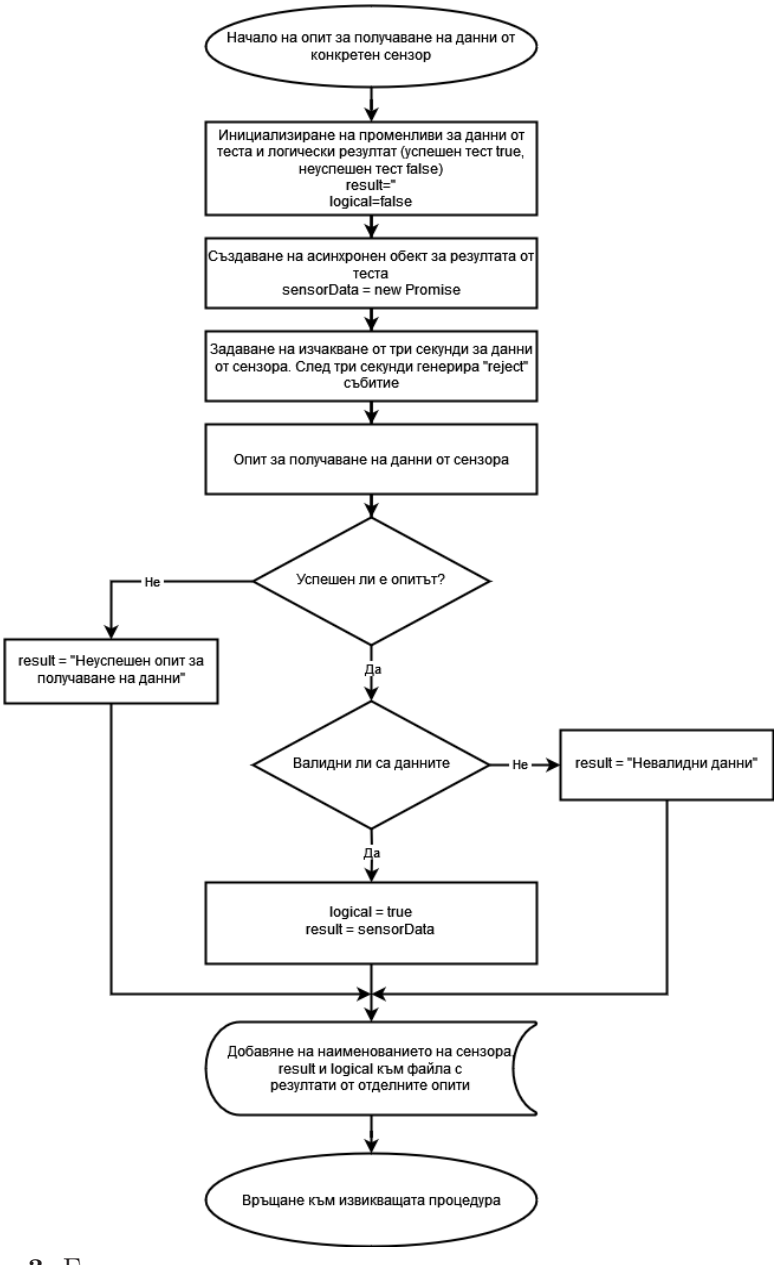
Ако не съществува файл с резултати, се извиква createReportFiles.php, PHP скрипт който създава файл за резултатите и файл за протокола от теста. Бутонът за изпълнение на теста е активен, а хипервръзката към протокола е скрита.

След това се прави опит за достъп до следните сензори:

- сензор за ориентация в пространството;
- сензор за ускорение;
- сензор за честота на завъртане;
- сензор за близост до обект;
- сензор за ниво на осветеност на околната среда;



Фигура 2. Блок-схема на експеримента



Фигура 3. Блок-схема на алгоритъм за опит за получаване на данни от конкретен сензор

• магнитен сензор (този сензор има две приложения – измерване на нивото на магнитното поле и компас);
• сензор за батерията;
• сензор за географско позициониране;
• камера;
• микрофон.

По време на теста се извеждат кратки информативни съобщения за изпълнение на опит за достъп до всеки от сензорите. В протокола се добавя подходящо съобщение, в зависимост от резултата от проверката за достъп до сензорите. Алгоритъмът за провеждане на експеримента е показан с блок-схемата на фиг. 2, а алгоритъмът на опит за получаване на данни от конкретен сензор е показан с блок-схемата на фиг. 3.

Възниква законният въпрос за поверителността на резултатите.

По време на теста (ако потребителят даде разрешение) се прави снимка с камерата на устройството и се записва звук. Така създадените файлове не се съхраняват на сървъра и са достъпни само докато страницата е отворена.

Имената на файловете за отделните протоколи се генерират автоматично. Следователно съществува възможност за опит за налучкване на името на файла с протокол, генериран за друг потребител. За да се избегне тази ситуация, са предприети следните мерки.

• За всеки потребител се създава PHP сесия. След това към името на файла, съдържащ отчета, се добавя уникалният идентификатор на PHP сесията. Така името на файла не може да бъде налучкано.

• Допълнително са зададени права за достъп до папката, в която се съхраняват протоколите, така че съдържанието ѝ да не може да бъде разгледано.

5. Резултати от проведените експерименти

По време на опитите се оказа, че при опит за задействане на вибрацията браузърът FireFox не дава съобщение за грешка. В същото време устройството не вибрира. Получаването на реална информация относно успешността на опита предполага анкетиране на потребителя. С цел да се облекчат потребителите при провеждане на експеримента, е взето решение данните от този сензор да не се подлагат на анализ.

Данните за достъп до геолокация, камера и микрофон не са анализирани, тъй като при опит за достъп до тези сензори потребителят винаги получава предупреждение (освен ако не е дал дългосрочно разрешение). Следователно може да се приеме, че въпросът относно сигурността на тези сензори е решен. Включването на тези сензори в експеримента цели да насочи вниманието на потребителите към това какви разрешения

дават, и да сравнят с това, че за другите данни не се дават разрешения.

Както се вижда от блок-схемата на алгоритъма за достъп до конкретен сензор (фиг. 3) – има логическа променлива, именувана *logical.* Възможните стойности на променливата са: състояние „Истина“ (отговаря на цифрата едно) или състояние „Лъжа“ (отговаря на цифрата нула). В началото на всеки от опитите за достъп до конкретен сензор променливата *logical* се инициализира със стойност „Лъжа“. Ако сензорът върне съобщение за грешка, опитът за достъп се счита за неуспешен и *logical* остава без промяна. Ако сензорът върне данни, опитът се счита за успешен и стойността на *logical* се променя на „Истина“. По време на експеримента е установено, че някои браузъри връщат стойност NULL като данни от сензор. В резултат на това променливата *logical* получава стойност „Истина“, без да са получени реални данни от сензора. Такъв резултат е фалшиво положителен (табл. 1).

Таблица 1. Възможни резултати при опит за получаване на достъп до сензор

	Отговор от сензор		
	Получено съобщение за грешка при опит за достъп до сензор	Получени реални данни от сензор	Получени NULL данни от сензор
Резултат	Отрицателен	Положителен	Фалшиво положителен

Това налага ръчна обработка на данните за получаване на реални резултати (табл. 2). В обобщените резултати фалшиво положителните тестове са отчетени като неуспешни опити.

Таблица 2. Примери за фалшиво положителни резултати

Прот.	OS	Browser	Device	Orientation	Acceleration
1.	Android 14	Chrome 117.0.59.38	Google Pixel 8	1	1
				alpha: undfined,. . .	accelerationX: 0,. . .
2.	Android 13	FireFox 119.1.1	Samsung Galaxy	1	1
				alpha: 13.504,. . .	accelerationX: null,. . .

Забележка: Таблицата е съставена по данни от описания експеримент.

И в двата случая очевидно това е въпрос на програмна реализация от страна на разработчиците на браузъра.

5.1. Проведени еккспериментии

Проведени бяха редица експерименти с реални мобилни устройства. **Изследвани бяха 32 мобилни телефона**, а именно: Samsung Galaxy S22, Google Pixel 5, Google Pixel 6, Google Pixel 7, Google Pixel 7-Real, Google Pixel 8, iPhone 11, iPhone 11 Pro Max, iPhone 12, iPhone 12 Pro Max, iPhone 13, iPhone 13 Pro, iPhone 13 Pro Max, iPhone 14 Pro Max, iPhone 14Pro, iPhone 15, iPhone 15 Pro Max, iPhone 7, iPhone 8, iPhone XS, Samsung Galaxy A32 5G, Samsung Galaxy S20, Samsung Galaxy S21, Samsung Galaxy S22, Samsung Galaxy S23, Samsung Galaxy Z Fold 5, Samsung Galaxy A11, Samsung Galaxy A52, Xiaomi Redmi Note 11, Xiaomi Redmi Note 12 4G, Xiaomi Redmi Note 12 Pro, Xiaomi Redmi Note 8.

Изследвани бяха и 13 базови версии на операционни системи за мобилни устройства: Android 10, Android 11, Android 12, Android 13, Android 14, Android 9, iOS 10, iOS 12, iOS 13, iOS 14, iOS 15, iOS 16, iOS 17, като данните от експериментите са достъпни на адрес:

https://github.com/StoyanMechev/mdsati_results

Обобщените данни от опитите за достъп до сензорите на мобилните устройства са представени в таблица 3.

Таблица 3. Обобщени данни по изследваните сензори

Брой опити/сензор	O	A	R	P	L	M	C	B
Успешни	3	27	26	0	0	0	24	22
Неуспешни	122	98	99	125	125	125	101	103
Общо	125	125	125	125	125	125	125	125
% успешни	2,40	21,60	20,80	0	0	0	19,20	17,60

Легенда: O – Orientation; A – Acceleration; R – RotationRate; P – Proximity; L – Light; M – Magnetometer; C – Compass; B – Battery.

5.2. Обобщени резултати по браузъри и операционни системи

По време на провеждане на експериментите не бяха налични браузъри Samsung Internet за iOS и Safari за Android. По тази причина липсват такива експериментални данни. Съкращенията за наименованията на сензорите са същите, както в таблица 3. В таблици 4 и 5 са представени обобщените данни за Chrome.

Става ясно, че браузърът Chrome почти винаги има достъп до сензорите за ускорение и честота на ротация, до компаса и до нивото на зареждане на батерията, когато работи под Android, и само веднъж има

достъп до сензорите за ориентация, ускорение и честота на ротация, когато работи под iOS. Може да се приеме, че Chrome е уязвим за атаки по странични канали, когато работи под Android.

Таблица 4. Браузър Chrome под Android

Брой опити/сензор	O	A	R	P	L	M	C	B
Успешни	0	18	17	0	0	0	17	18
Неуспешни	18	0	1	18	18	18	1	0
Общо	18	18	18	18	18	18	18	18
% успешни	0	100	94,44	0	0	0	94,44	100

Таблица 5. Браузър Chrome под iOS

Брой опити/сензор	O	A	R	P	L	M	C	B
Успешни	1	1	1	0	0	0	0	0
Неуспешни	20	20	20	21	21	21	21	21
Общо	21	21	21	21	21	21	21	21
% успешни	4,76	4,76	4,76	0	0	0	0	0

В таблици 6 и 7 са представени обобщените данни за FireFox.

Таблица 6. Браузър FireFox под Android

Брой опити/сензор	O	A	R	P	L	M	C	B
Успешни	1	0	0	0	0	0	0	0
Неуспешни	12	13	13	13	13	13	13	13
Общо	13	13	13	13	13	13	13	13
% успешни	7,69	0	0	0	0	0	0	0

Таблица 7. Браузър FireFox под iOS

Брой опити/сензор	O	A	R	P	L	M	C	B
Успешни	0	0	0	0	0	0	0	0
Неуспешни	41	41	41	41	41	41	41	41
Общо	41	41	41	41	41	41	41	41
% успешни	0	0	0	0	0	0	0	0

Браузърът FireFox е получил достъп до сензора за ориентация само в един от проведените експерименти. Следователно FireFox е много устойчив на атаки по странични канали.

В таблица 8 са представени експерименталните резултати за Samsung Internet. За този браузър са направени по-малко експерименти, тъй като той е разработен само за мобилните телефони Samsung.

Таблица 8. Браузър Samsung Internet под Android

Брой опити/сензор	O	A	R	P	L	M	C	B
Успешни	0	7	7	0	0	0	7	4
Неуспешни	7	0	0	7	7	7	0	3
Общо	7	7	7	7	7	7	7	7
% успешни	0	100	100	0	0	0	100	57,14

Резултатите показват, че този браузър е уязвим за атаки по странични канали за сензорите за ускорение, честота на въртене и за показанията на компаса във всички изследвани случаи.

В таблица 9 са представени обобщените резултати за Safari.

Таблица 9. Браузър Safari под iOS

Брой опити/сензор	O	A	R	P	L	M	C	B
Успешни	1	1	1	0	0	0	0	0
Неуспешни	24	24	24	25	25	25	25	25
Общо	25	25	25	25	25	25	25	25
% успешни	4	4	4	0	0	0	0	0

Резултатите показват, че този браузър е устойчив на атаки по странични канали. Само в един от случаите има достъп до сензорите за ориентация, ускорение и честота на въртене.

5.3. Анализ на получените резултати

• Показателно е, че всички опити за достъп до сензорите Proximity, Light и Magnetometer са неуспешни. Това свидетелства за стремежа на разработчиците на софтуер да повишат сигурността на браузърите. Например спирането на поддръжката на Ambient Light Sensor (в експеримента Light) предпазва от атаката, демонстрирана от Olejnik през 2017 г. (Olejnik 2017).

• Може да се направи изводът, че браузърите Chrome и Samsung Internet, работещи под Android, са податливи на атаки по странични канали, а FireFox и Safari са устойчиви.

• Въпреки че поддръжката на интерфейса за следене на нивото на батерията е прекратена още през 2017 г. (Olejnik et al. 2017), експерименталните резултати показват, че този интерфейс се поддържа от Chrome.

• От една страна, може да се създаде впечатлението, че софтуерът не работи правилно под iOS, тъй като почти всички опити за достъп до сензорите са неуспешни. От друга страна, при iPhone XS с iOS 12, Chrome и Safari имат достъп до три от сензорите. Едновременно с това, при същото устройство, но с iOS 13 и iOS 14 и двата браузъра нямат

достъп до сензорите (табл. 5 и табл. 9). Следователно iOS, като цяло, е по-малко податлива на атаки по странични канали чрез браузър, тъй като достъпът на браузърите до сензорите е ограничен.

6. Дискусия и заключение

Група от уязвимости, открити и публикувани през 2016 г., са подложени на експериментално обследване със специално разработен в хода на изследването софтуер.

Става ясно, че има сензори, за които потребителят е задължително известен, когато има опит за достъп до тях. Такива са камерата, микрофонът и геолокацията. С оглед на описаните уязвимости за останалите сензори, като сензорите за ускорение, ориентация, фоново осветление, компас (Diamantaris et al. 2020), е логично да се препоръча да се използва същият подход за известяване на потребителя.

Експериментът потвърди, че към края на проучването част от уязвимостите не са отстранени, въпреки тяхната важност за сигурността на ОСМУ.

Разработеният за експеримента софтуер е на работещо ниво и с перспективи за по-нататъшно усъвършенстване.

Софтуерът MDSATI може да послужи за повишаване на осъзнатостта на потребителите за рисковете за сигурността по отношение на разрешенията, които дават, и по отношение на сайтовете, които посещават.

Следва да се отчете, че при опита за достъп до сензора за ориентация има значително количество фалшиво положителни резултати. На даденото ниво на разработка на MDSATI не е предвидено автоматично отчитане на фалшиво положителните резултати при опит за достъп до сензор. Това налага допълнителна ръчна обработка на опитните данни за получаване на достоверни резултати. За в бъдеще следва да се подобри алгоритъмът по такъв начин, че този пропуск да бъде отстранен. Друга посока на разширение на функционалността на софтуера е да се добави към протокола по-подробна информация относно това кой сензор на каква атака може да бъде подложен.

БЕЛЕЖКИ

1. Атака по страничен канал, на английски: side-channel attack – атака, при която някои от сензорите се използват по алтернативен начин, за да се извлече чувствителна информация.
2. Statista. Number of smartphone users 2014-2029. https://www.statista.com/forecasts/1143723/smartphone-users-in-the-world. Viewed 03.10.2024.

3. Statcounter. Mobile Operating System Market Share Worldwide. https://gs.statcounter.com/os-market-share/mobile/worldwide. Viewed 03.10.2024.
4. Statcounter. Mobile Browser Market Share Worldwide. https://gs.statcounter.com/browser-market-share/mobile/worldwide/ 2023. Viewed 03.10.2024.
5. W3C, 2023. W3C standards and drafts. https://www.w3.org/TR/. Viewed 03.10.2024.

REFERENCES

DIAMANTARIS, M., MARCANTONI, F., SIOANNIDIS, S., POLAKIS, J., 2020. The Seven Deadly Sins of the HTML5 WebAPI. *ACM Transactions on Privacy and Security*, vol. 23, no. 4, pp. 1 – 31. https://doi.org/10.1145/3403947.

JENNEX, M.E., 2015. Literature reviews and the review process: An editor-in-chiefs perspective. *Communications of the Association for Information Systems*, vol. 36, pp. 139 – 146. https://doi.org/10.17705/1CAIS.03608.

MEHRNEZHAD, M., TOREINI, E., SHAHANDASHTI, S.F., HAO, F., 2016. TouchSignatures: Identification of user touch actions and PINs based on mobile sensor data via JavaScript. *Journal of Information Security and Applications*, vol. 26, pp. 23 – 38. https://doi.org/10.1016/J.JISA.2015.11.007.

MEHRNEZHAD, M., TOREINI, E., 2019. What is this sensor and does this app need access to it? *Informatics*, vol. 6, no. 1. ISSN 22279709. Available from: https://doi.org/10.3390/INFORMATICS6010007.

MOREAU, S., 2021. The evolution of iOS. *Computerworld*. https://www.computerworld.com/article/2975868/the-evolution-of-ios.html

OLEJNIK, L., 2017. *Stealing sensitive browser data with the W3C Ambient Light Sensor API*. https://blog.lukaszolejnik.com/stealing-sensitive-browser-data-with-the-w3c-ambient-light-sensor-api/.

OLEJNIK, L., ENGLEHARDT, S., NARAYANAN, A., 2017. Battery Status Not Included: Assessing Privacy in Web Standards. In: *CEUR Workshop Proceedings*, vol. 1873, pp. 17 – 24. https://senglehardt.com/papers/iwpe17_battery_status_case_study.pdf

RAPHAEL, JR, 2023. Android versions: A living history from 1.0 to 14. *Computerworld*. https://www.computerworld.com/article/3235946/android-versions-a-living-history-from-1-0-to-today.html.

AN APPROACH FOR DETECTING SECURITY VULNERABILITIES IN WEB BROWSERS FOR MOBILE OPERATING SYSTEMS

Abstract. This paper demonstrates that it is possible to access a mobile phone's sensors through a web browser without the user's knowledge. Thus mobile phones are vulnerable to side-channel attacks. A software tool was developed to check access to a smartphone's sensors and generate a report with results and recommendations.

After reviewing scientific publications on the topic, a technical experiment was conducted to prove that access to various sensors of the examined device can be accessed without the user being aware of it. The experimental results show that Chrome and Samsung Internet browsers, running on Android, are vulnerable to side-channel attacks. The research methods used in this paper were comparative analysis, synthesis and technical experiment.

Keywords: Android; iOS; security; vulnerability; browser; web browser; side-channel attacks

✉ **Dr. Stoyan Mechev, Assist. Prof.**
ORCID iD: 0000-0002-0809-8538
WoS Researcher ID: AAB-4270-2021
Nikola Vaptsarov Naval Academy
73, Vasil Drumev St.
9002 Varna, Bulgaria
E-mail: st.mechev@naval-acad.bg