The Rings of Tracking: Evaluating Security and Privacy in the Smart Ring Ecosystem

Johannes Ludwig
Ruhr University Bochum
AWARE7 GmbH
johannes@aware7.de

Veelasha Moonsamy
Ruhr University Bochum
email@veelasha.org

Matteo Große-Kampmann
Rhine-Waal University of Applied Sciences
AWARE7 GmbH
matteo.grosse-kampmann@hochschule-rhein-waal.de

***Abstract*—Smart rings represent the next innovation in health-tracking devices, capable of monitoring fitness, health, and sleep data. However, these devices also introduce user risks due to the sensitive data they handle. Despite their growing popularity, limited scientific research exists on their security and privacy implications. This paper systematically analyzed five smart rings, covering a range of prices and release dates. A test suite of 29 cases was applied to assess the security of the smart ring, companion app, and cloud backend. Critical vulnerabilities were found, including cleartext communication of sensitive data, unauthenticated firmware updates, and privacy violations like the lack of user consent. The findings highlight significant security and privacy risks in current devices, underscoring the need for CTI teams to integrate wearables and wearable devices into their threat models. Furthermore, we disclosed all identified vulnerabilities responsibly to the vendors.**

***Index Terms*—IoT, Tracking, Security**

1. Introduction

The rapid growth of the personal health tracker market has led to the emergence of smart rings, a new category of wearable devices that have gained popularity due to their small size and versatile health-tracking capabilities [1]. Despite their small form factor, these rings can perform various measurements comparable to traditional fitness trackers and smartwatches. The capabilities range from monitoring health metrics such as blood pressure, blood oxygen saturation, body temperature, and stress levels to measuring the quality of sleep [2]. However, with this technological advancement comes significant data protection, ethical, and security concerns [3]. Smart rings lack built-in user interfaces, leaving users with limited control over their functionality and increasing the risk of undiscovered security vulnerabilities, analogous to IoT devices [4]. The constrained hardware of these devices may compromise cryptographic algorithms and network protocols, raising concerns about the adequacy of security by design in developing smart rings.

This paper addresses the concerns above by conducting a systematic security and privacy analysis of the smart ring ecosystem. Specifically, it seeks to uncover vulnerabilities and assess the potential for attacks to mitigate risks and protect users' sensitive health data. Our work makes the following contributions to the field:

- A comprehensive collection of test cases and attack simulations targeting various components of the smart ring ecosystem, applied to five different smart rings.
- An in-depth risk and criticality assessment of the vulnerabilities uncovered, explicitly focusing on the impact on users, manufacturers, and third parties.
- Recommendations for mitigation strategies to address security issues and improve the overall robustness of smart ring devices.
- Discussion on the resulting implications for the Counter Threat Intelligence Community, as smart rings are currently not monitored by threat intelligence feeds and services.

2. Background & Related Work

This section provides the necessary technical background and an overview of existing research on the security and privacy of mobile applications, cloud servers, and wearable devices, particularly fitness trackers.

2.1. Background

Similar to other fitness trackers, smart rings consist of a wearable device (the ring), a companion app, and a cloud server infrastructure (backend). One unique factor of smart rings is that users cannot interact with them directly, as they lack buttons or user interfaces. The user wears the ring and starts to measure different biometric signals. This becomes important as the form factor is essential to how users perceive and communicate about privacy concerns [5]. For instance, if a smartwatch were to respond significantly slower to user input, record abnormal health metrics, or become paired with another Bluetooth device, a user would detect unusual behavior much more quickly than a smart

The authors acknowledge funding from the German Federal Office for Information Security (FKZ: Pentest-5GSec - 01MO23025A, 01MO23025B), the Federal Ministry of Education and Research (FKZ: HealthNet - 16KISR001K) and the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2092 CASA - 390781972.

ring or another device without a user interface [6]. With the possibility of limited or no control, a user does not have the opportunity to intervene promptly in case of unusual behavior or notifications when the companion application is not available. The diminutive size also imposes significant constraints on internal hardware [7]. This constraint raises concerns about adherence to best practices, particularly regarding cryptographic algorithm and network protocol implementations, which may not be consistently applied. Many algorithms could have been changed or redeveloped to run on the restricted hardware [8], [9]. Given the numerous constraints vendors face with smart rings, including design and development costs for the small form factor, battery life, features, and the pressure for a rapid market release, neglecting security across the entire smart ring ecosystem could become a potential scenario. Communication between the ring and app is typically carried out via Bluetooth Low Energy (BLE), while app-to-cloud communication uses HTTPS. Two typical communication schemes exist: one uses end-to-end encryption, ensuring that data is encrypted within the ring and inaccessible to the app, while the other treats the app as a central node, with data accessible to the app but relying on encrypted BLE and HTTP communication [10].

BLE is the primary communication protocol used by smart rings and is an energy-efficient protocol designed for low-power devices like wearables. It features two primary roles: the *Peripheral* (e.g., the smart ring) and the *Central* (e.g., the mobile phone), which initiate and manage the connection. BLE security mechanisms include device authentication, encryption, and secure pairing procedures, which protect data transmission between devices. However, the security of BLE can vary depending on the device's implementation [11]–[13].

Android applications, including companion apps for smart rings, require various permissions to access sensitive device resources. This is managed through runtime permissions, and special precautions must be taken to prevent unauthorized access, particularly for applications handling health data [14]. Rooting an Android device can expose vulnerabilities by granting high-level access, but developers can implement root detection to mitigate this risk [15]. Furthermore, adherence to data privacy regulations such as GDPR is crucial, ensuring user consent for data collection, secure data storage, and the right to be forgotten, especially regarding healthcare data [16].

2.2. Related Work

Fitness trackers, which collect sensitive health and activity data, have been the subject of multiple studies. Rahman et al. [17] conducted one of the first security analyses of fitness trackers by reverse-engineering the Fitbit ecosystem, revealing vulnerabilities such as unencrypted data transmission and exposure of user credentials in plaintext. Goyal et al. [18] later examined the security of companion apps and Bluetooth communication for fitness trackers, identifying privacy risks due to unchanging Bluetooth addresses and vulnerabilities in certificate validation. Similarly, Zhang et al. [19] found widespread failures in BLE security, including unencrypted data transmission and fake authentication. These studies reveal that fitness trackers often fail to meet basic security and privacy standards despite the sensitive nature of the data they handle. Alrawiet al. [20] proposed a general methodology to analyze security properties of home-based IoT devices, to which group the smart rings can be counted.

Mobile applications, especially those associated with wearables, also face significant security challenges. Gruber et al. [21] comprehensively analyzed childcare apps, uncovering the widespread use of insecure third-party libraries, excessive permissions, and poor privacy practices. In a similar study, Ioannidou and Sklavos [22] identified inappropriate permission requests and third-party data sharing in fitness tracker companion apps. These studies demonstrate that mobile applications, particularly those tied to health data, frequently expose users to privacy risks due to weak data handling practices.

Fitness trackers often rely on cloud servers for data storage and processing, introducing additional security concerns. Fereidooni et al. [23] assessed the security of fitness trackers that use cloud services, discovering that none of the devices analyzed implemented end-to-end encryption. This allowed the injection of falsified fitness data into the cloud. Wood et al. [24] found similar issues with medical IoT devices, which leaked sensitive patient data in cleartext during communication. These studies highlight the need for robust encryption and secure data handling practices in cloud-based fitness tracker ecosystems.

3. Method

This section outlines the methodology applied to test the security and privacy of the smart ring ecosystem. The tests are divided into three main components: the smart ring, companion app, and cloud.

3.1. Test Environment

The test environment comprises five smart rings, a Samsung Galaxy M12 mobile phone running Android 11, and a Lenovo T470 laptop running Kali Linux. Bluetooth communication is monitored using an nRF52840 dongle[1], while web traffic is intercepted with Burp Suite[2]. A high-level overview of the test framework can be found in figure 1. Various software tools, including Wireshark[3] to capture and analyze BLE traffic, MobSF[4] for automated static and dynamic security analysis, Frida[5] and Objection[6] as a mobile runtime exploitation kit, and Ghidra[7] for reverse-engineering the firmware of the smart rings, are employed

1. https://www.nordicsemi.com/Products/Development-hardware/nRF52840-Dongle
2. https://portswigger.net/burp/pro
3. https://www.wireshark.org
4. https://mobsf.live/about
5. https://frida.re
6. https://github.com/sensepost/objection
7. https://ghidra-sre.org

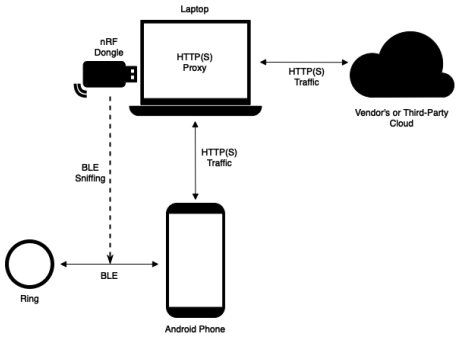

Figure 1: The test environment with the communication channels involved.

for security testing. For the privacy analysis, we use Exodus[8], which detects tracker libraries and permissions. The second tool that will be used for privacy analysis is RethinkDNS[9]. RethinkDNS is not primarily a tool but a DNS provider that blocks DNS requests to known tracker domains. To block tracking domains, the maintainers keep a large open-source list of known tracking domains.[10] When a tracker connects to its domain, the corresponding DNS request is monitored and visible through the companion app[11]. The companion apps will be installed on the test mobile phone.

3.2. Attacker Models

This paper's attacker model encompasses all three components within the smart ring ecosystem: the smart ring device, its companion app, and the cloud. This work considered both a passive attacker and an active attacker. Active attackers can sniff either data packets over the Bluetooth Low Energy channel or are placed within a Machine-in-the-Middle scenario in the network and can monitor HTTP traffic between the companion app and the cloud. The passive carries out attacks that do not alter communication content but rather eavesdrop on communication between two endpoints or on one.

BLE Attacker. The BLE attacker must be close to intercepting or interacting with Bluetooth communication. This model covers both passive eavesdropping and active attacks.

App Attacker. An app attacker requires initial access to the smartphone and can exploit app vulnerabilities through normal or root privileges. The attacker can manipulate or inspect app behavior, source code, and stored data.

Cloud Attacker. The attacker engages with cloud services remotely, focusing on passive data observation and active manipulation. Such an attacker can tamper with app-server communications or exploit cloud-based weaknesses.

8. https://exodus-privacy.eu.org
9. https://rethinkdns.com
10. https://github.com/serverless-dns/blocklists/tree/main
11. https://play.google.com/store/apps/details?id=com.celzero.bravedns

3.3. Test Case Selection Process

The test cases are selected based on the OWASP Mobile Application Security Testing Guide (MASTG), the Mobile Application Security Verification Standard (MASVS), and prior research on fitness trackers. The test cases are categorized into three main sections: test cases that focus on the security & privacy of the smart ring, then different test cases for the companion app, and finally test cases that focus on cloud security & privacy. Each section targets different aspects of the component to ensure coverage of the whole smart ring ecosystem. Table 1 overviews all test cases.

System	Test Case	Aspect	ID
Smart Ring	Missing End-to-End Encryption	Security	TC1
Smart Ring	Firmware Vulnerabilities	Security	TC2
Smart Ring	BLE Cleartext Communication	Security	TC3
Smart Ring	Modification of GATT Characteristics	Security	TC4
Smart Ring	Denial of Service	Security	TC5
Smart Ring	Usage of Public Bluetooth Device Addresses	Privacy	TC6
Smart Ring	Usage of Bluetooth General Discovery	Privacy	TC7
Smart Ring	No Usage of Bluetooth Secure Connections	Security	TC8
Smart Ring	Insecure Update Procedure	Security	TC9
Companion App	Support for Outdated Android Version	Security	TC10
Companion App	Missing Source Code Obfuscation	Security	TC11
Companion App	Insecure Data Storage	Privacy	TC12
Companion App	Privacy Policy Violation	Privacy	TC13
Companion App	Unauthorized Third-party Data Sharing	Privacy	TC14
Companion App	Missing or Insufficient Certificate Pinning	Security	TC15
Companion App	Hard-coded Credentials	Security	TC16
Companion App	Debugging Symbols	Security	TC17
Companion App	Invasive Tracking	Privacy	TC18
Companion App	Dangerous Permissions	Privacy	TC19
Companion App	Sensitive Information in Log Files	Privacy	TC20
Companion App	Insecure Cryptographic Algorithms	Security	TC21
Companion App	Insecure Third-Party Libraries	Security	TC22
Companion App	Missing Root Detection	Security	TC23
Cloud	No Protection of Data Integrity	Security	TC24
Cloud	Outdated or Insecure Software	Security	TC25
Cloud	Misconfigured Cross-Origin Resource Sharing (CORS)	Security	TC26
Cloud	Insecure Communication	Security	TC27
Cloud	Open Cloud Storage	Security	TC28
Cloud	Insecure Session Token	Security	TC29

TABLE 1: Summarized test suite for the smart ring ecosystem.

The nine smart ring test cases focus on vulnerabilities related to Bluetooth communication, firmware security, and the update procedure. As shown, the test cases target seven security issues such as missing encryption (TC1), cleartext communication (TC3), and denial of service attacks (TC5). Firmware vulnerabilities (TC2) and insecure update procedures (TC9) are also assessed. Furthermore, two test cases analyze the device's privacy. The 14 companion app

tests examine permissions, privacy violations, and reverse engineering resistance. The six privacy-related test cases focus on different privacy aspects of mobile applications, such as insecure data storage (TC12) and privacy policy violations (TC13). Eight security test cases ensure the application is secure from active or passive attackers. The attacks we tested, such as certificate pinning (TC15) and root detection (TC23), are also evaluated. Six cloud test cases focus on server communication, user data handling, and the security of the cloud infrastructure used. The test cases include examining insecure communication protocols (TC27), open cloud storage (TC28), and insecure session tokens (TC29). Each test ensures that cloud services are configured securely to protect user data from unauthorized access or manipulation.

4. Results

The smart ring ecosystem's systematic security and privacy analysis revealed a total of 74 (average: 14.8) vulnerabilities. These include 23 (average: 4.6) severe vulnerabilities that significantly risk sensitive data and users' privacy. We found security and privacy vulnerabilities in every ring and its companion app. Table 2 shows all test case results of our work and gives the overall number of vulnerabilities per ring and critical vulnerabilities per ring.

4.1. Critical Findings

The following subsections describe the most critical findings we identified during our analysis. A complete overview of the identified vulnerabilities can be found in our work's supplementary material. It highlights the most critical vulnerabilities which pose a significant risk to sensitive data. A vulnerability is classified as critical when an attacker can exploit it with minimal prerequisites while achieving a substantial impact [25].

4.1.1. BLE Cleartext Communication. R2 transmits health and fitness data without encryption over the BLE link. An attacker near the victim can intercept the communication and capture sensitive data like current vital parameters or stress levels.

4.1.2. Unauthenticated Firmware Update. R4 allows firmware installation without authentication. A Generic Attribute Profile (GATT) enables custom firmware installation without pairing or bonding. This unintentional exposure of a test function allows attackers to control the ring's behavior by installing malicious firmware.

4.1.3. Missing Certificate Pinning. The R1 app lacks certificate pinning, making it vulnerable to MitM attacks. This allows an attacker to inspect and manipulate traffic. Compounding this issue, the app's backend server uses a self-signed certificate and outdated TLS versions. While fitness data isn't synchronized to the cloud, login credentials could be compromised, and attackers can install custom firmware by manipulating app-server communication.

Test Case	R1	R2	R3	R4	R5
TC1: Missing End-to-End Encryption	-	-	×	-	-
TC2: Firmware Vulnerabilities	◯	◯	◯	-	◯
TC3: BLE Cleartext Communication	◯	×	◯	◯	◯
TC4: Modification of GATT Characteristics	◯	◯	×	×	◯
TC5: Denial of Service	×	◯	◯	◯	◯
TC6: Usage of Public Bluetooth Device Addresses	×	×	×	×	×
TC7: Usage of Bluetooth General Discovery	×	×	◯	×	×
TC8: No Usage of Bluetooth Secure Connections	×	×	×	×	◯
TC9: Insecure Update Procedure	×	×	◯	×	◯
TC10: Support for Outdated Android Version	×	×	×	×	×
TC11: Missing Source Code Obfuscation	×	◯	×	◯	◯
TC12: Insecure Data Storage	×	×	×	×	×
TC13: Privacy Policy Violation	◯	×	×	×	×
TC14: Unauthorized Third-party Data Sharing	◯	×	×	×	×
TC15: Missing or Insufficient Certificate Pinning	×	×	×	◯	◯
TC16: Hard-coded Credentials	◯	×	◯	◯	◯
TC17: Debugging Symbols	×	×	◯	◯	◯
TC18: Invasive Tracking	◯	×	◯	×	×
TC19: Dangerous Permissions	◯	◯	◯	×	×
TC20: Sensitive Information in Log Files	◯	◯	×	◯	×
TC21: Insecure Cryptographic Algorithms	◯	×	◯	◯	◯
TC22: Insecure Third-Party Libraries	×	×	×	×	◯
TC23: Missing Root Detection	×	×	×	◯	×
TC24: No Protection of Data Integrity	-	-	×	-	-
TC25: Outdated or Insecure Software	×	×	◯	◯	◯
TC26: Misconfigured Cross-Origin Resource Sharing (CORS)	◯	◯	×	◯	◯
TC27: Insecure Communication	×	×	×	×	×
TC28: Open Cloud Storage	-	-	◯	◯	◯
TC29: Insecure Session Token	×	-	◯	×	◯
Overall number of vulnerabilities	15	18	16	14	11
Overall number of critical vulnerabilities	4	6	4	5	4

TABLE 2: Summary of test case results, indicating vulnerabilities (×), non-vulnerabilities (◯), and non-applicable tests (-).

4.1.4. Usage of Deprecated Cryptographic Algorithms. Blaupunkt's use (R2) of the DES algorithm is concerning, as DES was deemed insecure and withdrawn in 2005[12]. Additionally, the DES key is hard-coded into the app's source code. While it was impossible to analyze the data flow due to code obfuscation thoroughly, this vulnerability presents a severe cryptographic risk.

4.1.5. Not Asking for User Consent. R2, R3, R4, and R5 do not request user consent before collecting and sharing data with third parties. This is a major privacy violation, as users are unaware of their data being processed and cannot control its use.

4.1.6. Insecure Storage. All tested apps fail to protect data at rest adequately. They store health, fitness, and authentication data in unencrypted databases, log files, and configuration files, making them vulnerable to attackers with initial access to the Android system.

12. https://web.archive.org/web/20080625202735/http://csrc.nist.gov/publications/fips/05-9945-DES-Withdrawl.pdf

4.1.7. Usage of Static BDAs. None of the manufacturers implemented private BDAs for their rings, allowing attackers to track user locations based on the static BDA, which is constantly broadcasted and uniquely identifies the device [26].

4.1.8. Usage of Deprecated TLS Versions. The backend servers of all tested apps use outdated and insecure TLS versions (1.0 and 1.1). These versions are deprecated as of 2019 under RFC 8996. They risk losing confidentiality and integrity in HTTPS communication because they rely on insecure hashing mechanisms, especially SHA-1 [27], [28].

4.2. Strengths

It is important to note that several test cases did not uncover any vulnerabilities in the rings, apps, or cloud backends. The absence of these vulnerabilities suggests that developers are aware of potential weaknesses and prioritize addressing them more effectively. Additionally, certain test cases could not be conducted due to the presence of protection mechanisms. The following two test cases did not reveal vulnerabilities:

- Firmware Vulnerabilities
- Open Cloud Storage

No firmware vulnerabilities were found due to the complexities of analyzing SoC firmware. Most existing firmware analysis tools focus on embedded Linux systems with file systems, while the firmware in this work requires a different approach. In terms of cloud storage, the growing adoption of well-established cloud solutions, such as AWS S3 and Firebase, has maybe led to better configuration practices as many work pointed out misconfiguration issues are the most prevalent for cloud security [29], [30]. Furthermore, an in-depth analysis of security and privacy of the respective cloud infrastructure would need to be approved by the vendor (see section 6 for details).

5. Discussion

This section interprets the results in a broader context. It highlights risks and countermeasures.

Our analysis of related work in section 2.2 resulted in different threats observed in different ecosystem components. We did not expect that there would be so many severe vulnerabilities in the rings and applications. With the introduction of the new device category of smart rings, simple vulnerabilities again found their way into the devices and apps. Developers did not learn from the issues of previous wearable developers, highlighting the need for developer-centric security approaches and secure development APIs [31]. It furthermore underscores the need for thorough security and privacy testing, even if some vulnerabilities seem too essential to be expected, like outdated TLS versions.

This work has uncovered several weaknesses that were not expected on a large scale due to their simplicity. Some vulnerabilities are found in every ring, and every application is concerning, as this implies that security measures against these types of vulnerabilities represent a pattern of failure by the responsible vendors. Examples are the usage of static Bluetooth Device Addresses (BDA), the support for outdated Android versions, and privacy policy violations. It can be assumed that the vulnerabilities arose because vendors were more concerned with getting a cheap device to market quickly rather than investing in security and privacy, as implementing rigorous security and privacy policies is likely to increase the time-to-market for new products even though it provides long-term benefits [32]. Vulnerabilities such as not using BLE Secure Connections, BLE cleartext communication, or an open BLE update channel violate common security practices [33].

5.1. Implications

The vulnerabilities pose several risks to smart ring users, manufacturers, and third parties. The following scenarios highlight imminent threats:

1) **Financial Loss**: Fitness data can be manipulated, risking financial losses for platforms rewarding sporting activity. This can occur via firmware manipulations, modifications of unencrypted databases, or during app-cloud synchronization. Also there is a risk for the vendor of being sued due to not adhering to the GDPR or other privacy regulations like the CCPA.
2) **Spying**: Static BDAs enable tracking of smart ring users. Attackers can identify and track users based on the unique BDA of the ring.
3) **Cloud Independence**: With custom firmware installation, attackers can set up a custom companion app and cloud, bypassing the provider's subscription-based cloud services and collecting data from unauthorized users.
4) **Damage**: Installing faulty custom firmware without physical access could damage the ring, causing financial and reputational damage for both users and vendors.
5) **Legal Implications**: Vendors risk violating GDPR regulations. The new *Cyber Resilience Act* (CRA) further regulates security measures for products with digital components, which these smart rings do not currently comply with.
6) **Technology Mediated Abuse**: Rings have a special meaning in various cultures, especially among couples, and can serve as a symbolic warning for other, maybe interested humans [34]. If a partner wears a ring and the ring is connected to an application on the other partner's smartphone, this poses a possibility for technology-mediated abuse [35].

5.2. Recommendations

Vendors should address the vulnerabilities found and patch the security and privacy issues. The following technical and organizational recommendations are provided to enhance the security and privacy of smart trackers in general and smart rings specifically:

1) **BLE Security**: Follow Bluetooth SIG guidelines for secure BLE-enabled devices. This includes enabling BLE Secure Connections, disabling general advertisement post-pairing, and securing GATT characteristics.
2) **End-to-End Encryption**: Encrypt data at the ring level to protect against attacks along the data transit route and minimize risks of data loss and manipulation.
3) **Following Standards**: Implement security controls based on the OWASP MASV[13]. This can prevent vulnerabilities like missing certificate pinning, hardcoded credentials, deprecated cryptographic algorithms, and unencrypted data storage.
4) **Tests, Audits, and Reviews**: Regular security audits and privacy reviews should be integrated into product development under the principle of "Security by Design."

5.3. Implications for Counter Threat Intelligence

The vulnerabilities discovered in this study also have significant implications for counter-threat intelligence (CTI). CTI focuses on understanding and mitigating threat actors' tactics, techniques, and procedures (TTPs) [36]. The findings from this work highlight several gaps in smart ring security, which can inform proactive defense strategies and improve response frameworks for both vendors and security teams. Arguably, the need for such solutions will rise with the rising market share of smart rings. The vulnerabilities identified, such as cleartext communication over BLE, insecure firmware updates, and the lack of certificate pinning, present opportunities for attackers to exploit users' and companies sensitive data with minimal effort. These findings emphasize the need for proactive detection mechanisms in company networks that allow devices from users to connect to the network, maybe also through a company-owned gateway device. For example, security teams can deploy advanced packet sniffing detection tools to monitor BLE traffic in public spaces and identify potential attackers intercepting unencrypted data. Connecting insecure BLE devices such as smart rings over an application to a company network is also not recommended by NIST [33]. As part of counter-threat intelligence, focusing on behavior-based anomaly detection—such as monitoring for unusual traffic patterns between wearables and cloud servers—can help mitigate these attacks before they succeed. Threat hunters can integrate these findings into their detection models to prioritize security misconfigurations in connected devices. CTI frameworks should include scenarios where attackers exploit outdated encryption methods or bypass inadequate data storage protections. As noted by Bodeauet al. [37], mapping vulnerabilities to known attacker techniques (e.g., MITRE ATT&CK) can help responders anticipate and contain incidents more effectively.

6. Limitations & Ethics

This section describes the limitations of our approach as well as ethical considerations.

13. https://mas.owasp.org/MASVS/

6.1. Limitations

Several limitations were encountered during testing. The apps of R4 and R5 could not be thoroughly tested due to strong certificate pinning and root detection mechanisms. These security features could not be bypassed, limiting the ability to analyze app $\leftrightarrow$ cloud communication. Given the 29 test cases conducted on five rings, 145 tests were required. Time constraints limited the depth of specific tests, such as binary patching to bypass security mechanisms. Additionally, existing firmware analysis tools were not designed for the type of firmware used in these rings, making manual vulnerability analysis extremely difficult.

6.2. Ethical Considerations

While no ethics committee covers this type of work at the organizations involved in this research, strict laws and privacy regulations are in place, and we discussed our analysis plan with several peers and professionals to ensure proper design. We conducted our work according to ethical best practices detailed in the *Menlo Report* [38]. For example, we used artificial information for the user accounts and handled the required data securely in access control and (encrypted) storage. Only the researchers who conducted the static and dynamic analysis had access to the test systems. These researchers are professional penetration testers who adhere to strict "ethical hacking" guidelines. We always used a minimally invasive approach and never accessed, altered, or interacted with other user's data.

Responsible Disclosure. We responsibly disclosed the issues to the vendors on the 27th of September 2024 after asking for specific contacts via mail on 12th of September as we could not find a contact according to RFC 9116 [39]. Four vendors responded three weeks after responsibly disclosing our findings and stated they would address them. We sent each vendor a complete report of our findings. Each report contained all findings for the specific ecosystem, including hints on how to fix each issue. Nevertheless, we hope our findings make vendors and developers aware of the security and privacy issues they introduce.

7. Conclusion

This paper evaluated the security and privacy of five smart rings through a test suite of 29 cases covering BLE, mobile apps, firmware, and cloud backends. The tests applied techniques such as BLE sniffing, denial of service, firmware modification, and certificate pinning bypassing to uncover critical vulnerabilities. Additionally, privacy violations were prevalent—three apps failed to ask for consent, and four privacy policies lacked transparency regarding third-party providers. Smart rings threaten corporate environments when connected to smartphones, posing potential entry points for attackers.

References

[1] DataHorizon Research, "Smart Ring Market," 2023. Available at: https://datahorizzonresearch.com/smart-ring-market-2390.

[2] M. Kurz, R. Gstoettner, and E. Sonnleitner, "Smart rings vs. smartwatches: Utilizing motion sensors for gesture recognition," *Applied Sciences*, vol. 11, no. 5, p. 2015, 2021.

[3] M. Gladiš, M. Mesarčík, and N. Slosiarová, "Advising ai assistant: ethical risks of oura smart ring," *AI and Ethics*, pp. 1–13, 2024.

[4] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, and A. Zanella, "Iot: Internet of threats? a survey of practical security vulnerabilities in real iot devices," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8182–8201, 2019.

[5] V. G. Motti and K. Caine, "Users' privacy concerns about wearables: impact of form factor, sensors and type of data collected," in *Financial Cryptography And Data Security: FC 2015 International Workshops, BITCOIN, WAHC, And Wearable, San Juan, Puerto Rico, January 30, 2015, Revised Selected Papers*, pp. 231–244, Springer, 2015.

[6] A. Rostami, M. Vigren, S. Raza, and B. Brown, "Being hacked: Understanding victims' experiences of {IoT} hacking," in *Eighteenth Symposium on Usable Privacy and Security (SOUPS 2022)*, pp. 613–631, 2022.

[7] H. Jayakumar, K. Lee, W. S. Lee, A. Raha, Y. Kim, and V. Raghunathan, "Powering the internet of things," in *Proceedings of the 2014 international symposium on Low power electronics and design*, pp. 375–380, 2014.

[8] L. Marin, M. Piotr Pawlowski, and A. Jara, "Optimized ecc implementation for secure communication between heterogeneous iot devices," *Sensors*, vol. 15, no. 9, pp. 21478–21499, 2015.

[9] M. Schöffel, F. Lauer, C. C. Rheinländer, and N. Wehn, "Secure iot in the era of quantum computers—where are the bottlenecks?," *Sensors*, vol. 22, no. 7, p. 2484, 2022.

[10] J. Classen, D. Wegemer, P. Patras, T. Spink, and M. Hollick, "Anatomy of a Vulnerable Fitness Tracking System: Dissecting the Fitbit Cloud, App, and Firmware," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 2, mar 2018.

[11] Bluetooth SIG, "Bluetooth® Technology Website." Available at: https://www.bluetooth.com/learn-about-bluetooth/tech-overview/.

[12] Bluetooth SIG, "Bluetooth Core Specification v5.4." Available at: https://www.bluetooth.com/de/specifications/specs/core-specification-5-4/.

[13] M. Cäsar, T. Pawelke, J. Steffan, and G. Terhorst, "A survey on bluetooth low energy security and privacy," *Computer Networks*, vol. 205, p. 108712, 2022.

[14] E. Alepis and C. Patsakis, "Unravelling security issues of runtime permissions in android," *Journal of Hardware and Systems Security*, vol. 3, pp. 45–63, 2019.

[15] S.-T. Sun, A. Cuadros, and K. Beznosov, "Android rooting: Methods, detection, and evasion," in *Proceedings of the 5th Annual ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices*, pp. 3–14, 2015.

[16] M. Fan, L. Yu, S. Chen, H. Zhou, X. Luo, S. Li, Y. Liu, J. Liu, and T. Liu, "An empirical evaluation of gdpr compliance violations in android mhealth apps," in *2020 IEEE 31st international symposium on software reliability engineering (ISSRE)*, pp. 253–264, IEEE, 2020.

[17] M. Rahman, B. Carbunar, and U. Topkara, "Secure management of low power fitness trackers," *IEEE Transactions on Mobile Computing*, vol. 15, p. 447–459, Feb. 2016.

[18] R. Goyal, N. Dragoni, and A. Spognardi, "Mind the tracker you wear: a security analysis of wearable health trackers," in *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, SAC '16, (New York, NY, USA), p. 131–136, Association for Computing Machinery, 2016.

[19] Q. Zhang and Z. Liang, "Security analysis of bluetooth low energy based smart wristbands," in *2017 2nd International Conference on Frontiers of Sensors Technologies (ICFST)*, pp. 421–425, 2017.

[20] O. Alrawi, C. Lever, M. Antonakakis, and F. Monrose, "Sok: Security evaluation of home-based iot deployments," in *2019 IEEE symposium on security and privacy (sp)*, pp. 1362–1380, IEEE, 2019.

[21] M. Gruber, C. Höfig, M. Golla, T. Urban, and M. Große-Kampmann, ""we may share the number of diaper changes": A privacy and security analysis of mobile child care applications," *Proceedings on Privacy Enhancing Technologies*, 2022.

[22] I. Ioannidou and N. Sklavos, "On general data protection regulation vulnerabilities and privacy issues, for wearable devices and fitness tracking applications," *Cryptography*, vol. 5, no. 4, 2021.

[23] H. Fereidooni, T. Frassetto, M. Miettinen, A.-R. Sadeghi, and M. Conti, "Fitness trackers: Fit for health but unfit for security and privacy," in *2017 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, pp. 19–24, 2017.

[24] D. Wood, N. Apthorpe, and N. Feamster, "Cleartext data transmissions in consumer iot medical devices," in *Proceedings of the 2017 Workshop on Internet of Things Security and Privacy*, ACM, Nov. 2017.

[25] J. Franklin, C. Wergin, H. Booth, *et al.*, "Cvss implementation guidance," *National Institute of Standards and Technology, NISTIR-7946*, 2014.

[26] T. Issoufaly and P. U. Tournoux, "Bleb: Bluetooth low energy botnet for large scale individual tracking," in *2017 1st International Conference on Next Generation Computing Applications (NextComp)*, pp. 115–120, IEEE, 2017.

[27] K. Moriarty and S. Farrell, "Deprecating TLS 1.0 and TLS 1.1," *Internet Engineering Task Force, RFC*, vol. 8996, 2021.

[28] K. Bhargavan and G. Leurent, "Transcript collision attacks: Breaking authentication in tls, ike, and ssh," in *Network and Distributed System Security Symposium–NDSS 2016*, 2016.

[29] S. Pletinckx, K. Borgolte, and T. Fiebig, "Out of sight, out of mind: Detecting orphaned web pages at internet-scale," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 21–35, 2021.

[30] A. Continella, M. Polino, M. Pogliani, and S. Zanero, "There's a hole in that bucket! a large-scale analysis of misconfigured s3 buckets," in *Proceedings of the 34th Annual Computer Security Applications Conference*, pp. 702–711, 2018.

[31] M. Green and M. Smith, "Developers are not the enemy!: The need for usable security apis," *IEEE Security & Privacy*, vol. 14, no. 5, pp. 40–46, 2016.

[32] U. Tariq, I. Ahmed, A. K. Bashir, and K. Shaukat, "A critical cybersecurity analysis and future research directions for the internet of things: a comprehensive review," *Sensors*, vol. 23, no. 8, p. 4117, 2023.

[33] J. Padgette, K. Scarfone, and L. Chen, "Guide to bluetooth security," *NIST special publication*, vol. 800, no. 121, pp. 657–696, 2017.

[34] E. S. Person, "Symbolism in love and sex," in *Language, symbolization, and psychosis*, pp. 202–218, Routledge, 2018.

[35] S. Parkin, T. Patel, I. Lopez-Neira, and L. Tanczer, "Usability analysis of shared device ecosystem security: informing support for survivors of iot-facilitated tech-abuse," in *Proceedings of the new security paradigms workshop*, pp. 1–15, 2019.

[36] G. Sakellariou, P. Fouliras, I. Mavridis, and P. Sarigiannidis, "A reference model for cyber threat intelligence (cti) systems," *Electronics*, vol. 11, no. 9, p. 1401, 2022.

[37] D. Bodeau and R. Graubart, "Cyber resiliency design principles: Selective use throughout the engineering lifecycle," tech. rep., MITRE Corporation, McLean, VA, USA, 2018.

[38] US Department of Homeland Security, "The Menlo Report: Ethical Principles Guiding Information and Communication Technology Research," Aug. 2012. https://www.caida.org/publications/papers/2012/menlo_report_actual_formatted/, as of November 3, 2024.

[39] E. Foudil and Y. Shafranovich, "Rfc 9116: A file format to aid in security vulnerability disclosure," 2022.