Department of Health and Human Services
Office of Inspector General
Office of Audit Services

December 2024 | A-18-22-09008

Some HHS Requirements for Vetting Mobile Apps Were Not Followed Prior to the Release of the AHRQ Question Builder App

REPORT HIGHLIGHTS

December 2024 | A-18-22-09008

Some HHS Requirements for Vetting Mobile Apps Were Not Followed Prior to the Release of the AHRQ Question Builder App

Why OIG Did This Audit

- HHS and its operating divisions offer mobile applications (apps) to deliver services and information to people.
- Security vulnerabilities that may exist in HHS mobile apps could be potentially exploitable and lead to compromise of the underlying mobile device or sensitive data on the device or connected cloud systems.
- The Agency for Healthcare Research and Quality (AHRQ) Question Builder app's purpose is to help patients and caregivers prepare for and get the most out of medical appointments.
- We assessed the app's cybersecurity controls between March and April 2022 to determine whether AHRQ followed required security standards and policies for developing and vetting the mobile app before it was released.

What OIG Found

- The AHRQ Question Builder app had cybersecurity controls that were generally effective in preventing our simulated cyberattacks.
- The AHRQ Question Builder app did not comply with a National Institute of Standards and Technology security control to provide only the necessary functionality for an app to operate.
- AHRQ's Mobile Application Development Policy did not include all standards and requirements that project officers must follow before submitting a mobile app to an app store.

What OIG Recommends

We made three recommendations to AHRQ, including that it reassess the Question Builder app to determine if unnecessary functionality should be removed or disabled and update the AHRQ Mobile Application Development Policy to include requirements related to least functionality and secure coding. The full recommendations are in the report.

In written comments on our draft report, AHRQ indicated that it agreed with our findings and described actions it has taken and plans to take to address our recommendations.

TABLE OF CONTENTS

INTRODUCTION

WHY WE DID THIS AUDIT

At the time of our audit, the Department of Health and Human Services (HHS) and its operating divisions (OpDivs) offered 56 mobile applications (apps) for installation on both iOS and Android devices that they developed to deliver services and information to citizens. Some of these apps could store and transmit personally identifiable information (PII) and individually identifiable health information. Security vulnerabilities that may exist in HHS mobile apps could be potentially exploitable and lead to compromise of the underlying mobile device or sensitive data on the device or connected cloud systems.[1]

This audit is one in a series of audits assessing the security of HHS and its OpDivs' mobile apps. For this audit, we conducted a security assessment of the Agency for Healthcare Research and Quality's (AHRQ) Question Builder app (version 1.0.11 (iOS) and version 2.0.1 (Android)) in accordance with security testing guidelines outlined by the National Institute of Standards and Technology (NIST). We selected the Question Builder app for testing based on the number of installations on mobile devices in addition to it having one of the highest risk scores compared to other AHRQ apps, as determined by the AppVet system.[2] Specifically, the AppVet system determined that the Android version of the AHRQ Question Builder app had a medium vulnerability level.[3] We chose to test both the Android and iOS versions of the app.

OBJECTIVES

Our objectives were to determine whether:

- the AHRQ Question Builder app had cybersecurity controls that were effective in preventing certain cyberattacks and

- AHRQ followed the required security standards and policies for developing and vetting mobile applications.[4]

---

[1] Cloud systems are the servers, virtual machines, storage, networks, operating systems, or services used to deliver computing services over the internet (the cloud).

[2] AppVet is a web application service provided by the Department of Homeland Security for managing and automating the process for vetting apps. AppVet facilitates the vetting workflow by providing an intuitive user interface for submitting and testing apps, managing reports, and assessing risk.

[3] No AHRQ apps were determined to have a high vulnerability level.

[4] An app-vetting process is a sequence of activities performed by an organization to determine if a mobile app conforms to the organization's app security requirements. See https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-163r1.pdf. Accessed on May 15, 2024.

BACKGROUND

Agency for Healthcare Research and Quality

AHRQ is the lead Federal agency charged with improving the safety and quality of healthcare for all Americans. AHRQ develops the knowledge, tools, and data needed to improve the healthcare system and help consumers, healthcare professionals, and policymakers make informed health decisions. AHRQ's mission is to produce evidence to make healthcare safer, higher quality, more accessible, equitable, and affordable, and to work within HHS and with other partners to make sure that the evidence is understood and used.

HHS Mobile Applications

Although mobile apps designed to run on mobile devices such as smartphones and tablets often provide users with similar services to those accessed on computers, they are distinct from applications designed to run on computers and web applications that run on mobile web browsers. Development, availability, and use of mobile apps have increased exponentially since being introduced in the 2000s. As of 2022, over 2 million mobile apps were available on the Google Play Store, and over 4 million apps were available on the Apple App Store. The use of mobile apps to access medical records has also increased significantly from 2020 through 2022.[5] With the increased use of mobile apps, there has also been an increase in mobile app security threats.

Mobile apps that do not properly secure, validate,[6] and sanitize data are vulnerable to attacks and increase the risk of unauthorized access of sensitive data, manipulation of app functionality, and potential compromise of the underlying mobile device. Apps that do not have adequate privacy controls can potentially expose PII (e.g., names, addresses, Social Security numbers, and email addresses) and individually identifiable health information (e.g., medical history, health insurance information, and test results).

At the time of our audit, HHS and its OpDivs offered 56 different apps across various platforms, 3 of which were offered by AHRQ. At a minimum, when developing and deploying apps, HHS and its OpDivs must follow HHS Policy Number HHS-OCIO-OES-2019-08-005, "HHS Policy for Software Development Secure Coding Practices," (herein referred to as HHS's secure coding

[5] Fox, Andrea, *Healthcare IT News*, "Patient preferences for accessing medical data are shifting, says ONC." Available online at: https://www.healthcareitnews.com/news/patient-preferences-accessing-medical-data-are-shifting-says-onc. Accessed on Oct. 16, 2023.

[6] There are various types of validation, including input validation (a frequently used technique that checks for potentially dangerous inputs) and output validation (a technique used to prevent data corruption or presentation vulnerabilities through the execution of malicious code).

policy). Additionally, all Federal civilian agencies must implement security controls described in NIST Special Publication (SP) 800-53.

AHRQ Question Builder App

The AHRQ Question Builder app was initially released in March 2019 with the purpose of helping patients and caregivers prepare for medical appointments and maximize the value of the time they have with their providers. With the app, users can access consumer education materials and videos, as well as input information they wish to share with providers during their visits using prepopulated questions or typing in questions manually. Users can also prepare and organize questions by type of medical encounter and take photos of insurance cards as well as medication labels. Once the information is added, users can send the information to their default calendar or email app on the same device that the Question Builder app is installed on to save or transmit the information for use during a future medical appointment.

The Question Builder app does not store PII, individually identifiable health information, or other sensitive information within local storage on a user's device. Rather, it temporarily stores this information within its temporary memory and is designed to be used for preparing questions for a single visit. Once the app is closed, the information is removed from the app's temporary memory and a new session is created the next time the app is opened. The Question Builder app is available in the Google Play Store and Apple App Store and has been downloaded more than 1,000 times, as of April 2024.

HOW WE CONDUCTED THIS AUDIT

We conducted this audit of the AHRQ Question Builder app between March and April 2022. Our work focused on the mobile app's security controls. We also determined whether AHRQ followed NIST, HHS, and AHRQ security standards and policies when developing and vetting the AHRQ Question Builder app.

As part of this work, we contracted with NowSecure to assist in conducting the security assessment of the app. NowSecure provided subject matter expertise throughout the assessment of the app. We performed testing in accordance with the agreed-upon Rules of Engagement (ROE) document signed in March 2022 by OIG, NowSecure, and AHRQ. We provided detailed documentation about our preliminary findings to AHRQ in advance of issuing our draft report.

We conducted this performance audit in accordance with generally accepted government auditing standards. Those standards require that we plan and perform the audit to obtain sufficient, appropriate evidence to provide a reasonable basis for our findings and conclusions based on our audit objectives. We believe that the evidence obtained provides a reasonable basis for our findings and conclusions based on our audit objectives.

Appendix A contains the details of our audit scope and methodology, Appendix B describes the tools we used to conduct the audit, and Appendix C contains HHS and Federal requirements.

FINDINGS

The AHRQ Question Builder App had cybersecurity controls in place that were generally effective in preventing our simulated cyberattacks. However, we found that the AHRQ Question Builder App (1) did not comply with the NIST SP 800-53, Revision 5, CM-7 Least Functionality security control and (2) did not follow required HHS security policy for vetting the app before it was released for public use. This occurred because the AHRQ Mobile Application Development Policy followed by project officers and app developers lacked some necessary requirements. As a result, the app was released with unnecessary functionality, potentially exploitable vulnerabilities, and missing security functionality that could be exploited to gain access to data on a user's mobile device or to execute malicious code that may steal, alter, or destroy the user's private information.

AHRQ DID NOT COMPLY WITH NIST LEAST FUNCTIONALITY SECURITY CONTROL

Based on our security assessment, we determined that the AHRQ Question Builder app had some functionality and privileges built in that may not be necessary for it to operate as designed and may make it susceptible to exploitation. Per NIST SP 800-53, Revision 5, CM-7 Least Functionality security control, the app should be configured to provide only the organization's defined mission-essential capabilities and unnecessary or unused functionality should be removed or disabled where feasible.

AHRQ did not limit the Question Builder app's functionality as required by the NIST CM-7 security control because the AHRQ Mobile Application Development Policy did not require project officers or app developers to assess mobile apps for unnecessary or unused functionality and remove such functionality before submitting it to an app store.

As a result, the Question Builder app was deployed with unnecessary functionality and privileges that—under certain conditions—could be exploited to gain access to data on a user's mobile device or to execute malicious code that may steal, alter, or destroy the user's private information.

AHRQ DID NOT FOLLOW REQUIRED HHS SECURITY POLICY FOR VETTING THE QUESTION BUILDER APP

AHRQ did not follow some HHS secure coding policy requirements for vetting the security of the Question Builder app before releasing it for public use. Some app-vetting of the AHRQ Question Builder app was performed by mobile app store vendors as part of the vendors'

mobile app review processes.[7, 8] However, AHRQ did not adequately vet the security of the AHRQ Question Builder app in accordance with two HHS secure coding policy requirements before submitting it to app stores. Specifically, section 6.1 of HHS's secure coding policy requires HHS OpDivs, when developing and maintaining software, to (1) ensure software is free from exploitable code vulnerabilities and (2) ensure the software, when executed, will provide security functionality as intended (see Appendix C).

AHRQ did not adequately vet the security of the app in accordance with HHS's secure coding policy because its project officers and app developers followed the AHRQ Mobile Application Development Policy which was missing important requirements that are specified in the HHS policy. Specifically, the AHRQ policy did not require project officers to vet a mobile app for compliance with the HHS secure coding policy before submitting it to an app store.

As a result of AHRQ not performing effective vetting of the Question Builder app before releasing it for public use, the app was deployed with potentially exploitable vulnerabilities and missing security functionality that could be exploited to gain access to data on a user's mobile device or to execute malicious code that may steal, alter, or destroy the user's private information.

RECOMMENDATIONS

We recommend that the Agency for Healthcare Research and Quality:

- reassess the Question Builder app to determine if the unnecessary functionality and privileges built into the app can and should be removed or formally assess, document, and accept the risk of not removing them;

- update the AHRQ Mobile Application Development Policy to require project officers and app developers to assess AHRQ mobile apps for unnecessary or unused functionality and remove or disable such functionality where feasible before submitting it to an app store and establish a procedure to ensure adherence to these requirements; and

- update the AHRQ Mobile Application Development Policy to require vetting the security of all AHRQ mobile apps for compliance with the HHS secure coding policy requirements and correcting any security vulnerabilities identified before releasing a mobile app to app stores for public use and establish a procedure to ensure adherence to these requirements.

---

[7] For example, details on the technical, content, and design criteria used by Apple for reviewing apps are available online at https://developer.apple.com/distribute/app-review. Accessed on Apr. 30, 2024.

[8] Mobile app store vendors typically assess mobile apps based on their own requirements (e.g., compatibility, usefulness, and legal). Although their app-vetting can also identify certain security and privacy vulnerabilities, such as the existence of malware in an app or an app that collects personal information without notifying the user, this app-vetting may not be sufficient to comply with HHS's secure coding policy requirements.

AHRQ COMMENTS AND OFFICE OF INSPECTOR GENERAL RESPONSE

In written comments on our draft report, AHRQ did not explicitly indicate concurrence or nonconcurrence with our recommendations. However, it agreed with our findings and described actions it has taken and plans to take to address our recommendations. Specifically, AHRQ stated that it updated its AHRQ Mobile Application Development Policy and is taking corrective actions related to least functionality in the Question Builder app. We are encouraged by AHRQ's response and look forward to receiving and reviewing information from AHRQ on the actions it has taken in response to our recommendations.

AHRQ's comments are included in their entirety as Appendix D.

APPENDIX A: AUDIT SCOPE AND METHODOLOGY

SCOPE

We conducted a security assessment of two variants of the AHRQ Question Builder app (iOS version 1.0.11 and Android version 2.0.1) between March and April 2022. The security assessment of the two variants focused on assessing select required NIST and HHS security controls, as specified within the audit plan.

In our security assessment, we evaluated the operating effectiveness of NIST SP 800-53, Revision 5 controls related to the AHRQ Question Builder mobile app and whether AHRQ followed Federal and HHS security requirements when developing and vetting the app. We did not assess all internal control components and principles. The information technology control deficiency we identified is described in the "Findings" section of this report. However, our audit may not have disclosed all control deficiencies that may have existed with the app at the time of our fieldwork. We reviewed various NIST SP 800-53, Revision 5 controls, including but not limited to:

- SI-2 Flaw Remediation,
- SA-11 Developer Testing and Evaluation,
- IA-5 Authenticator Management,
- SC-8 Transmission Confidentiality and Integrity, and
- SI-10 Information Input Validation.

We performed our work remotely between March 2022 and April 2022.

METHODOLOGY

We relied on the work of specialists to assist with the series of OIG audits utilizing vulnerability scanning and network and code analysis tools. OIG contracted with NowSecure to conduct the security assessment of one AHRQ mobile application. NowSecure provided subject matter experts who conducted the security assessment of the application identified in the ROE document. OIG oversaw the work to ensure that all objectives were met, and that testing was performed in accordance with Government auditing standards and the ROE document.

Our testing focused on one publicly available AHRQ mobile app (Question Builder). To accomplish our objectives, OIG and NowSecure prepared the ROE document that outlined the general rules, logistics, and expectations for the mobile app security assessment. AHRQ officials provided a signed ROE document indicating that AHRQ agreed with the rules to be followed during our testing.

In March 2022, we began to test the AHRQ mobile app. We performed various mobile application analysis techniques to determine if the applications were susceptible to exploits by an external attacker.

NowSecure performed procedures, including:

- *Data-at-rest analysis:* Analysts install the application on real devices with the target mobile operating system (iOS or Android) and conduct a forensic analysis of the device for specific application or data storage vulnerabilities.

- *Data-in-transit analysis:* Analysts operate all aspects of the application as a user would and attempt to detect vulnerabilities via network communications. Taking the position of an attacker, analysts will attempt to compromise the network in a variety of ways.

- *Static binary analysis:* Analysts use open source and proprietary tools to evaluate the fully compiled mobile app program and discover flaws in the logic that could result in a vulnerability.

- *Reverse engineering:* This process includes the use of tools that help analyze the mobile app program in an effort to understand the source code. This activity reveals what additional information or exploitation vectors an attacker could discover by analyzing the source code.

We also identified applicable Federal security requirements for mobile apps and determined whether AHRQ followed these requirements when developing and vetting the app.

We reviewed AHRQ's Mobile Application Development Policy and App Release Procedures and determined whether they contained security requirements for vetting mobile apps. We also determined whether AHRQ followed these requirements when developing and vetting the app.

We provided detailed documentation about our preliminary findings to AHRQ in advance of issuing our draft report.

We conducted this performance audit in accordance with generally accepted government auditing standards. Those standards require that we plan and perform the audit to obtain sufficient, appropriate evidence to provide a reasonable basis for our findings and conclusions based on our audit objectives. We believe that the evidence obtained provides a reasonable basis for our findings and conclusions based on our audit objectives.

APPENDIX B: EXAMPLES OF TOOLS WE USED TO CONDUCT THE AUDIT

Frida
Frida is a dynamic code instrumentation toolkit. This tool lets testers inject snippets of JavaScript or a custom library into native apps on Windows, macOS, GNU/Linux, iOS, watchOS, tvOS, Android, FreeBSD, and QNX. Frida also provides testers with some simple tools built on top of the Frida API. These can be used as-is, tweaked to the tester's needs, or serve as examples of how to use the Application Programming Interface (API).

R2Frida
R2Frida is a plugin for radare2, a reverse engineering framework. This plugin aims to join the capabilities of static analysis of radare2 and the instrumentation provided by Frida. Even though it is possible to achieve similar results using Frida alone, radare2 helps to assemble patches in memory and static analysis.

Jadx
This package contains a Dex-to-Java decompiler. This package contains a command line and GUI tools for producing Java source code from Android Dex and APK files.

Hopper
Hopper Disassembler is a reverse engineering tool that allows testers to disassemble, decompile, and debug applications.

Objection
Objection is a runtime mobile exploration toolkit, powered by Frida. This toolkit was built with the aim of helping assess mobile applications and their security posture without the need for a jailbroken or rooted mobile device.

Burp Suite
Burp Suite is used to automate repetitive testing tasks—then dig deeper with its expert-designed manual and semi-automated security testing tools.

MitmProxy
MitmProxy is used for debugging, testing, privacy measurements, and security assessments. This tool can be used to intercept, inspect, modify, and replay web traffic such as HTTP/1, HTTP/2, WebSockets, or any other SSL/TLS-protected protocols.

Xcode
Xcode is an integrated development environment created by Apple Inc. with different functions such as source code editing, asset cataloging, version control, code documenting, app simulating, compiling, and static code analysis.

Apktool
Apktool is a tool for reverse engineering Android APK files.

APPENDIX C: HHS AND FEDERAL REQUIREMENTS

NIST SP 800-53, Revision 5, Security and Privacy Controls for Information Systems and Organizations, states:

CM-7 Least Functionality

Control:
a. Configure the system to provide only [*Assignment: organization-defined mission essential capabilities*]; and
b. Prohibit or restrict the use of the following functions, ports, protocols, software, and/or services: [*Assignment: organization-defined prohibited or restricted functions, system ports, protocols, software, and/or services*].

Discussion: Systems provide a wide variety of functions and services. Some of the functions and services routinely provided by default may not be necessary to support essential organizational missions, functions, or operations. Additionally, it is sometimes convenient to provide multiple services from a single system component, but doing so increases risk over limiting the services provided by that single component. Where feasible, organizations limit component functionality to a single function per component. Organizations consider removing unused or unnecessary software and disabling unused or unnecessary physical and logical ports and protocols to prevent unauthorized connection of components, transfer of information, and tunneling. Organizations employ network scanning tools, intrusion detection and prevention systems, and end-point protection technologies, such as firewalls and host-based intrusion detection systems, to identify and prevent the use of prohibited functions, protocols, ports, and services. Least functionality can also be achieved as part of the fundamental design and development of the system (see SA-8, SC-2, and SC-3).

HHS-OCIO-OES-2019-08-005, HHS Policy for Software Development Secure Coding Practices, states**:**

6.1. Minimum Secure Coding Practices
At a minimum, OpDivs must implement the following requirements when developing and maintaining software:

1. Follow secure coding best practice requirements, at a minimum, as recommended by United States Computer Emergency Readiness Team (US-CERT) and the Software Engineering Institute (SEI) CERT including:
   a. The SEI CERT Top 10 Secure Coding Practices;
   b. The SEI CERT language-specific coding standards;
   c. The US CERT Build Security In Guidance; and
   d. The Open Web Application Security Project (OWASP) Secure Coding Practices.

2. Ensure the code meets the level of confidence that the software is free from exploitable code vulnerabilities, regardless of whether they are already designed into the software or inserted later in its life cycle.

3. Ensure the code provides predictable execution or justifiable confidence and that the software, when executed, will provide security functionality as intended.

4. Do not trust the validity of data coming into the system (i.e., system input data) and input parameters passed for software components (e.g., input variables).

5. Verify all input data matches the specified data types and valid range of values.

6. Do not rely on unauthorized parties to store sensitive data (e.g., PII, specific security information, etc.).

7. Do not use sensitive or overly descriptive information when constructing an error message that is returned to the user.

8. Use object inheritance, encapsulation, and polymorphism wherever possible.

9. Use environment variables prudently and always check boundaries and buffers.

10. Ensure common vulnerabilities are mitigated, including but not limited to:
    a. Cross-Site Scripting;
    b. SQL Injection;
    c. HTTP Response Splitting and Header Security;
    d. Content Spoofing; and
    e. Information Leakage.

11. Ensure web-based software that utilizes mobile code technologies (e.g., Java applets, ActiveX controls, scripts, Flash animations, PDF documents and embedded scripts, Shockwave movies, Office macros, browser scripts, .NET code, HTML Application Host, scripts that execute in Windows Scripting Host, etc.) meet the following additional requirements:
    a. The software must not use high-risk mobile code technologies[9], including unsigned ActiveX controls, Shockwave Xtras, unsigned scripts that execute in Windows Scripting Host, scrap objects, and software/scripts that execute in Microsoft HTML Application Host.

[9] Use of high risk mobile code results in the user's client software needing to be configured to enable the use these technologies, thus exposing the user to attack by malware that also use these technologies. Accessed on Jun. 3, 2022.

b. All mobile code must not execute by default without privileges in a constrained execution environment (e.g., Java sandbox, .NET Common Runtime, browser sandbox).

c. All mobile code that requires privileges at runtime and software components (e.g., scripts, macros, ActiveX controls, and other executables) must be digitally signed with an HHS-approved code-signing certificate prior to being deployed and installed on a Web server. Only authorized individuals must be permitted to digitally sign the software components. The digital signature must be verified by client software (e.g., browsers, browser plugins, Microsoft Office, Windows) at runtime to validate the source and integrity of the software.

12. Do not develop software while on travel (official or unofficial) to foreign countries, especially those that are considered high-risk countries.

13. Fully test software and fix all bugs and anomalies prior to deployment.

14. Ensure developers have the appropriate credential to develop secure software.

15. Implement secure code repositories that support distributed code contribution with check-in/check-out functionality for securing HHS applications and systems.

16. Ensure that all software components are digitally signed with an HHS-approved code-signing certificate prior to being stored in the repositories to ensure the integrity of the software. Only authorized individuals must be permitted to digitally sign the software. The certificates must be validated and the digital signature must be verified prior to the code being distributed and shared.

APPENDIX D: AHRQ COMMENTS

DEPARTMENT OF HEALTH AND HUMAN SERVICES

Agency for Healthcare Research and Quality

5600 Fishers Lane
Rockville, MD 20857
www.ahrq.gov

TO: Amy J. Frontz, Deputy Inspector General for Audit Services

FROM: Robert Otto Valdez, Ph.D., M.H.S.A., Director, Agency for Healthcare Research and Quality Robert Otto Valdez, Ph.D. Digitally signed by Robert Otto Valdez, Ph.D. Date: 2024.11.07 07:12:49 -08'00'

SUBJECT: OIG Draft Report: *Some HHS Requirements for Vetting Mobile Apps Were Not Followed Prior to the Release of the AHRQ Question Builder App*, A-18-22-09008

We appreciate the time and effort put into the recent audit of our agency's mobile application, the Question Builder app. We have carefully reviewed the findings and recommendations you have provided.

The audit identified that we did not fully comply with the NIST Least Functionality Security Control and the HHS Secure Coding Practices. These findings are of great significance to us, as they underscore the need to reassess and strengthen our mobile application development and vetting policies.

We have taken these findings seriously and have already begun to implement measures to address the identified issues. We understand the importance of adhering to the highest standards of security to protect the data and privacy of our users.

The following is a detailed response to the audit findings and our proposed corrective action plans.

Attachment

FINDING # 1: AHRQ DID NOT COMPLY WITH NIST LEAST FUNCTIONALITY SECURITY CONTROL [Concur]

In response to the recent OIG audit findings, AHRQ is committed to implementing stringent measures to correct the identified shortcomings related to the NIST Least Functionality Security Control on the Question Builder app. Our corrective action plan includes the following steps:

1. Re-evaluate the current AHRQ Mobile Application Development Policy to clearly define the requirement for project officers and app developers to assess mobile apps for unnecessary or unused functionality.

2. Update the policy to ensure that such functionality is removed before submitting the app to any app store.

3. Conduct a thorough review and assessment of the Question Builder app to identify and remove any unnecessary functionality and privileges.

4. Establish a regular review process for all mobile apps to ensure compliance with the updated policy and NIST CM-7 security control.

5. Train our app developers and project officers on the updated policy and the importance of limiting app functionality to only what is necessary for the organization's mission-essential capabilities.

These corrective actions are designed to improve the security of our mobile applications and protect users' data from potential exploitation. We commit to completing these actions within 8 weeks of the final publication of this recommendation.

FINDING #2: AHRQ DID NOT FOLLOW REQUIRED HHS SECURITY POLICY FOR VETTING THE QUESTION BUILDER APP [Concur]

In response to the recent OIG audit findings, AHRQ has taken immediate action to address the concerns raised regarding the adherence to HHS Secure Coding Practices in the development and deployment of our mobile applications.

As a corrective measure, we have comprehensively updated the AHRQ Mobile Application Development Policy to incorporate explicit reference to the HHS Secure Coding Practices. The updated policy now mandates thorough vetting of the security of all AHRQ mobile applications to ensure compliance with the HHS secure coding policy requirements.

In addition to this, we have established stringent procedures to identify and rectify any potential security vulnerabilities before releasing a mobile app to app stores for public use. This includes a thorough security check to ensure software is free from exploitable code vulnerabilities and will provide security functionality as intended.

The updated policy and the new procedure underscore our commitment to ensuring the highest level of security for our users and their data. These measures will significantly reduce the potential for any exploitation of vulnerabilities and strengthen the overall security of our mobile applications.

We appreciate the OIG's efforts in identifying these issues and remain committed to adhering to the highest standards of security for our mobile applications. Please let us know if you have any further comments or questions regarding these recommendations and corresponding actions.

Report Fraud, Waste, and Abuse

OIG Hotline Operations accepts tips and complaints from all sources about potential fraud, waste, abuse, and mismanagement in HHS programs. Hotline tips are incredibly valuable, and we appreciate your efforts to help us stamp out fraud, waste, and abuse.

TIPS.HHS.GOV

Phone: 1-800-447-8477

TTY: 1-800-377-4950

Who Can Report?

Anyone who suspects fraud, waste, and abuse should report their concerns to the OIG Hotline. OIG addresses complaints about misconduct and mismanagement in HHS programs, fraudulent claims submitted to Federal health care programs such as Medicare, abuse or neglect in nursing homes, and many more. Learn more about complaints OIG investigates.

How Does it Help?

Every complaint helps OIG carry out its mission of overseeing HHS programs and protecting the individuals they serve. By reporting your concerns to the OIG Hotline, you help us safeguard taxpayer dollars and ensure the success of our oversight efforts.

Who Is Protected?

Anyone may request confidentiality. The Privacy Act, the Inspector General Act of 1978, and other applicable laws protect complainants. The Inspector General Act states that the Inspector General shall not disclose the identity of an HHS employee who reports an allegation or provides information without the employee's consent, unless the Inspector General determines that disclosure is unavoidable during the investigation. By law, Federal employees may not take or threaten to take a personnel action because of whistleblowing or the exercise of a lawful appeal, complaint, or grievance right. Non-HHS employees who report allegations may also specifically request confidentiality.

Stay In Touch

Follow HHS-OIG for up to date news and publications.

 OIGatHHS

 HHS Office of Inspector General

Subscribe To Our Newsletter

OIG.HHS.GOV

Contact Us

For specific contact information, please visit us online.

U.S. Department of Health and Human Services
Office of Inspector General
Public Affairs
330 Independence Ave., SW
Washington, DC 20201

Email: Public.Affairs@oig.hhs.gov