Security Analysis of Mobile Web Browser Hardware Accessibility: Study with Ambient Light Sensors

Sanghak Lee[1,2], Sangwoo Ji[1], and Jong Kim[1](✉)

[1] Department of Computer Science and Engineering, Pohang University of Science and Technology (POSTECH), Pohang, South Korea
{uzbu89,sangwooji,jkim}@postech.ac.kr

[2] Defense Industry Technology Center (DITC), Pohang, South Korea

Abstract. Mobile web browsers are evolved to support the functionalities presented by HTML5. With the hardware accessibility of HTML5, it is now possible to access sensor hardware of a mobile device through a web page regardless of the need for a mobile application. In this paper, we analyze the security impact of accessing sensor hardware of a mobile device from mobile web page. First, we present the test results of hardware accessibility from mobile web browsers. Second, to raise awareness of the seriousness of hardware accessibility, we introduce a new POC attack LightTracker which infers the victim's location using light sensor. We also show the effectiveness of the attack in real world.

1 Introduction

HTML5 is a fifth hypertext markup language standard which have introduced many new features to keep pace with advances in computer performance and increased access to websites. HTML5 has aimed to reduce the necessity for plugins (e.g., Flash, Silverlight). Browsers that support HTML5 can play multimedia such as video and audio, utilize offline storage of web resource, and graphic work using canvas API without the assistance of external programs.

However, as the functionalities of HTML5 have increased, there have been a number of attack paths exploiting vulnerabilities of the functionalities. Numerous privacy attack methods have been reported using HTML5 features, for example browser fingerprinting through canvas API [10], and side-channel attack through Application Cache [24].

One new feature of HTML5 is hardware accessibility of web browsers. This feature has big security implication for mobile devices which have various types of sensor. On Android system, in order to access the hardware of a mobile device, it needs a capability called *permission*. Most of popular browsers require over eight *permissions* related to hardware access. In this environment, if the hardware access permission is given to a web browser and it does not have a fine access control mechanism for hardware, an attacker can create a malicious web server

B. B. Kang and J. Jang (Eds.): WISA 2018, LNCS 11402, pp. 3–15, 2019.
https://doi.org/10.1007/978-3-030-17982-3_1

and control the hardware of the connected user device without any restriction and get the hardware data.

In this paper, we investigate the security impact of hardware accessibility through mobile web browsers. First, to identify reality of hardware access problem and aware their risk, we investigate the security and privacy aspects and the current status of hardware access in various environments. We show accessible hardware components from various mobile web browsers and their access test results. Through this investigation, we show that most of popular browsers have no access restriction to motion sensor and ambient light sensor. Second, to show the seriousness of this hardware accessibility, we show the POC attack LightTracker using ambient light sensor. Most attack methods through mobile device hardware are concentrated only to using motion sensor. The risk of leaking light data is underestimated despite of the possibility of leaking privacy. To show the hardware access problem, we introduce LightTracker which infers the victim's location through ambient light sensor data. We show the evaluation results of LightTrack's effectiveness in real environments.

This paper is organized as follows. In Sect. 2, we show the hardware accessibility status in mobile environments. Section 3 introduces the POC attack LightTracker using web browser in mobile device. Section 4 shows the evaluation results of LightTracker. Finally, in Sect. 5, we conclude the work and discuss the future works.

2 Hardware Accessibility in Mobile Environments

In this section, we investigate the security & privacy aspects of hardware access and the current state of hardware access in various environments, i.e., combinations of mobile phone and web browsers.

2.1 Security and Privacy Consideration of Hardware Components

Web applications tend to have more accessibility to hardware of mobile devices. According to World Wide Web Consortium (W3C), mobile web applications can access the hardware data such as battery status [25], GPS [26], vibration [27], ambient light sensor [28], multimedia (camera, microphone) [29], and motion sensor [30].

W3C specification provides security & privacy consideration on hardware access within mobile browsers. Including these consideration, we investigate all concerns for each hardware component.

Battery Status. Battery status API shows current remaining power, charging status, and (dis)charging time of the hosting device. With tracking the change of the remaining power, it could leak the user's privacy [20]. To prevent such kind of threats, W3C suggests that web browsers should (1) not expose high precision data of battery, (2) enforce the user permission requirements, (3) inform the user of the API use, and (4) obfuscate the exposed value of the battery.

GPS. Web applications can access GPS data through Geolocation API. This API is used to retrieve the geographic location of a hosting device. This API directly exposes the user's location, so location data should be controlled with the user's perception, i.e., the browsers should acquire permission through a user interface, and this permission should be a revocable one.

Vibration. Vibrator is an actuator. Vibration functionality does not generate data to be used to leak privacy data. However, vibration stimuli can be used as an agitation source for privacy attack. For example, accelerators or gyroscopes may be subject to tiny imperfections during their manufacturing process [5]. Vibration API can help discovering those imperfections by introducing vibration patterns. Also, any unique imperfection can be used to fingerprint mobile devices. Moreover, continuous method calls for vibration could exhaust the battery power of the device which harms the availability or leads to privacy leaks [13].

Generic Sensor. Generic sensors consist of accelerometer, gyroscope, magnetometer, and ambient light sensor. When these sensors are in combination with other functionality, or used over time, user's privacy can be in risk by a simple attack such as user identification with fingerprinting. Numerous researchers propose various privacy attack methods using motion sensor data [4,32].

Multimedia. Allowing the access of multimedia data in mobile devices is directly related to the user's privacy. Leaking multimedia data itself can harm the confidentiality of the device [6], moreover other non-multimedia data can also be used to leak the user's private data, for example, embedding the user's location in the metadata of EXIF file.

2.2 Current Status of Hardware Access

Test Browser Selection. We test the hardware accessibility of mobile web browsers to identify how the real browsers handle security consideration described in 2.1. From the market share report of NETMARKETSHARE [19], we select representative mobile browsers such as Chrome, UC Browsers, Samsung Internet, and Firefox for hardware accessibility tests (Table 1). We exclude Safari browser because it is only for iOS, so we cannot compare this browser with

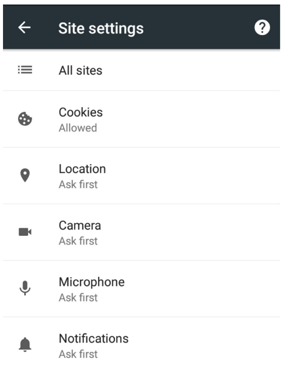
Table 1. Mobile browsers: numbers of downloads and tested version

Browser	Tested version	# of downloads
Chrome	58.0.3029.83	1,000,000,000+
UC Browser	11.3.2.960	100,000,000+
Samsung Internet	5.4.00-75	100,000,000+
Firefox	53.0.2	100,000,000+

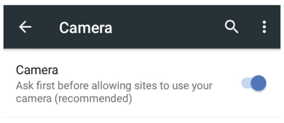
other browsers in Android. Also, we select Firefox because of its high HTML5 acceptance score [8]. Note that all tested browsers require most of hardware access permissions such as *camera, microphone, GPS, network connection state,* and *vibration*. Therefore if browsers have no self-control mechanism while accessing the hardware, then web applications are able to access the hardware of the mobile device without any restrictions.

Hardware Access Test. We test the data accessibility of eight hardware components (i.e., camera, microphone, GPS, network connection state, vibration, motion, light, and battery status) of three different mobile devices, Galaxy Note3, Galaxy S5, and Nexus 5. For privacy-sensitive hardware components such as camera, microphone, and GPS, all tested browsers require user's permission to access the hardware component's data. However, other hardware components' data such as network connection state, vibration, motion sensor, ambient light sensor, and battery status could be accessed without permission, except the vibration by the Firefox browser. Also, we observe that each browser shows different access control policy to hardware. As we mentioned in Sect. 2.1, seemingly privacy-insensitive hardware in mobile devices can be used to leak the user's privacy.

Also, we survey whether each tested browser has setting options for the hardware access. The result shows that only the Chrome browser has the setting options for camera, microphone, and GPS (Fig. 1(a)). This option only covers the default access policy, whether asking first before allowing sites to use hardware component or completely blocking the access to the hardware component (Fig. 1(b)) (Table 2).


Site settings
All sites
Cookies
Allowed
Location
Ask first
Camera
Ask first
Microphone
Ask first
Notifications
Ask first

(a) Access Setting


Camera
Camera
Ask first before allowing sites to use your camera (recommended)

(b) Access Blocked

Fig. 1. Setting options for the hardware access in Chrome

Table 2. Hardware access tests in mobile browsers (O: Required permission, X: Zero-permission, -: Not supported)

Device	Browser	Hardware							
		Camera	Mic	GPS	Network	Vibration	Motion	Light	Battery
Galaxy Note3	Chrome	-	-	-	X	X	X	-	-
	UC	O	O	O	-	X	X	-	-
	SI	O	O	O	X	X	X	-	-
	Firefox	O	O	O	X	O	X	X	-
Galaxy S5	Chrome	-	-	-	X	X	X	-	-
	UC	-	O	O	-	X	X	-	X
	SI	O	O	O	X	X	X	-	-
	Firefox	O	O	O	X	O	X	X	-
Nexus 5	Chrome	O	O	O	X	X	X	-	-
	UC	O	O	O	-	X	X	-	X
	SI	-	-	-	X	X	X	-	-
	Firefox	O	O	O	X	O	X	X	-

Table 3. Reported attack using hardware or its data in mobile devices (Accel: Accelerometer, Gyro: Gyroscope, Orient: Orientation, Magnet: Magnet)

Attack	Work	Access	Hardware
Inferring input	PIN Skimmer [22]	in-app	Camera, Microphone
	PIN Skimming [23]	in-app	Light sensor
	Keylogging by Mic [16]	in-app	Microphone
	ACCessory [21]	in-app	Accel
	Tapprints [14]	in-app	Accel, Gyro
	Accel side channel [1]	in-app	Accel
	Motion side channel [3]	in-app	Accel, Gyro
	TapLogger [31]	in-app	Accel, Orient
	TouchLogger [2]	in-app	Orient
	TouchSignatures [11]	in-browser	Accel, Gyro, Orient
Tracking	Route identification [17]	in-app	Accel, Gyro, Magnet
	ACComplice [7]	in-app	Accel
	Activity recognition [9]	in-app	Accel
	Peril tracking [18]	in-app	Accel, Gyro, Magnet
	Sensor fingerprinting [4]	in-browser	Accel, Gyro

2.3 Threat Models in Hardware Access

To better understand the security and privacy issues on hardware access in mobile devices, we performed a survey of attacks using sensor data through both browsers and applications. We classified the attacks into two classes, *Inferring input*, and *Tracking*, then we compare the access route and used hardware components of each attack (Table 3).

Inferring Input. Inferring input attacks consist of identifying the user's touch input to infer password/PINs using sensors such as camera, microphone, light, accelerometer, gyroscope, and magnetometer. Except PIN Skimmer [22], PIN Skimming [23], and Keylogging [16], most of the reported inferring input attacks exploit motion sensor data from accelerometer especially. The attacks exploit the characteristic; when the user types or touches a screen on his/her device, these actions induce the device orientation or motion traces which might be distinguishable from those of other actions.

Tracking. Tracking attacks include inferring the victim's route (or activity) and fingerprinting victim's device. Most of reported tracking attacks exploit motion sensor data as same as inferring input attacks. These kinds of attacks use the data whose variation is usually bigger than those of inferring input attacks. This feature is well fit for attacks through browsers; because browsers have a reduced sampling rate of sensors which is about 3–5 times slower than the original sampling rate [11,12], it is relatively hard to distinguish the small variation of data at browsers.

Through the investigation, we have observed that most of the reported attacks consider only motion sensors among zero-permission hardware components. Only PIN Skimming [23] uses the light data from ambient light sensors in mobile devices. The risk of leaking light data is underestimated despite of the possibility of side channel attack exploiting light data. Also, most attacks are conducted through applications, not browsers. However, attacks through browsers can expose a victim easier than attacks through applications, because browser-based attacks need no installation, but just need to allure the victim to the attacker's web pages.

3 POC: LightTracker

In this section, to raise the awareness of the risk of the ambient light sensor data, we introduce LightTracker, which infers the victim's location through ambient light sensor data. LightTracker is deployed on the attacker's server. When a victim visits the attacker's web pages, LightTracker collects the light data of the victim's mobile device. Next, LightTracker compares the collected data and predefined light map data to infer the victim's position.

3.1 Attack Procedure

We explain overall attack procedures of LightTracker (Fig. 2). It assumes that a victim visits the attacker's web pages. The attacker's web server includes JavaScript which makes the client's device leak the sensor data.

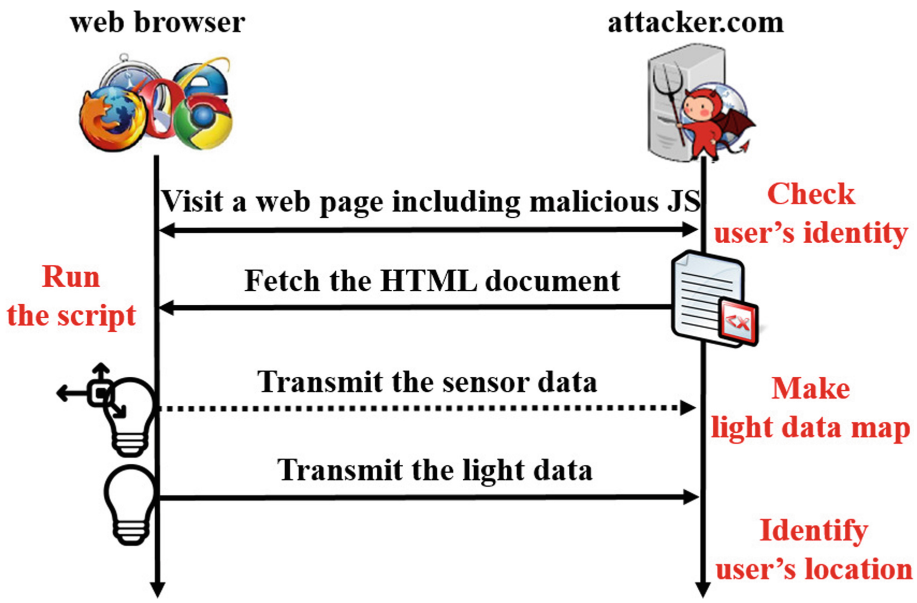


Fig. 2. Overall attack procedure of LightTracker

❶ A victim visits the web pages (attacker.com) which include a malicious JavaScript. Then the attacker checks the victim's identity, and examines whether this access is the victim's first visit or not. If this is the first visit, the attacker fingerprints the client browser. Note that we assume the user's IP address (subnet) represents the user's location.

❷ The client web browser starts to download the HTML document and runs the script.

❸ By running the script, the client browser transmits the sensor data to the attacker's server. To create light map, LightTracker manages the motion sensor data with light data. Note that this process is not mandatory if the server already has a light map for the victim; in this case, LightTracker requires no privilege to access motion sensor data.

❹ The client browser transmits the light data to the server. Then LightTracker checks the data with the map to identify the victim's location.

3.2 Attack Details

In this subsection, we explain details of the attack; checking user's identity, making light data map, and identifying user's location.

Checking User's Identity. To distinguish a visited victim, LightTracker needs to check the victim's identity. Various fingerprinting work [4,15] can be used. As HTML5 has numerous new features such as canvas API and hardware access API, attack surfaces for fingerprinting are extended.

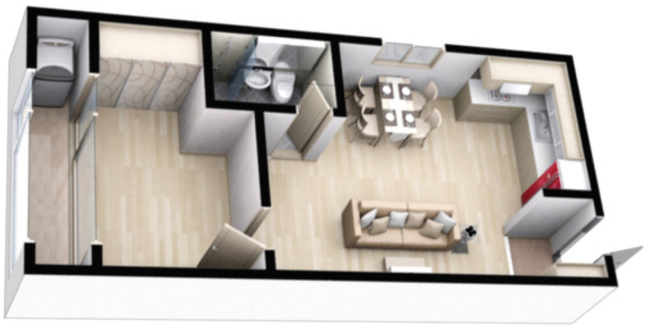
LightTraker collects user's IP address to match user's location with the location's light data map. The user may visit the attacker's site with different IP addresses as he moves around to different places. For each IP address, we may build a light data map for a user's mobile device. Therefore user's identity with an IP address is used to find a matching light data map. We assume that each device needs a different light data map since light sensors are not equal. However, if we use a normalization method for sensor data, it may be possible to use the same light data map for different mobile devices.

Making the Light Data Map. A Light data map represents the value of the light intensity for each area unit. After identifying the victim's identity and approximate location, LightTracker checks whether the light map for the victim's location exists or not. If LightTracker has already built the corresponding light map of a mobile device for the location, then LightTracker skips making light map phase. On the other hands, if LightTracker does not have enough data for a light map for the corresponding location, it collects data from motion sensors (accelerometer, gyroscope, and magnetometer) to infer the victim's movements. By inferring this movement, LightTracker is able to make the naive geographic map [17]. At the same time, to make a light data map, LightTracker also gathers the ambient light sensor data and combines the data with each area unit of the geographic map. Note that collecting motion data is not mandatory. A light map data for a specific place can be generated if that place are physically accessible to the attacker. He or she can create more accurate light map and matches the map with the IP address of that place. In this way, LightTracker obtains more accurate light map than the map derived from the motion sensor data.

Identifying the User's Location. With the light data map, LightTracker can infer the current victim's location with transmitted light data from the victim's device. LightTracker compares the transmitted data with the value of unit areas in the light data map, and finds the exact location which has the most similar light value.

4 LightTracker Evaluation

We implemented a prototype of LightTracker and tested its effectiveness.


(a) The apartment modeling

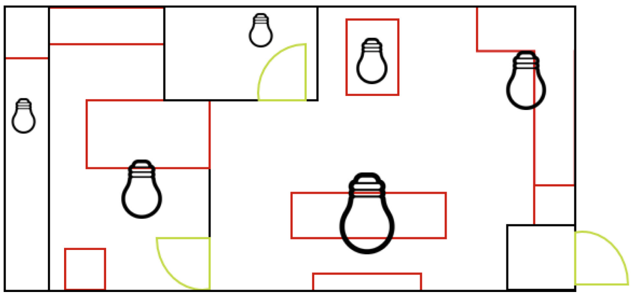

(b) Floorplan of the apartment

Fig. 3. Modeling and floorplan of test environments used for testing LightTracker. Each bulb icon denotes a light source, and its size represents the light power of the source.

4.1 Test Environments

We tested the effectiveness of LightTracker in a typical apartment covering a $10 \times 6\,\mathrm{m}^2$ area and consisting of a living room connected to the bedroom and the toilet (Fig. 3(a)). We made a light map manually for an accurate location inferring. We used Samsung Galaxy Note3 as a test device and a Firefox browser to acquire ambient light data. We divided the apartment into $1 \times 1\,\mathrm{m}^2$ area units, and measured the ambient light data for each area.

4.2 Making Light Data Map

We measured the ambient light sensor data for each unit area 10 times, and recorded the mean value. With this data, we made a light map to test the effectiveness of LightTracker. The light intensity value is directly related to the distance from light sources. In Fig. 4, we visualized the light map in log scale, and it shows that we can infer the victim's location with the light intensity value.

Table 4. Light intensity data for each area

Light (lux)	8–16	16–32	32–64	64–128	128–256	256–512	512–1024	Total
Area (m^2)	8.915	11.33	9.545	10.68	6.39	2.53	0.61	50
Percent (%)	17.83	22.66	19.09	21.36	12.78	5.06	1.22	100

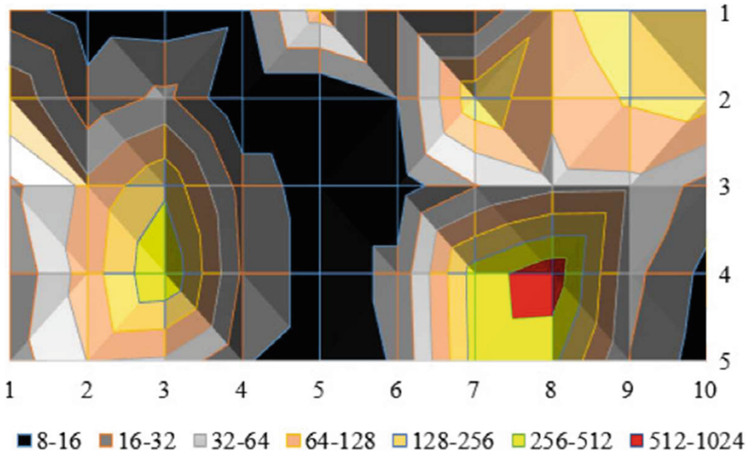


Fig. 4. The light map visualization in log scale

4.3 Effectiveness Evaluation

With the light map, LightTracker significantly reduces candidates of victim's location (Table 4). In most case, the candidates are decreased to roughly 20% of total areas, through the light intensity value of that area. Especially, as the light intensity value is bigger, LightTracker infers a more accurate position. For example, among 50 m^2 area, there exists only 1.22 m^2 area whose light intensity value is over 512 lux. Also, as the number of divided areas in location data map increases, LightTracker can specify the victim's location more accurately. Moreover, as LightTracker can be combined with motion-based user tracking system orthogonally, it is possible to highly enhance the accuracy of user tracking.

5 Conclusion

In this paper, we investigated the security and privacy consideration of hardware access through mobile web browsers. Despite of the W3C specification and numerous in-app attacks, most popular browsers (e.g., Google Chrome, UC Browser, Samsung Internet, and Mozilla Firefox) do not have access control mechanism for motion sensor, vibration, battery status, and ambient light sensor. Although many researchers have shown attacks with motion sensor, ambient light sensor is underestimated despite of the possibility that it can be a medium of side-channel attacks. To raise the awareness of the risk of ambient light sensor, we have introduced a POC attack, LightTracker, which infers the victim's location. With the light data map, LightTracker can specify the victim's position accurately when the light intensity value is bigger.

Acknowledgement. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2017R1A2B4010914).

References

1. Aviv, A.J., Sapp, B., Blaze, M., Smith, J.M.: Practicality of accelerometer side channels on smartphones. In: Proceedings of the 28th Annual Computer Security Applications Conference, pp. 41–50. ACM (2012)
2. Cai, L., Chen, H.: TouchLogger: inferring keystrokes on touch screen from smartphone motion. HotSec **11**, 9 (2011)
3. Cai, L., Chen, H.: On the practicality of motion based keystroke inference attack. In: Katzenbeisser, S., Weippl, E., Camp, L.J., Volkamer, M., Reiter, M., Zhang, X. (eds.) Trust 2012. LNCS, vol. 7344, pp. 273–290. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30921-2_16
4. Das, A., Borisov, N., Caesar, M.: Tracking mobile web users through motion sensors: attacks and defenses. In: Proceedings of the 23rd Annual Network and Distributed System Security Symposium (NDSS) (2016)
5. Dey, S., Roy, N., Xu, W., Choudhury, R.R., Nelakuditi, S.: AccelPrint: imperfections of accelerometers make smartphones trackable. In: NDSS (2014)
6. Diao, W., Liu, X., Zhou, Z., Zhang, K.: Your voice assistant is mine: how to abuse speakers to steal information and control your phone. In: Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices, pp. 63–74. ACM (2014)
7. Han, J., Owusu, E., Nguyen, L.T., Perrig, A., Zhang, J.: ACComplice: location inference using accelerometers on smartphones. In: 2012 Fourth International Conference on Communication Systems and Networks (COMSNETS), pp. 1–9. IEEE (2012)
8. HTML5TEST: how well does your browser support HTML5? (2017). https://html5test.com/results/mobile.html
9. Kwapisz, J.R., Weiss, G.M., Moore, S.A.: Activity recognition using cell phone accelerometers. ACM SigKDD Explor. Newsl. **12**(2), 74–82 (2011)
10. Laperdrix, P., Rudametkin, W., Baudry, B.: Beauty and the beast: diverting modern web browsers to build unique browser fingerprints. In: 2016 IEEE Symposium on Security and Privacy (SP), pp. 878–894. IEEE (2016)

11. Mehrnezhad, M., Toreini, E., Shahandashti, S.F., Hao, F.: TouchSignatures: identification of user touch actions and PINs based on mobile sensor data via JavaScript. J. Inf. Secur. Appl. **26**, 23–38 (2016)
12. Michalevsky, Y., Boneh, D., Nakibly, G.: Gyrophone: recognizing speech from gyroscope signals. In: USENIX Security, pp. 1053–1067 (2014)
13. Michalevsky, Y., Schulman, A., Veerapandian, G.A., Boneh, D., Nakibly, G.: PowerSpy: location tracking using mobile device power analysis. In: USENIX Security, pp. 785–800 (2015)
14. Miluzzo, E., Varshavsky, A., Balakrishnan, S., Choudhury, R.R.: TapPrints: your finger taps have fingerprints. In: Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services, pp. 323–336. ACM (2012)
15. Mowery, K., Shacham, H.: Pixel perfect: fingerprinting canvas in HTML5. In: Proceedings of W2SP, pp. 1–12 (2012)
16. Narain, S., Sanatinia, A., Noubir, G.: Single-stroke language-agnostic keylogging using stereo-microphones and domain specific machine learning. In: Proceedings of the 2014 ACM Conference on Security and Privacy in Wireless & Mobile Networks, pp. 201–212. ACM (2014)
17. Narain, S., Vo-Huu, T.D., Block, K., Noubir, G.: Inferring user routes and locations using zero-permission mobile sensors. In: 2016 IEEE Symposium on Security and Privacy (SP), pp. 397–413. IEEE (2016)
18. Narain, S., Vo-Huu, T.D., Block, K., Noubir, G.: The perils of user tracking using zero-permission mobile apps. IEEE Secur. Priv. **15**(2), 32–41 (2017)
19. NETMARKETSHARE: Mobile/tablet top browser share trend (2017). https://www.netmarketshare.com/browser-market-share.aspx?qprid=1&qpcustomb=1
20. Olejnik, Ł., Acar, G., Castelluccia, C., Diaz, C.: The leaking battery. In: Garcia-Alfaro, J., Navarro-Arribas, G., Aldini, A., Martinelli, F., Suri, N. (eds.) DPM/QASA -2015. LNCS, vol. 9481, pp. 254–263. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-29883-2_18
21. Owusu, E., Han, J., Das, S., Perrig, A., Zhang, J.: Accessory: password inference using accelerometers on smartphones. In: Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications, p. 9. ACM (2012)
22. Simon, L., Anderson, R.: PIN skimmer: inferring PINs through the camera and microphone. In: Proceedings of the Third ACM Workshop on Security and Privacy in Smartphones & Mobile Devices, pp. 67–78. ACM (2013)
23. Spreitzer, R.: PIN skimming: exploiting the ambient-light sensor in mobile devices. In: Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices, pp. 51–62. ACM (2014)
24. Van Goethem, T., Joosen, W., Nikiforakis, N.: The clock is still ticking: timing attacks in the modern web. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp. 1382–1393. ACM (2015)
25. World Wide Web Consortium: Battery status API (2016). https://www.w3.org/TR/battery-status/
26. World Wide Web Consortium: Geolocation API specification, 2nd (edn.) (2016). https://www.w3.org/TR/geolocation-API/
27. World Wide Web Consortium: Vibration API, 2nd (edn.) (2016). https://www.w3.org/TR/vibration/
28. World Wide Web Consortium: Ambient light sensor (2017). https://www.w3.org/TR/ambient-light/
29. World Wide Web Consortium: HTML media capture (2017). https://www.w3.org/TR/html-media-capture/

30. World Wide Web Consortium: Motion sensors explainer (2017). https://www.w3.org/TR/motion-sensors/
31. Xu, Z., Bai, K., Zhu, S.: TapLogger: inferring user inputs on smartphone touchscreens using on-board motion sensors. In: Proceedings of the Fifth ACM conference on Security and Privacy in Wireless and Mobile Networks, pp. 113–124. ACM (2012)
32. Yue, C.: Sensor-based mobile web fingerprinting and cross-site input inference attacks. In: 2016 IEEE Security and Privacy Workshops (SPW), pp. 241–244. IEEE (2016)