DRAFT NISTIR 8235

Security Guidance for First Responder Mobile and Wearable Devices

Gema Howell
Kevin G. Brady, Jr.
Don Harriss
Scott Ledgerwood

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8235-draft

National Institute of Standards and Technology
U.S. Department of Commerce

DRAFT NISTIR 8235

Security Guidance for First Responder Mobile and Wearable Devices

Gema Howell
Kevin G. Brady, Jr.
Applied Cybersecurity Division
Information Technology Laboratory

Donald Harriss
Scott Ledgerwood
Public Safety Communications Research Division
Communications Technology Laboratory

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8235-draft

September 2020

U.S. Department of Commerce
Wilbur L. Ross, Jr., Secretary

National Institute of Standards and Technology
Walter Copan, NIST Director and Under Secretary of Commerce for Standards and Technology

National Institute of Standards and Technology Interagency or Internal Report 8235
90 pages (September 2020)

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8235-draft

Public comment period: *September 28, 2020* through *November 30, 2020*

National Institute of Standards and Technology
Attn: Applied Cybersecurity Division, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930
Email: NISTIR8235_Comments@nist.gov

All comments are subject to release under the Freedom of Information Act (FOIA).

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of other than national security-related information in federal information systems.

Abstract

Public safety officials utilizing the forthcoming public safety broadband networks will have access to devices, such as smartphones, tablets and wearables. These devices offer new ways for first responders to complete their missions but may also introduce new security vulnerabilities to their work environment. To investigate this impact, the security objectives identified in NIST Interagency Report (NISTIR) 8196, *Security Analysis of First Responder Mobile and Wearable Devices,* were used to scope the analysis of public safety mobile and wearable devices and the current capabilities that meet those security objectives. The ultimate goal of this effort is to provide guidance that enables jurisdictions to select and purchase secure devices and assist industry to design and build secure devices tailored to the needs of first responders.

Keywords

cybersecurity; first responders; internet of things; IoT; mobile security; public safety; wearables.

Acknowledgments

First and foremost, the authors wish to gratefully acknowledge the contributions of the public safety professionals offering their time and rich expertise to our previous study which assisted in the production of NISTIR 8196 *Security Analysis of First Responder Mobile and Wearable Devices*. Additionally, information gleaned from the Association of Public-Safety Communications Officials (APCO), specifically Mark Reddish, was invaluable. The authors also would like to thank their colleagues who reviewed drafts of this document and contributed to its technical content including John Beltz, Michael Ogata, Andrew Regenscheid, and Nelson Hastings of NIST; Vincent Sritapan of DHS S&T.

Audience

This document is intended for those acquiring mobile devices and wearables for deployment in public safety scenarios. This document may also be useful for those designing public safety smartphones, tablets, and wearable devices.

Call for Patent Claims

This public review includes a call for information on essential patent claims (claims whose use would be required for compliance with the guidance or requirements in this Information Technology Laboratory (ITL) draft publication). Such guidance and/or requirements may be directly stated in this ITL Publication or by reference to another publication. This call also includes disclosure, where known, of the existence of pending U.S. or foreign patent applications relating to this ITL draft publication and of any relevant unexpired U.S. or foreign patents.

ITL may require from the patent holder, or a party authorized to make assurances on its behalf, in written or electronic form, either:

a) assurance in the form of a general disclaimer to the effect that such party does not hold and does not currently intend holding any essential patent claim(s); or

b) assurance that a license to such essential patent claim(s) will be made available to applicants desiring to utilize the license for the purpose of complying with the guidance or requirements in this ITL draft publication either:

   i. under reasonable terms and conditions that are demonstrably free of any unfair discrimination; or
   ii. without compensation and under reasonable terms and conditions that are demonstrably free of any unfair discrimination.

Such assurance shall indicate that the patent holder (or third party authorized to make assurances on its behalf) will include in any documents transferring ownership of patents subject to the assurance, provisions sufficient to ensure that the comMitMents in the assurance are binding on the transferee, and that the transferee will similarly include appropriate provisions in the event of future transfers with the goal of binding each successor-in-interest.

The assurance shall also indicate that it is intended to be binding on successors-in-interest regardless of whether such provisions are included in the relevant transfer documents.

Such statements should be addressed to: NISTIR8235_Comments@nist.gov

Table of Contents

List of Appendices

List of Tables

List of Figures

1 Introduction

Public safety first responders are the first at the scene of an emergency incident. Their day-to-day includes life-saving and sometimes life-threatening activities. As commercial and enterprise technology advance, first responders have the opportunity to take advantage of this technology to enhance their efficiency, safety, and capabilities during an incident. The nationwide public safety broadband network (NPSBN), is steadily deployed across the United States and operated by AT&T under the guidance of the First Responders FirstNet Authority (FirstNet)., per the Middle Class Tax Relief and Job Creation Act of 2012 [1]. Networks like those provided by FirstNet by AT&T and the NPSBN will allow first responders to use modern communication technology (smartphones/mobile devices) as well as other smart devices (smart wearables) to accomplish their public safety mission.

As with any new technology, there are security concerns, such as the vulnerabilities and threats to their users. In the case of public safety there are concerns that exploits of vulnerabilities may inhibit first responders from performing their duties and put their safety at risk. NISTIR 8196 *Security Analysis of First Responder Mobile and Wearable Devices*, is a document that was produced in a previous study to understand the specific security needs of smart devices for first responders [2]. The document captures the various use cases of public safety mobile and wearable devices, the known attacks on public safety mobile and wearable devices, and information received from interviews with actual public safety officials. Due to their unique roles, environments, and situations, the information in NISTIR 8196 is important to grasp the first responder perspective and analyze the security objectives necessary for all first responder devices.

Mass production of mobile and wearable smart devices makes it easy to find and buy any device that may meet one's wants and needs. Technology is primarily produced for the general consumer or enterprise and not specifically designed with public safety in mind. This could lead to potential repercussions if the appropriate device is procured without consideration of the security and safety of first responders. When it comes to selecting mobile and wearable devices, there is little security guidance that focuses on the particular needs of public safety. During an emergency, a first responder should have some assurance that their devices are reliable and secure.

1.1 Purpose

The purpose of this document is to share a high-level overview of the current capabilities of public safety mobile and wearable devices. This will give insight of the security capabilities available within today's devices. Additionally, this document provides guidance for procuring and designing secure mobile and wearable devices specifically for public safety. This document includes the following contributions:

- A list of tests developed to analyze public safety mobile and wearable devices
    - Each test provides an overview of the outcome and the analysis derived from observation of that outcome
- A collection of best practices and guidance for public safety mobile and wearable devices

1.2 Scope

This research effort focuses primarily on public safety mobile and wearable devices. Securing broadband networks, for instance, the management, and operation of cellular networks are out of scope. An entire class of devices exists under the IoT umbrella; however, this document solely focuses on wearable IoT devices that may be used by public safety. Additionally, mobile applications that ship with a public safety smartphone are considered in scope, as they are often required to perform typical public safety activities, such as voice communication. Backend services and the communication paths utilized by these mobile applications, to include data transmission from an application to supporting infrastructure, are in scope. Finally, public safety officials work in a variety of disciplines, this Interagency Report (IR) is focused on first responders (i.e., fire service, EMS, and law enforcement) and the public safety device administrators that provide devices to first responders. Testing scenarios, gaps, analysis and guidance beyond the scope of this document or the needs of first response, may consult supplementary resources such as the NIST Cybersecurity Framework, the NIST Mobile Security Framework, the Open Web Application Security Project (OWASP), and other device specific security hardening resources.

1.3 Document Structure

The document is organized into the following major sections:

- Section 2 provides an overview of the technology analyzed,
- Section 3 outlines the methodology used for analysis
- Section 4 summarizes the test plan and findings
- Section 5 suggests best practices and guidance for public safety mobile and wearable devices
- Section 6 concludes the document with a review of the document, future considerations, and other related NIST work
- Section 7 contains a list of references used in the development of this document

The document also contains appendices with supporting material:

- Appendix A defines selected acronyms and abbreviations used in this publication, and
- Appendix B provides a detailed description of each test, including, procedures, analysis, gaps, and guidance

2 Technology Overview

The following section describes the technologies reviewed throughout this effort. When selecting the public safety devices to analyze, PSCR Engineers searched for public safety-grade technology and devices that could be used in the future to assist first responders. Below is an overview of the types of the devices and why those devices are relevant to this project.

2.1 Public Safety Mobile Devices

The selection of public safety mobile devices was based on knowledge of the upcoming public safety communication systems. The Federal Communications Commission has allocated a portion of the 700 MHz band as the public safety spectrum. This portion of the spectrum is also known as the Band 14 spectrum, which is to be utilized as the national public safety broadband network. This spectrum will allow for device communications to penetrate walls and buildings and prevent congestion issues due to flooded transmissions during an emergency. PSCR Engineers sought out mobile devices that utilized band 14, as well as other mobile devices that are not band 14 capable but may be ruggedized or have a more secure operating system.

The analyzed public safety mobile devices use a fully-fledge mobile operating system. Typically, the mobile devices used an android operating system. The version of the operating system varied per device, some being 4-5 versions behind the latest release.

2.2 Public Safety Wearable Devices

Wearable devices made specifically for public safety are slowly being introduced to the marketplace. Outside of public safety specific wearable devices, PSCR Engineers also acquired wearable devices that may assist first responders in different ways, such as,awareness, communication, and data sharing. Examples of wearable devices include the following:

- Bluetooth headset
- Body camera
- Smart glasses
- Vital-sign monitors/Body sensors

Most of the wearable devices analyzed, use some variation of Bluetooth and/or Wi-Fi as their wireless communication protocol. These protocols allow for communication between a wearable device and a mobile device or desktop. Wearable devices typically do not have a complex operating system and perform minimal tasks that enable them to process and send information to be interpreted by an application on another system such as a mobile device or desktop computer. Many of the wearable devices analyzed through this research, are dependent on being able to send information to a mobile application to be interpreted, stored, and possibly shared through cloud services.

3 Analysis Methodology

This section gives an overview of the methodology used to develop the best practices and guidance for securing First Responder mobile and wearable devices. The process required thorough understanding of the security objectives from the perspective of first responders. This was accomplished through interviews with public safety officials and development of NISTIR 8196, *Security Analysis of First Responder Mobile and Wearable Devices* [2].

With the information gathered from NISTIR 8196, PSCR Engineers were able to take the steps necessary to analyze the security of current mobile and wearable devices and compare their analysis with the security objectives of first responders. This exercise resulted in this document and ultimately security guidance that describes the security capabilities that should be included in mobile and wearable devices for first responders.

3.1 Test Plan

The previous effort, NISTIR 8196, identified eight (8) security objectives, documented below:

Table 1 - Handset and Wearable Security Objectives

Availability	Confidentiality
Ease of Management	Authentication
Interoperability	Integrity
Isolation	Healthy Ecosystem

Using these security objectives, the first step was to develop a test plan to perform a security analysis of public safety mobile and wearables devices. The security objectives, which focus on the security needs of public safety, are used to define the scope of the tests. Some, not all, security objectives have sub-objectives. A list of these sub-objectives can be found below:

Table 2 - Handset and Wearable Security Sub-objectives

SECURITY OBJECTIVE	SUB-OBJECTIVE(S)
AVAILABILITY	Network Availability Network Agility Data Availability Device Availability
EASE OF MANAGEMENT	N/A
INTEROPERABILITY	Device Configuration Infrastructure Interoperability

	Network Interoperability Security Technology Interoperability Data Format Interoperability
ISOLATION	Data Isolation Application Isolation
CONFIDENTIALITY	Data In Transit Data At Rest
AUTHENTICATION	Ease of Authentication User to Device Authentication Device to Network Authentication User to Third Party Service/Mobile Device/ Wearables
INTEGRITY	N/A
HEALTHY ECOSYSTEM	Configuration Updates Bundled Applications

Many of the sub-objectives are not in scope for this analysis, as these sub-objectives require a more in-depth analysis and test plan than intended for the purposes of this project. The excluded security objectives are important to the needs of public safety and may be analyzed in future research.

3.2 Testing & Analysis

PSCR Engineers gathered a series of mobile and wearable devices that are advertised for public safety use or could be used to assist first responders. Using the test plan, PSCR Engineers applied the tests to the acquired devices. With the observed results, an analysis was performed that gave understanding of the current security posture of these devices. Using information gathered from the initial research in NISTIR 8196 and the results from this security analysis, a gap analysis was performed to identify any missing features or capabilities within the public safety mobile and wearable devices. The results of all research allowed for the next step in the overall methodology, the development of best practices and guidance for acquiring secure mobile and wearable devices for public safety.

3.3 Develop Guidance

After completion of the security testing and gap analysis, for the final step in the methodology PSCR Engineers developed best practices and guidance. To develop this guidance, PSCR Engineers used information gathered from the test analysis and referenced current security best practices for general information systems that can apply to mobile and wearable devices. These references include the Cybersecurity Framework Version 1.1 [3], NISTIR 8228, *Considerations* *for Managing Internet of Things (IoT) Cybersecurity and Privacy Risks* [4]*,* and DRAFT (2nd) NISTIR 8259, *Recommendations for IoT Device Manufacturers: Foundational Activities and Core Device Cybersecurity Capability Baseline* [5].

4 Test Overview

The type of testing performed for this analysis demonstrates an understanding of the state of firmware/software that is pre-installed, the vulnerabilities present on the device, and the types of secure technologies included within the devices. This effort will also assist with understanding what type of external certifications and testing occurs for these devices, such as Ingress Protection (IP) ratings.

This document does not identify specific devices, manufacturers, or service providers. NIST does not condone, endorse, dissuade or dismiss the use of any specific device, manufacturer, service provider or analysis tool utilized for information collection. All test information was gathered at a specific date and time before the writing of this document and may not accurately reflect the current state, condition or availability of information pertaining to a specific device. In this section information will be collated to reflect a summary of information regarding all devices tested.

The following sections provide a summary of the test findings for mobile and wearable devices. Each section starts with a table that provides an overview of the tests used to analyze the security capabilities of mobile and wearable devices. The table includes the following:

- *Test Number* – The number associated with each test
- *Test Name* –The test name, which summarizes the purpose of the test
- *Security Objective(s)* – The mapping to one or more of the security objectives from NISTIR 8196
- *Test Description* – The test description describes the information the test will provide in relation to the security analysis of the mobile and wearable devices

For more information about the test outcomes, including a detailed analysis of potential impacts , future considerations for public safety, and any gaps found as a result of the test, see Appendix B.

4.1 Mobile Test Results Summary

Table 3 - Mobile Device Tests

Test No.	Test Name	Security Objectives	Test Description
1	Obtain General Hardware Information	Ease of Management Data Availability Healthy Ecosystem	This test identifies information about the device, and how easy it is to do so.
2	Obtain General Software Information	Ease of Management Network Agility	This test identifies the name and software version of operating system and major applications that are shipped with the device. This will also attempt to understand the protocol versions for the primary

		Healthy Ecosystem	wireless protocols (i.e., Wi-Fi, Bluetooth, and Cellular).
3	Device Ruggedization Ratings	Device Availability Ease of Management Healthy Ecosystem	Implementation of ruggedization ensures durability for First Responder applications and survivability of day-to-day use. This test identifies the Ingress Protection (IP) ratings and any ruggedization information available for the device. Physical survivability of First Responder mobile devices ensures the integrity of responder data. IP ratings and certification ensure data integrity by reducing occurrence of device failure in extreme environments as well as reliable communications.
4	Obtaining Vulnerability Information from OS version and known databases	Device Availability Data Availability Integrity Healthy Ecosystem	In this test, PSCR Engineers manually check the software versions of the OS that shipped within the device against a list of vulnerabilities within public databases to understand the types of vulnerabilities already known within the OS. PSCR Engineers look to understand the impact and criticality of all the known vulnerabilities.
5	Vulnerability Scan via Mobile Threat Defense (MTD) Application	Device Availability Data Availability Integrity Healthy Ecosystem	This test uses publicly available mobile threat defense (MTD) applications to identify vulnerabilities within the mobile OS and applications shipped with the device. PSCR Engineers look to understand the impact and criticality of all the known vulnerabilities
6	External Fingerprinting	Confidentiality Integrity	Fingerprinting a device is often an initial stage of information gathering before it is attacked. Device integrity can be verified by performing external scanning and fingerprinting over a network connection. This test uses a set of common network scanning tools to understand the types of ports and protocols open and running on the device.
7	External Vulnerability Scan	Data Availability Confidentiality Integrity Healthy Ecosystem	This test uses a set of common vulnerability scanners to understand the types of vulnerabilities within the device. An external vulnerability scan device is often part of an information gathering phase before it is attacked. PSCR Engineers look to understand the impact and criticality of all the known vulnerabilities
8	MAC Address Randomization	Confidentiality	Device confidentiality and autonomy can be maintained through the use of MAC address randomization. This test identifies if the device is utilizing MAC addresses randomization. This includes the Bluetooth MAC addresses.

9	Device Update Policy	Healthy Ecosystem	This test seeks to understand how often the device is scheduled to receive security updates and other software from the vendor. Specifically, the regularity / cadence, type, and reasons for updating the device and applying security patches will be reviewed.
10	Rogue Base station Detection	Availability Confidentiality Integrity	This test identifies if the public safety mobile devices can detect rogue base stations affecting their cellular traffic in malicious ways.
11	Configuration Guidance	Integrity Interoperability Healthy Ecosystem	This test reviews the type of guidance provided from the vendor to the public safety professionals, and if any of this is security guidance dedicated to properly owning, operating, and configuring the device for public safety use.
12	Wi-Fi Man-in-the-Middle (MitM) Detection	Availability Confidentiality Integrity	This test checks to see if the mobile device is able to locally detect man-in-the-middle attacks when using Wi-Fi.
13	Boot Integrity	Integrity	This test checks to see if the mobile device is performing some form of boot validation. Boot validation is an integrity check on device boot files and processes to verify that the mobile OS has successfully executed into a valid state. If validation succeeds, the device will continue to load the system and may perform additional validation. If validation fails, the device will stop the boot sequence, enter an error state and/or reboot.
14	Data Isolation	Integrity Isolation	In this test, PSCR Engineers seek to understand if the mobile device is utilizing an isolation technology such as SELinux.
15	Device Encryption	Confidentiality Ease of Management	In this test, PSCR Engineers seek to understand if the device is locally utilizing device-wide encryption, and how difficult it is to use.

PSCR Engineers found that most smart mobile devices have the built-in capabilities and the information necessary to meet the various security objectives of First Responders. Smart Mobile devices have been around for more than 10 years, which has allowed growth in many areas (e.g., functionality and security). With a full OS and screen display, users/administrators can easily find device information within the *Settings* menu (i.e., hardware and software information). Additional information (i.e., configuration guidance and update policies) is easily accessible in the user manuals available online. All of this information is useful for device administrators to use when making risk decisions and deciding whether to use a specific mobile device that meets the identified First Responder requirements.

Security is not automatically enabled in mobile devices. Although mobile devices have built-in security features, enabling those features requires additional APIs. For example, PSCR Engineers leveraged a free 3rd party mobile application called a Mobile Threat Defense tool to analyze any potential or current vulnerabilities on the mobile device under analysis. A Mobile Threat Defense tool can detect the presence of malicious apps or operating system (OS) software, known vulnerabilities in software or configurations, and connections to blocklisted websites/servers or networks [6]. There are other applications/tools that can enable different security features within a mobile device, such as a VPN connection or enforce policies/device configurations.

PSCR Engineers found that a few mobile devices were operating on an outdated OS. Using an outdated OS allowed the device to continue to use Public Safety mobile applications that are only supported by the old OS. OS updates are developed to improve features or patch bugs/vulnerabilities. Using an outdated OS may allow a First Responder to use the Public Safety application they need for their daily activities, but may also leave the phone in a vulnerable state because it has not received the necessary patches.

Lastly, PSCR Engineers found that mobile devices are not able to detect a rogue/fake base station and prevent connection to these base stations. Rogue base stations are not owned or operated by a Mobile Network Operator (MNO), they broadcast cellular network information, and masquerade as a legitimate network [7]. These base stations can be used for MitM attacks to eavesdrop, perform a denial of service, or gather information to track a user's location. A common attack is using a rogue base station as an International Mobile Subscriber Identity (IMSI) catcher. When a mobile device attempts to connect to a rogue base station, they are able to gather that device's IMSI information. With a device's IMSI information, an attacker can track a device as it moves from base station to base station. Recent updates to the 3GPP cellular standards conceal the subscriber identity so that rogue base stations are unable to track the location a user's device [8]. Although this may defeat IMSI catchers, this does not resolve the other potential attacks because mobile devices are constantly trying to connect to a cellular network and may connect to a rogue base station if it has the strongest signal. There are ongoing standards activities and research projects to improve mobile device technology and protect devices against rogue base station attacks.

4.2 Wearable Test Results Summary

Table 4 - Wearable Device Tests

Test No.	Test Name	Security Objectives	Test Description
1	Obtain General Hardware Information	Ease of Management Data Availability Healthy Ecosystem	This test identifies information about the device, and how easy it is to obtain that information.
2	Obtain General Software	Ease of Management	This test identifies the name and software version of operating system and major applications that are shipped with the device. Note that this is much more

	Information	Network Agility Healthy Ecosystem	difficult on a wearable device than on a mobile device, and NIST engineers will not be performing firmware and binary extraction activities. This will also attempt to understand the protocol versions for the primary wireless protocols (i.e., Wi-Fi, Bluetooth, and Cellular). This test will also investigate the use of wearable specific protocols such as Near field communications (NFC), ZigBee, and Z-Wave.
3	Device Ruggedization Ratings	Device Availability Ease of Management Healthy Ecosystem	Implementation of ruggedization ensures durability for First Responder applications and survivability of day-to-day use. This test identifies the Ingress Protection (IP) ratings and any ruggedization information available for the device.
4	Obtaining Vulnerability Information from OS version and known databases	Device Availability Data Availability Integrity Healthy Ecosystem	In this test, PSCR Engineers manually check the software versions of the OS that shipped within the device against a list of vulnerabilities within public databases to understand the types of vulnerabilities already known within the OS. PSCR Engineers look to understand the impact and criticality of all the known vulnerabilities.
5	Device Pairing	Authentication Integrity	This test identifies how the wearable device pairs and authenticates to a mobile device, such as the use of an insecure pairing mechanism. Investigate any encryption, privacy protections, device names, and insecure pairing types.
6	Device Encryption	Confidentiality	This test identifies how the wearable device communicates with a mobile device, specifically using encryption. This will include the use of secure algorithm, reasonable key sizes, and any man in the middle protection.
7	Configuration Guidance	Integrity Interoperability Healthy Ecosystem	This test reviews the type of guidance provided from the vendor to the public safety professionals, and if any of this is security guidance dedicated to properly owning, operating, and configuring the device for public safety use.
8	MAC Address Randomization	Confidentiality	Device confidentiality and autonomy can be maintained through the use of MAC address randomization. This test identifies if the device is utilizing MAC addresses randomization. This includes the Bluetooth MAC addresses.
9	Device Update Policy	Healthy Ecosystem	This test seeks to understand how often the device is scheduled to receive security updates and other software from the vendor. Specifically, the regularity / cadence, type, and reasons for updating the device and applying security patches will be reviewed.

Through testing and analysis, PSCR Engineers found that most wearable devices have minimal functionality. The limited functionality seems to be partially intentional because the device requires limited processing power which minimizes batter power usage and allows for longer battery life. This also restricts the general capabilities of the device, including the security capabilities. Wearable devices often do not have a screen display and require another application (e.g., mobile application) to interface with the device and gather information about the device. Alternatively, detailed device information can be found in the user manual or on the device manufacturer's website.

When reviewing access to wearable device information, PSCR Engineers found limited and varying information available on each device. Some information required network traffic analysis to identify information such as, the version of the network protocol being used, or the security levels being implemented by the wearable device. Most devices did not provide an update policy or secure configuration guidance.

Network protocols varied amongst the wearable devices, with few using Wi-Fi or Cellular protocols. The most common network protocol used across the wearable DUTs, was Bluetooth. Many of the devices were using older versions of the Bluetooth specification or were able to downgrade to an older spec for device compatibility reasons. PSCR Engineers analyzed the authentication and encryption capabilities with regards to the Bluetooth device pairing process.

For authentication, most wearable DUTs use Simple Pairing Mode to request device access, which does not provide MitM protection. This potentially leaves wearable devices vulnerable to eavesdropping, a denial of service, and location tracking. Devices that utilize version Bluetooth 4.0 or greater have the ability to use Bluetooth Smart or Bluetooth Smart Ready, which can provide MitM protection if user input is available. Most wearables do not have a way to input the PIN code required for MitM protection. PSCR Engineers found that one device used MitM protection, but the PIN was static and could easily be brute forced or found in the device manual. Overall most devices used the older Bluetooth pairing method (Simple Pairing Mode) and auto accepts any connection requests. More information can be found in section B.2.5.

The encryption used by the wearable DUTs followed that of devices using older versions of Bluetooth (e.g., Bluetooth version 2.1) and secure simple pairing with security level 2, which uses unauthenticated keys. Some older versions of Bluetooth use encryption algorithms that are no longer approved by the Federal Information Processing Standards (FIPS). Bluetooth versions 4.1 or greater and Bluetooth Low Energy all use FIPS approved algorithms [9].

Ultimately, PSCR Engineers concluded that wearables are currently able to adhere to a minimum number of Public Safety security objectives. Wearable devices are built to emphasize usability rather than security. In a field such as Public Safety, usability is vital for a First Responder to perform their life-saving activities, but without the proper hardening this could impact the usability of a wearable device (e.g., Denial-of-Service or transmission of inaccurate data) [18]. Wearable devices may require future improvements to better meet the security needs of First Responders.

5 Best Practices and Guidance

After reviewing the test analysis results, PSCR Engineers gained an understanding of the current state of mobile and wearable devices with regards to their security capabilities. These results were then compared to the First Responder security objectives from NISTIR 8196. This comparison was done to understand gaps in the current capabilities of these devices vs. what first responders are looking for when it comes to the security of their devices.

In this section, PSCR Engineers provide guidance to assist first responders when acquiring mobile and wearable devices that meet their security needs. This guidance is intended to be beneficial and understandable for all stakeholders within the public safety mobile and wearable device arena. First responders can benefit from this guidance because they are the primary users of these devices and a secure device allows them to focus on their life-saving activities. Also, first responders should have a way to communicate their needs with regards to a secure device. Public safety device administrators are responsible for distribution and configuration of mobile and wearable devices. This guidance will help administrators ensure they are aware of what security features to ask for, how to apply the security features, and train their users for proper use. Finally, this guidance will give device manufacturers insight into the security features and capabilities that first responders are looking for within their mobile and wearable devices. With this information, manufactures can build to meet the security objectives of first responders.

PSCR Engineers used the Cybersecurity Framework version 1.1, to aid in the guidance communication. The Cybersecurity Framework is a tool that can be used to communicate cybersecurity information to various technical levels within an organization. The Cybersecurity Framework defines five functions (Identify, Protect, Detect, Respond, and Recover) that are easy to understand and can be used to communicate in plain language to various members within an organization [3]. PSCR Engineers used these functions to provide high-level guidance to take into consideration when aspiring to acquire secure mobile and wearable devices.

5.1 Guidance for Mobile and Wearable Devices

Mobile devices have many built-in security capabilities. This is partially due to their size, storage capability, and fully-fledged operating systems. Somewhat mimicking traditional desktops, a mobile phone has various network capabilities (e.g., Bluetooth, Wi-Fi, and cellular connectivity), along with the ability to update firmware and download software to expand the devices abilities even further. Many mobile devices are capable or have the information necessary to meet the security objectives of first responders.

Wearable devices are very different from mobile devices, in that they are typically built primarily to accomplish a specific use (e.g., communication through a headset or to record vital signs). Due to their often-limited processing power, wearable devices do not have various options when it comes to functionality and security. Device information and capabilities vary per wearable device, and the inconsistency with wearable device information makes it difficult for interested parties to find what they need to make risk-decisions. While there is a variance in capabilities, this could be beneficial if the capabilities meet the needs of first responders using them (i.e., functionally and security-wise). The configuration of wearable device capabilities is not as flexible as with mobile devices. Often wearable devices only come with preset abilities and are not updatable. For some wearable devices that interfaced with a mobile application or other external software application, some areas of functionality/firmware could be updated. There are several areas where wearable devices can better address the security objectives of first responders, and they are highlighted in the guidance provided below.

Below is a chart that includes the following:

- Cybersecurity Framework Function – the Cybersecurity Framework function that provides the plain language term that applies to the guidance
- Guidance – the one-line notion that states guidance of what to consider when it comes to the security of first responder mobile and wearable devices

Table 5 – High-Level Guidance for Securing Mobile and Wearable Devices

Cybersecurity Framework Function	Guidance
Identify	**Identify** your public safety needs and devices
Protect	**Protect** yourself by applying security and training users
Detect	**Detect** issues by logging and monitoring your devices
Respond	**Respond** with a prepared plan
Recover	**Recover** by implementing the plan and constantly improving

The following subsections give more information about what should be considered when applying each aspect of the guidance mentioned in the chart above. These subsections also map the guidance to the First Responder security objective(s) that are addressed through the guidance. Lastly, the guidance is mapped to any tests that are relevant to the guidance being discussed.

5.1.1 Identify – your public safety needs and devices

The first step in making decisions about technology acquisition is understanding an organization's needs. An organization needs may be influenced by the following:

- use cases
- threat modeling/risk assessments
- business policies
- desired security objectives

An example of these influential components can be found in NISTIR 8196 [1]. This information can be used to guide the search for features and capabilities within a device. Here are some example features and capabilities that may be considered necessary for First Responder devices:

- Make & model of the device
- Firmware and software information
- Network protocols (e.g., Wi-Fi, Bluetooth, Cellular)
- Ruggedization ratings (e.g., IP ratings or MIL-STD)
- Security capabilities (e.g., authentication options and encryption)
- Update policies and schedules

Once the organization establishes their device needs, this can be used to identify devices that meet these needs. To identify these devices, device administrators will need to obtain information about their prospective or current devices. A device administrator can use this information to decide whether a device has most of their required features, which may be prioritized by usability and security capabilities [18].

PSCR Engineers found that mobile devices provide most of the information necessary to allow public safety device administrators to make decisions around whether a device has the security features that meets their needs. Wearable devices differed in that the device information provided varied per device. Many wearable devices require additional research or a discussion with the device vendor to find specific details about the device's specifications. Some wearable device information that was not readily available include the security capabilities and limitations (e.g., encryption, MitM protection, degradability) within a specific version of Bluetooth.

This guidance will assist public safety device administrators to identify devices that meet their specific public safety needs. Device information gives insights into device capabilities, including their interoperability with other devices/systems. Also, having information readily available about a device will help device administrators maintain and manage the devices that are used by first responders.

Security Objectives: Availability, Ease of Management, Interoperability, Healthy Ecosystem

Test References in Appendix B: B.1.1, B.1.2, B.1.3, B.1.9, B.1.11, B.1.13, B.2.1, B.2.2, B.2.3, B.2.9

5.1.2 Protect – yourself by applying security and training users

Once devices are acquired security must be applied. The security applied should go along with the public safety security needs identified through the prior guidance given in section 5.1.1. Some devices are built with security features automatically enabled. Most devices require secure configuration to allow an organization to configure to their specific needs (e.g., authentication and encryption requirements). When applying security, public safety device administrators should consider both usability and security [18]. Usability and security are both very important to public safety officials. A device needs to be usable to accomplish the necessary tasks during an emergency incident. Security is important because if not applied, it could leave a device vulnerable to attacks, which could then compromise the usability of the device during an emergency incident.

In addition to applying security, public safety device users should receive training to properly use their devices. User error can impact security if users do not do their part to secure their device. Most security configurations should be applied prior to providing a user with a device, but some security controls require user interaction. For example, a public safety user may be required to create a password or use an authenticator for their device. The user should understand the importance of applying the password and the potential risk to sharing their password or authenticators.

With few exceptions, mobile devices do not apply security by default. Some security features can be enabled manually by a public safety device administrator. Other features require additional third-party services to apply security features such as policy configurations, encrypt data transmissions, or analyze mobile applications. The practice guide, NIST SP 1800-21 *Mobile Device Security: Corporately-owned Personally-enabled,* discusses some of the various mobile device security solutions that can be used to apply security configurations and policies to a mobile device [10]. These solutions include an Enterprise Mobility Management (EMM) solution, Mobile Application Vetting (MAV), and Virtual Private Network (VPN).

PSCR Engineers developing applications for wearables may require an API on a mobile device or other system to update and apply certain features. Most security features were unchangeable, which is why it is very important to be aware of the security features within a wearable device; to ensure the device meets the desired public safety security objectives. If future wearable devices are more configurable with their security capabilities, this would allow a single device to be configured to meet the security needs of various different parties.

With the appropriate security applied to First Responder devices, this assists with mitigating against potential threats that could harm the security and usability of a device. Any risk to security of a device could put the safety of a first responder at risk. By applying security and training users in advance, first responders can focus on an emergency incident without the unnecessary distraction of interacting with a device.

Security Objectives: Availability, Isolation, Confidentiality, Authentication, Integrity

Test References: B.1.4, B.1.5, B.1.7, B.1.11, B.1.12, B.1.14, B.1.15, B.2.4, B.2.5, B.2.6, B.2.7

5.1.3 Detect – issues by logging and monitoring your devices

First Responder mobile and wearable devices should be constantly monitored to check for compliance, vulnerabilities, and any other issues. While monitoring, it is also important to log monitoring and general device activities. Compliance monitoring will check for any authorized changes to the device configuration such as, changing the password settings or downloading an unauthorized application to the device. Vulnerability monitoring can check for different types of vulnerabilities that may impact the device (e.g., application vulnerabilities, network vulnerabilities, or OS vulnerabilities). Potential issues related to device health are also important to monitor since they can also have significant consequences for the security and usability of devices (e.g., battery health and overheating).

Using device information (i.e., make/model, OS, network protocol), public safety device administrators can manually monitor devices by performing a web search for potential vulnerabilities. Mobile device security solutions (e.g., EMM and MTD) can monitor mobile devices and send notifications to the administrator and/or the user when it finds a potential vulnerability or policy violation. Some solutions can also perform compliance actions if it finds that a mobile device is violating an enforced policy. An example policy violation is a user removing a required authentication method. To address this policy violation, a compliance action could be enforced to restrict the device's access to an organization's resources, until the device is no longer in violation of the policy. Wearable devices do not have easily available monitoring tools and may require manual monitoring through research and analysis. Some devices may provide their own monitoring tools, but this is not consistent across all wearable devices.

By logging and monitoring devices, device administrators are aware of device issues and trends in device activity. This is the information needed to make decisions about how to address issues in the short-term and long-term. With insight into current or potential issues with a device, a device administrator can make risk-based decisions (e.g., likelihood, impact, etc.) for how to address any device concerns. Notification of any anomalous activity allows administrators to address device issues promptly. Lastly, continuous monitoring and logging information provides the ability to monitor cybersecurity incidents and review the effectiveness of the protective measures in place.

Security Objectives: Availability, Integrity, Ease of Management, Healthy Ecosystem

Test References: B.1.1, B.1.2, B.1.4, B.1.5, B.1.7, B.2.1, B.2.2, B.2.4

5.1.4 Respond – with a prepared plan to address issues

When device issues are found, it is helpful to be prepared with a plan of action to address issues. This may be an immediate plan of action. For example, in the short-term, device issues may be handled by:

- Removing a device from deployment and provide an alternative/back-up device to perform during an emergency incident
- Disconnecting a device's access to public safety resources

A combination of understanding the device issue and making a risk-based decision should be taken into consideration when deciding how to address device issues. For first responders, timing and impact of the remediation plan are a few key things to consider because a first responder may not want their device disconnected in the middle of an emergency incident. Communication of any remediation plans is important to share across the first responder team.

PSCR Engineers found that most mobile devices allowed for device administrators or users to apply some type of immediate response to address certain issues. Mobile tools, such as an EMM, can respond and update a device's configuration settings if there is a policy in place to address a particular issue or event. As mentioned before, an immediate change in device configuration could cause a disruption while a public safety official is responding to an emergency incident. Instead of applying immediate changes, an EMM can send notifications of any issues/anomalous events to the user/device administrator. With these notifications, the device administrator can make decisions to plan how to appropriately address the issue or event [12].

Wearable devices do not have the same flexibility with regards to updating device configurations. Most of the wearable devices reviewed by PSCR Engineers do not have a way to immediately apply fixes or update the device configurations. The lack of updatability may require device administrators to do additional planning for how to address wearable device vulnerabilities, when to decommission, and the purchase of new wearable devices. Devices that are able to be maintained, updated, and patched offer longer use and less of a need to purchase new devices.

Having a plan prepares public safety officials with methods to address device issues when they occur. Using an effective plan will help prevent first responders in the field from using devices potentially vulnerable to attack. Communication of any planned remediation keeps all public safety officials aware and allows everyone to plan/prepare accordingly.

Security Objectives: Ease of Management, Healthy Ecosystem

Test References: B.1.11, B.2.7

5.1.5 Recover - from issues by implementing the plan and constantly improving

After establishing a plan to handle issues/events, it is important to implement those plans/procedures to restore mobile and wearable devices affected by a cybersecurity issue/event. Additionally, any remediation of issues should be tested to ensure the issue is resolved as desired and does not impact device functionality. Device administrators should also take note of any lessons learned from the issue/event and from applying the remediation. Once again, communication is key here during and after recovery.

Some device issues require more time and consideration. Some example remediations that may require more planning and preparation include:

- Patch/update of a device and redeployment
- Decommission/dispose of a device and device replacement

Device vendors may provide an update policy and/or schedule. This was commonly provided amongst mobile devices. Updates/Patches to vulnerabilities are typically not applied automatically to mobile and wearable devices unless specified to do so. First responders may not want automatic updates because this could disrupt activities at an emergency incident. Without automatic updates, public safety device administrators can plan an appropriate schedule to apply changes to a public safety mobile and wearable devices. Wearable devices often did not have an update policy/schedule or were not capable of being updated at all. A risk analysis may be necessary to decide how to handle the wearable device issues/vulnerabilities. If, for example, a wearable device is unable to be updated/patched to address a high-risk issue/vulnerability, then the device may need to be decommissioned. Device administrators will then have to consider device replacement.

Implementing the plan to address device issues assists with protecting first responders and reducing risks to being vulnerable to attack and device malfunctions. Advanced planning for more impactful changes, such device updates and patches ensures that device maintenance doesn't interfere with first responder daily activities. Applying fixes on a schedule and preparing for decommission/device replacement ensures first responders have a device available to use during emergencies. Testing devices will check to see that the issue is remediated as desired and that any changes do not impact the device's functionality. The lessons learned throughout the recovery process can be used to improve your plan to address future device issues, more efficiently or before they occur. The fewer issues first responders need to address, the more they can focus on their daily live saving activities. Communication amongst all public safety officials involved helps with the following:

- Understanding what the device issue and why it is important to make changes to address the issue
- Scheduling an appropriate time for device maintenance that doesn't impact a first responder's work schedule
- Teaching/Learning any significant nuances to device functionality after the remediation is applied
- Ensuring the first responder is confident and comfortable using the device

Security Objectives: Healthy Ecosystem

Test References: B.1.9, B.1.11, B.2.7, B.2.9

6 Conclusion

Using the public safety security objectives defined in NISTIR 8196, PSCR Engineers analyzed the security capabilities of public safety mobile and wearable devices. The security objectives assisted in framing the test plan used to analyze the devices. The test analysis of devices fed into the development of suggestions and guidance for future public safety mobile and wearable devices.

The guidance derived from the test analysis, leverages the Cybersecurity Framework Functions to summarize and easily communicate the guidance to various levels within public safety organizations. PSCR Engineers suggest the following high-level guidance for public safety officials interested in acquiring mobile and wearable devices: *Identify* your public safety needs and devices; *Protect* yourself by applying security and training users; *Detect* issues by logging and monitoring your devices; *Respond* with a prepared plan; *Recover* by implementing the plan and constantly improving. In addition to this high-level guidance, PSCR Engineers detail specific information and features that should be taken into consideration to accomplish the guidance.

Throughout the analysis of mobile and wearable devices, PSCR Engineers found that smart mobile devices have advanced greatly over the years and are capable of meeting most of the public safety security objectives. Mobile technology still has room for improvement when it comes to capabilities, such as rogue base station detection. Wearable devices are still being introduced to the public safety market and due to their limited functionality, wearable devices struggle to meet some of the public safety security objectives. Wearable device information was inconsistently provided in manuals and many devices lack the ability to be updated or reconfigured to apply different security settings. Some wearable devices interact with an API, which allows a little more flexibility in gathering information or applying different settings. While Bluetooth specifications are constantly being improved and updated, commercially available wearables still seem to use older versions of Bluetooth, with minimal security levels. Overall, PSCR Engineers found that few devices are built with features that are specific to public safety, such as a ruggedization rating that meets the needs of firefighters.

Through this security analysis and guidance, PSCR Engineers strive to assist public safety officials interested in acquiring mobile and wearable devices that meet their security objectives. This information may also prove informative to device manufacturers that are interested in building devices that meet the public safety security objectives and include features to support our first responders. PSCR Engineers suggests the following publications as supplemental guidance for public safety mobile and wearable devices:

- NISTIR 8196, *Security Analysis of First Responder Mobile and Wearable Devices* [1]
- NISTIR 8080, *Usability and Security Considerations for Public Safety Mobile Authentication* [18]
- NISTIR 8228, *Considerations for Managing Internet of Things (IoT) Cybersecurity and Privacy Risks* [4]
- NISTIR 8259, *Foundational Cybersecurity Activities for IoT Device Manufacturers*[5]
- NISTIR 8259A, *IoT Device Cybersecurity Capability Core Baseline* [22]
- NIST SP 800-124 Revision 2, *Guidelines for Managing the Security of Mobile Devices in the Enterprise* [6]

- NIST SP 1800-13, *Mobile Application Single Sign-On: Improving Authentication for Public Safety First Responders* [19]
- NISTIR 8181*, Incident Scenarios Collection for Public Safety Communications Research: Framing the Context of Use* [20]

791 # References

[1] Middle Class Tax Relief and Job Creation Act of 2012, *PUBLIC LAW 112–96*, February 22, 2012. http://www.gpo.gov/fdsys/pkg/PLAW-112publ96/pdf/PLAW-112publ96.pdf [accessed 11/19/17].

[2] Franklin JM, Howell G, Ledgerwood S, Griffith J (2018) Security Analysis of First Responder Mobile and Wearable Devices (National Institute of Standards and Technology, Gaithersburg, MD), NIST Interagency or Internal Report (IR) 8196. https://doi.org/10.6028/NIST.IR.8196-draft

[3] National Institute of Standards and Technology (2019) Cybersecurity Framework. Available at https://www.nist.gov/cyberframework/

[4] Boeckl KR, Fagan MJ, Fisher WM, Lefkovitz NB, Megas KN, Nadeau EM, Piccarreta BM, Gabel O'Rourke D, Scarfone KA (2019) Considerations for Managing Internet of Things (IoT) Cybersecurity and Privacy Risks. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Interagency or Internal Report (IR) 8228. https://doi.org/10.6028/NIST.IR.8228

[5] Fagan M, Megas KN, Scarfone K, Smith M (2020) Foundational Cybersecurity Activities for IoT Device Manufacturers (National Institute of Standards and Technology, Gaithersburg, MD), NIST Interagency or Internal Report (IR) 8259. https://doi.org/10.6028/NIST.IR.8259

[6] Franklin JM, Howell G, Sritapan V, Souppaya MP, Scarfone KA (2020) Guidelines for Managing the Security of Mobile Devices in the Enterprise. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-124, Rev. 2 (Draft). https://doi.org/10.6028/NIST.SP.800-124r2-draft

[7] Cichonski JA, Franklin JM, Bartock MJ (2016) Guide to LTE Security. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-187. https://doi.org/10.6028/NIST.SP.800-187

[8] 3rd Generation Partnership Project, *Security architecture and procedures for 5G System,* 3GPP TS 33.501 V15, 2018. http://www.3gpp.org/ftp/specs/archive/33_series/33.501/ [accessed 6/11/18]

[9] National Institute of Standards and Technology (2001) Security Requirements for Cryptographic Modules. (U.S. Department of Commerce, Washington, DC), Federal Information Processing Standards Publication (FIPS) 140-2, Change Notice 2 December 03, 2002. https://doi.org/10.6028/NIST.FIPS.140-2

[10] Franklin JM, Howell G, Boeckl K, Lefkovitz N, Nadeau E, Shariati B, Ajmo J, Brown CJ, Dog S, Javar F, Peck M, Sandlin K. (2019) DRAFT Mobile Device Security: Corporate-Owned Personally-Enabled (COPE) (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 1800-21. https://www.nccoe.nist.gov/projects/building-blocks/mobile-device-security/corporate-owned-personally-enabled

[11] (n.d.). (Tenable) Retrieved February 25, 2020, from https://www.tenable.com

[12] *CVE security vulnerability database. Security vulnerabilities, exploits, references and more*. (n.d.). Retrieved 02/24/2020, from https://www.cvedetails.com/

[13] *Encryption : Android Open Source Project*. (2020, January 6). (Android) Retrieved February 25, 2020, from https://source.android.com/security/encryption

[14] Franklin JM, Bowler K, Brown CJ, Dog S, Edwards S, McNab N, Stelle M (2019). *Mobile Device Security Cloud and Hybrid Builds*. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 1800-4. https://doi.org/10.6028/NIST.SP.1800-4

[15] *iOS Security: iOS 12.1*. (2018, November). (Apple) Retrieved February 26, 2020, from https://www.apple.com/chde/business/docs/site/iOS_Security_Guide.pdf

[16] *Security-Enhanced Linux in Android : Android Open Source Project*. (2020, January 6). (Android) Retrieved February 25, 2020, from https://source.android.com/security/selinux

[17] Padgette J, Bahr J, Holtmann M, Batra M, Chen L, Smithbey R, Scarfone KA (2017) Guide to Bluetooth Security. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-121, Rev. 2. https://doi.org/10.6028/NIST.SP.800-121r2

[18] Choong Y-Y, Greene KK, Franklin JM (2016) Usability and Security Considerations for Public Safety Mobile Authentication. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Interagency or Internal Report (IR) 8080. https://doi.org/10.6028/NIST.IR.8080

[19] Fisher WM, Grassi P, Barker WC, Dog S, Jha S, Kim W, McCorkill T, Portner J, Russell M, Umarji S. (2019) Mobile Application Single Sign-On: Improving Authentication for Public Safety First Responders. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 1800-13. https://www.nccoe.nist.gov/publication/1800-13/

[20] Choong Y, Dawkins S, Greene K , Theofanos M, *NISTIR 8181- Incident Scenarios Collection for Public Safety Communications Research: Framing the Context of Use* NISTIR 8181, National Institute of Standards and Technology, June 2017. https://doi.org/10.6028/NIST.IR.8181

[21] Ogata MA, Franklin JM, Voas JM, Sritapan V, Quirolgico S (2019) Vetting the Security of Mobile Applications. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-163, Rev. 1. https://doi.org/10.6028/NIST.SP.800-163r1

[22] Fagan M, Megas KN, Scarfone K, Smith M (2020) IoT Device Cybersecurity Capability Core Baseline (National Institute of Standards and Technology, Gaithersburg, MD), NIST Interagency or Internal Report (IR) 8259A. https://doi.org/10.6028/NIST.IR.8259A

[redacted]

Appendix A—Acronyms

Selected acronyms and abbreviations used in this paper are defined below.

2G	2nd Generation
3G	3rd Generation
3GPP	3rd Generation Partnership Project
4G	4th Generation
5G	5th Generation
AES-CCM	Advanced Encryption Standard-Counter with CBC-MAC
APCO	Association of Public Safety Communications Officials
BLE	Bluetooth Low Energy
CBC-MAC	Cipher Block Chaining Message Authentication Code
DHS	Department of Homeland Security
ECDH	Elliptic-curve Diffie–Hellman
EMM	Enterprise Mobility Management
EMS	Emergency Medical Services
EMT	Emergency Medical Technician
FIPS	Federal Information Processing Standards
GSM	Global System for Mobile Communications
IP	Ingress Protection
IP	Internet Protocol
IR	Interagency Report
IoT	Internet of Things
ITL	Information Technology Laboratory
LE	Low Energy
LEO	Law Enforcement Officer
LMR	Land Mobile Radio
LTE	Long Term Evolution
MHz	Megahertz
MitM	Man in the Middle
MTD	Mobile Threat Defense
MAV	Mobile Application Vetting
NFC	Near Field Communication
NIST	National Institute of Standards and Technology
NPSBN	Nationwide Public Safety Broadband Network
OS	Operating System
OUI	Organizationally Unique Identifier
PAN	Personal Area Network
PIN	Personal Identification Number
PSCR	Public Safety Communications Research
RFID	Radio-Frequency Identification
SP	Special Publication
SSO	Single Sign-on
UI	User Interface
VPN	Virtual Private Network

Appendix B—Tests and Results

The type of testing performed for this analysis includes an understanding of the type and state of the software that is pre-installed, the vulnerabilities residing within the device, and the types of secure technologies included within the devices. This effort will also assist with understanding what type of external certifications and testing occurs for these devices, such as the Ingress Protection (IP) ratings.

This section provides the test plan used to analyze the security capabilities of the device. Below is an outline of the layout for each test case description:

- **Test Number: Test Name** – Each test is numbered and given a name with summarizes the purpose of the test.
- *Security Objective* – The objective of each test is mapped to one or more of the security objectives from NISTIR 8196
- *Test Description* – The test description describes the information the test will provide in relation to the security analysis of the mobile and wearable devices
- *Test Procedures* – PSCR Engineers documented the procedures used to perform each test. These procedures provide insight into how these tests can be replicated for personal analysis
- *Test Outcome* – After completion of each test, the engineers documented the outcome.
- *Analysis* – The results of each test are reviewed for potential impacts and future considerations for public safety. This analysis also includes gaps found as a result of the test.
- *Guidance* – Finally, each test concludes with suggested guidance for how to address the Security Objective(s) and concerns discussed in the Analysis. This guidance also includes potential benefits to implementing the provided guidance.

B.1 Mobile Test Results

B.1.1 Test 1: Obtain General Hardware Information

Security Objective(s): Ease of management of the mobile device, availability of technical specifications and the ability to maintain a healthy device ecosystem.

Test Description: Obtaining device documentation is the starting point towards understanding the basic operating functions of a mobile device. In this test, general information is gathered from the accompanied documentation contained in the box of the device, the manufacturer's web site or service provider's web site. Specific device information can also be obtained from the device's "About" or help settings. The intent of this test is to find hardware information/specifications and ease of access to assistive or help documentation.

Test Procedures: Check the accompanied documentation that shipped with the device. Record ease of access to the information and note the presence of quick-start guides, detailed guides, links to on-line resources. Check on-line web resources for ease of access, quick start guides and supplementary links. Check help and about settings on the device for on-line guides or search features. Note the presence of hardware information or specifications from these sources.

Test Outcome: General hardware information can be obtained directly from manufacturers' web sites. All devices tested contained a printed manual that contained information, quick start guides and/or links to web related resources. Both new and older devices contained at least one source of information to obtain general hardware information or help functions. A simple web search provided results to on-line resources to either the manufacturer or service provider of the mobile device. Newer devices had specific links to on-line help services from the mobile OS settings menu, however older devices only contained general hardware information from the "About" screen.

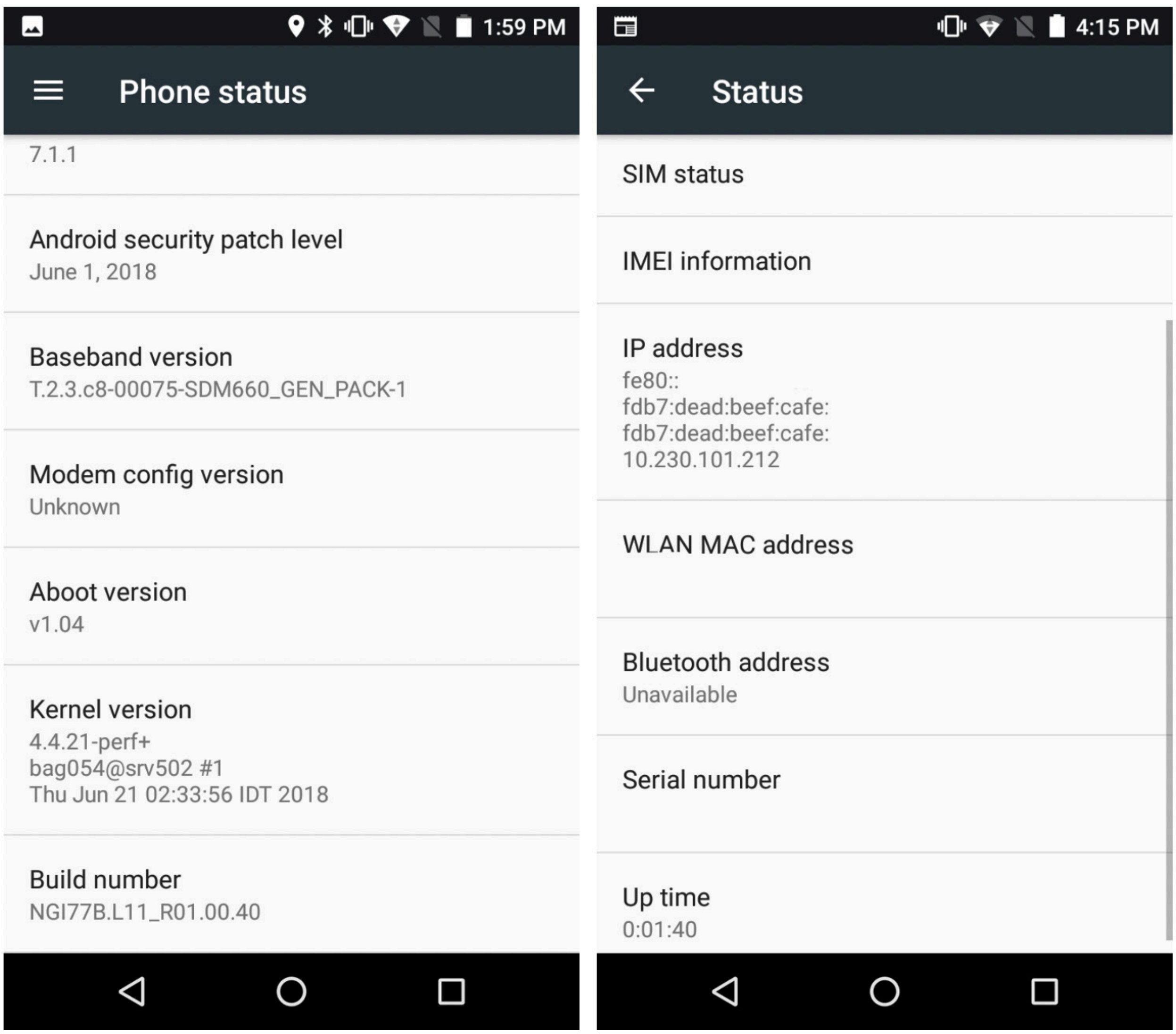


Figure 1 - Example 1: Device Information

Figure 1 shows the "About" and "Phone Status" screen on an Android device. These images show basic phone information including hardware platform, software versions and builds. This information can be used to obtain further information about the phone, either through web searches, manufacturers' web site or OS vendor web site. This information serves as a base reference for subsequent mobile tests performed in this guidance document.

Analysis: General hardware information for mobile devices is easy to obtain for both new and old mobile devices. With access to the mobile device, a user can find information within the Android "Settings" application under "About device" or "General > About" for iOS devices. This section provides information, such as the make and model of mobile device. Each device comes with a manual or data sheet within the packaging. Alternatively, a web search using the device's name and model provides direct links to the device's manufacturer and the device's manual and/or specification's sheet. Documentation accompanying the device contained general setup guidance that corresponded with the OEM OS and version contained on the device, out-of-the-box. Subsequent device updates from the OEM OS contained variations that did not match the insert documentation, however through intuition, settings often closely matched previous versions.

Gaps: Updates to the device's operating system may alter results, conflict or invalidate documentation sources. Device specifications may have slight variations among minor hardware revisions or among service providers that use the same manufacturer and model of a device. More in-depth web searches may be required by referencing the devices serial number or part number to ensure up-to-date and accurate documentation sources.

Guidance: Manufacturers should continue to provide the general hardware information for mobile devices and public safety users/device administrators should leverage this information as necessary (e.g., inventory, awareness, etc.). Documentation that accompanies the device should reflect the OEM OS contained on the phone, however valid web resources or links should be referenced so the user can obtain the latest update and guidance information.

Benefits: Easy access to the general hardware information allows the user to easily identify the device. Device serial numbers, OS version and model numbers can be used to gather more information to make configurations to the device, solve technical or usability issues, as well as secure the device. Device hardware on mobile devices is generally considered "non-upgradable" and therefore unlikely to deviate over the device's lifespan. Occasionally manufacturers may perform minor hardware revisions though the device's lifespan and is often reflected in the device's serial or hardware model number.

B.1.2 Test 2: Obtain General Software Information

Security Objective(s): Ease of Management, Network Agility and Healthy Device Ecosystem

Test Description: This test will identify the name and software version of operating system and major applications that are shipped with the device. This will also attempt to understand the protocol versions for the primary wireless protocols (i.e., Wi-Fi, Bluetooth, and Cellular).

Test Procedures: Device information is obtained via documentation obtained in using the methodology described in Test 1. OS software information can be obtained on Android devices under Settings > About or on iOS General > About. Web searches for the specific OS version were performed to find information from the OS software provider. Network capabilities are obtained via the device's technical specifications documentation or manufacturer web site. Applications that ship with the device are identified under the Settings > Applications (Apps) listing and/or within the "apps" menu. Apple iOS displays a list of apps under the settings menu.

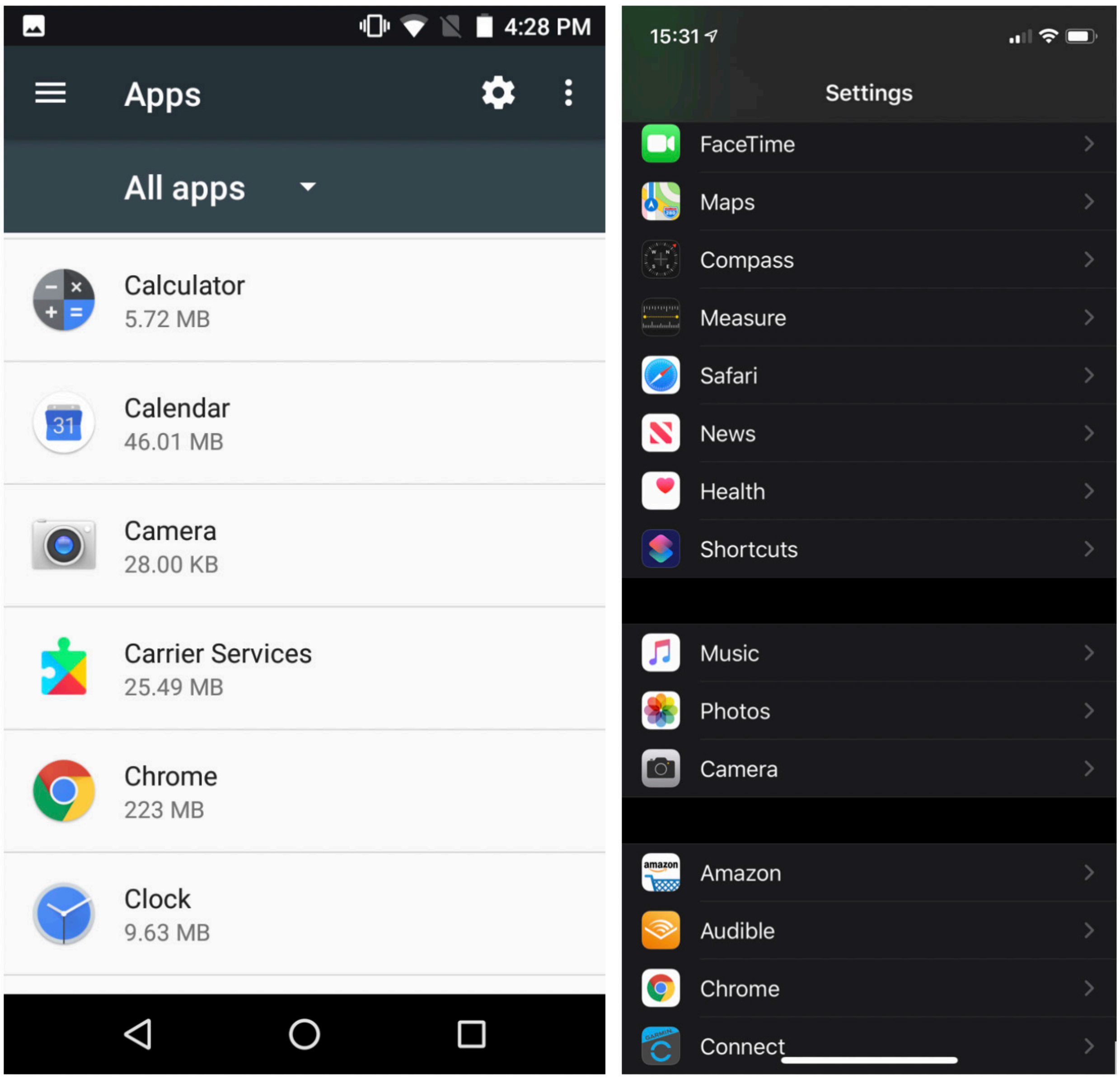


Figure 2 - id applications listing (left), iOS applications listing (right)

Test Outcome: Basic information can be gathered from the device through the use of the user interface or graphical user interface. Of the devices analyzed, the OEM OS was not at the latest patch level. Upon connecting to the internet, devices automatically downloaded new OS versions and/or patches that corrected most known vulnerabilities and added features. While pre-provisioned devices are at risk upon unboxing, it is commonly an accepted risk and part of normal onboarding operations for enterprise and First Responder mobile devices.

Some Pre-installed applications are viewable to the user under the applications listing or under Settings menus. Of the observed applications, only one observed device revealed a remote-management application. Upon further inspection, the application is used as a remote-management and provisioning platform used by enhanced management services. Unlike most general consumer market devices, First Responder devices only included applications such as the default Google applications, First Responder focused applications and/or service provider installed applications.

All devices observed are capable of Wi-Fi, Bluetooth and Cellular network capabilities. Of the devices tested, only three mobiles were Band 14 capable, however all devices but two supported up to Bluetooth version 5 and Wi-Fi 802.11ac also known as Wi-Fi 5. None of the devices tested supported Wi-Fi version 802.11ax also known as Wi-Fi 6.

Analysis: Operating system and application data can be easily obtained through the Settings menu within the mobile device. Application data is found within the applications menu and/or the settings menu. Of the applications observed, those that are not part of the default OS installation are designed to assist or enhance the experience for Public Safety officials. Those applications are specifically designed for mobility services, such as talk groups, remote management or public safety specific data services. Complete network capabilities are not easily obtained via the OS settings; however, the general specifications of network capability are contained within the device documentation as described in Test 1. All devices supported protocols and capabilities to operate on cellular and Wi-Fi networks, however older devices lacked hardware capability necessary to connect to future network technology protocols and methods.

Gaps: Many of the default OS shipped applications are not necessary or applicable to the First Responder mission or enhance the goals of Public Safety. Likewise, supplementary applications shipped with the device do not reflect the entirety of Public Safety's needs to include Police, Firefighters or EMS. Also note that some default OS applications cannot be removed. Similarly, some applications "hide" as background processes or daemons and cannot be easily analyzed without 3rd party tools. Such applications do not appear within the user space of the OS.

Guidance: Software information including OS, general app inventory and network protocols should be readily available to the Public Safety. To leverage the NSPBN FirstNet Network, Public Safety mobile devices must have band 14 capability. The FirstNet NPSBN contains a certified list of applications and requirements for certification available from the FirstNet developer portal at https://developer.firstnet.com. Applications should only be installed from trusted platform providers, such as Android Google Play or Apple iOS App Store. Any applications not relevant to the needs of first responders should be uninstalled, where possible. Onboarding practices vary by organization and mobility device management (MDM) implementations, however it is recommended that new device onboarding be performed on an isolated network segment. Isolated network segments only contain crucial network connections necessary for device updating, application installation, federation and device integration.

Devices that are onboarded via the cellular interface should utilize private VPN connections for MDM integration.

Benefits: Accessibility to OS, application data and network capability allow the user to understand software and hardware capability of the device. These factors foster comprehension of the device's point in its lifecycle. Similarly, the presence of default applications in first responder devices should reflect the goal or mission of the device. Network capability and performance should adequately support the purpose of default applications to ensure resilience and reliability required of First Responders.

Mobile devices with Band 14 capabilities can utilize the NSPBN FirstNet network, which hosts reserved spectrum for public safety to remediate against any concerns of potential congestion due to mass communications transmissions that may occur on the traditional cellular networks. This congestion may be caused due a heavily populated area without the supported infrastructure, a major emergency incident where citizens are attempting to contact loved ones all at the same time.

Most mobile devices have multiple network capabilities. This provides network agility by allowing the device to alternate between Wi-Fi, Bluetooth or cellular if one network protocol is unavailable. Awareness of the network protocols available on mobile device allows Public Safety Officials to be aware any potential limitations to their network agility.

B.1.3 Test 3: Device Ruggedization Ratings

Security Objective(s): Device availability and integrity through survivability, healthy mobile ecosystem through continuous operation and ease of management in day-to-day operations.

Test Description: Implementation of ruggedization ensures durability for First Responder applications and survivability of day-to-day use. This test identifies the Ingress Protection (IP) ratings and any ruggedization information available for the device. Physical survivability of First Responder mobile devices ensures the integrity of responder data. IP ratings and certification ensure data integrity by reducing occurrence of device failure in extreme environments as well as reliable communications.

Test Procedures: Utilizing the methodologies described in Test 1, obtain metrics to determine any certifications of ruggedization. Through local observation, inspect any protective surfaces or covers that enhance device survival in demanding environments. Check any fortifications that ensure battery operation or temperature threshold parameters.

Test Outcome: Device ruggedization metrics and certifications are obtained through a combination of on-line documentation, product inserts and queries to the manufacturer technical support. Physical observations can also determine if a device is built specifically for First Responder applications. Attributes include, but not limited to, features such as protective glass, fortified case and high impact plastics. The most common ruggedization standard utilized is the MIL-STD-810G. Of the phones analyzed, only three handhelds claim conformation to MIL-

STD-810G, one rating was self-certified. All devices under analysis conformed to IP67 water ruggedization certification. One device is certified IP69, which includes high-temperature, high-pressure ruggedizations.

Analysis: Devices that conform to the MIL-STD-810G standard are generally bulky and contain rubber and/or hard plastics to fortify against impacts and drops. Devices that contain IP67 certification are not as easily discernable, however of the devices that contained the certification and contained a removable battery, supplementary seals, screws and latches are present to enhance protection against water. It may also be noted that of the devices tested, the removable batteries do not correlate to the same temperature thresholds as the mobile device. Survivability of the device does not necessarily correlate to operational ability through a first responder event.

Figure 3 - Example ruggedized device

Figure 3 is an example of a mission critical handsets that is typically bigger, with ruggedized features adapted for mission critical applications. Handsets may include additional interfaces than consumer-based handsets, such as buttons for push-to-talk, emergency request buttons, and switches to toggle between talk groups.

Gaps: Although ruggedization rating information is available in some form. There are no specific standards with regards to what is required for a public safety device. The ruggedization rating may differ per public safety personnel (i.e., law enforcement, firefighter, EMS).

Ruggedization ratings may only be held at face value due to non-conformality or non-regulation of IP or MIL implementations. Comparison analysis among ratings standards may be required (by the user) to determine if a device applies to their need(s).

Guidance: While high-grade ruggedization may be ideal, public safety mobile devices should meet the appropriate ruggedization ratings for their purposes. This information should be easily available for Public Safety to determine whether the ruggedization level of the device meet their desired needs. Such information should be provided within the product documentation or on the manufacturer web site. Mobile carriers often group mission critical devices as a separate offering and are presented on a different web page than standard consumer mobile devices. Public safety devices that do not require or contain additional OEM ruggedization may benefit from the application of a mobile case and/or screen protector.

Benefits: Ruggedization certification ensures that a mobile device is properly designed with extreme environments in mind. A public safety specific ruggedization certification or guide could be beneficial to assist public safety personnel in choosing a device with the appropriate ruggedization grade. For example, a law enforcement officer's device may not require the same heat resistant capabilities as a firefighter's device. Due to the occupational extremities required of public safety and first responders, ruggedization is required for day-to-day survivability and operation of the device.

B.1.4 Test 4: Obtaining Vulnerability Information from OS version and known databases

Security Objective(s): Availability of the mobile operating system, integrity of the mobile and user data and maintaining a healthy device ecosystem.

Test Description: The Analysis of the OEM software version can be verified against a list of vulnerabilities within public databases describing Common Vulnerabilities and Exposures (CVEs). While most cellular service providers and device manufactures provide patching and updates to help mitigate known CVEs, the application of updates are generally initiated by the end user. Older mobile devices, particularly those that are out of production cycle or end-of-life, may lack necessary updates and patches to ensure operating system integrity. Since many public safety mobile devices are built for longevity and incur higher costs to the user/first responder organization, the likelihood of use beyond the manufacturer lifetime is higher than normal consumer mobile devices. By comparing the current operating system with known CVE databases, it can be determined if operating system support is being provided and known vulnerabilities are being patched by the user, device manufacturer or service provider.

Google » Android » 7.1.1 : Security Vulnerabilities

Cpe Name:*cpe:/o:google:android:7.1.1*
CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9
Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending

Total number of vulnerabilities : **544** Page : 1 (This Page) 2 3 4 5 6 7 8 9 10 11

Copy Results Download Results

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2019-14783	264			2019-08-08	2019-09-25	2.1	None	Local	Low	Not required	None	Partial	None
On Samsung mobile devices with N(7.x), and O(8.x), P(9.0) software, FotaAgent allows a malicious application to create privileged files. The Samsung ID is SVE-2019-14764.														
2	CVE-2019-2179	190		Overflow	2019-09-05	2019-09-06	4.3	None	Remote	Medium	Not required	Partial	None	None
In NDEF_MsgValidate of ndef_utils in Android 7.1.1, 7.1.2, 8.0, 8.1 and 9, there is a possible out of bounds read due to an integer overflow. This could lead to local information disclosure with no additional execution privileges needed. User interaction is needed for exploitation.														
3	CVE-2019-2178	787			2019-09-05	2019-09-06	7.2	None	Local	Low	Not required	Complete	Complete	Complete
In rw_t4t_sm_read_ndef of rw_t4t in Android 7.1.1, 7.1.2, 8.0, 8.1 and 9, there is a possible out of bounds write due to a missing bounds check. This could lead to local escalation of privilege in the NFC service with no additional execution privileges needed. User interaction is not needed for exploitation.														

Figure 4 - Example Android CVEs

Figure 4 is an example of one of the CVE databases that contain extensive analysis for each Android or Apple iOS version. Many databases rate the severity of the CVE, vulnerability type and when or if a patch is available. This data can be cross-referenced with the current running version on the handset under test to ensure it is protected.[12]

Test Procedures: Obtain the OS version of the device and search for CVEs on known databases. Where possible, search for the specific OS build number to provide more refined results. Make specific note of the number of vulnerabilities in critical categories.

In this test it is important to note that results reflect the date that the test was conducted. Reiterations of these tests will result in different outcomes due to newly discovered vulnerabilities and the issuance of new CVEs. Likewise, before all tests were performed, all devices under test (DUT) were upgraded and patched to the latest available version from the manufacturer or service provider. It is also important to note that older versions of operating systems do not necessary mean less patching support. Adequate patching of both new and old operating systems is necessary to ensure device integrity. Gaps in patching, delays in patching or missing patches were not instigated in this study.

Test Outcome: Of all of the devices, only one mobile contained a patch level within three months of the date of the testing. While this resulted in fewer CVEs, many critical categories remained. Likewise, only one device contained an operating system and patch level that was no longer supported by the OS provider. Two of the devices tested contained Android Version 7.1.1 with different patch levels and one device contained version 6.0.1 with a patch level issued within the past 3 months of testing.

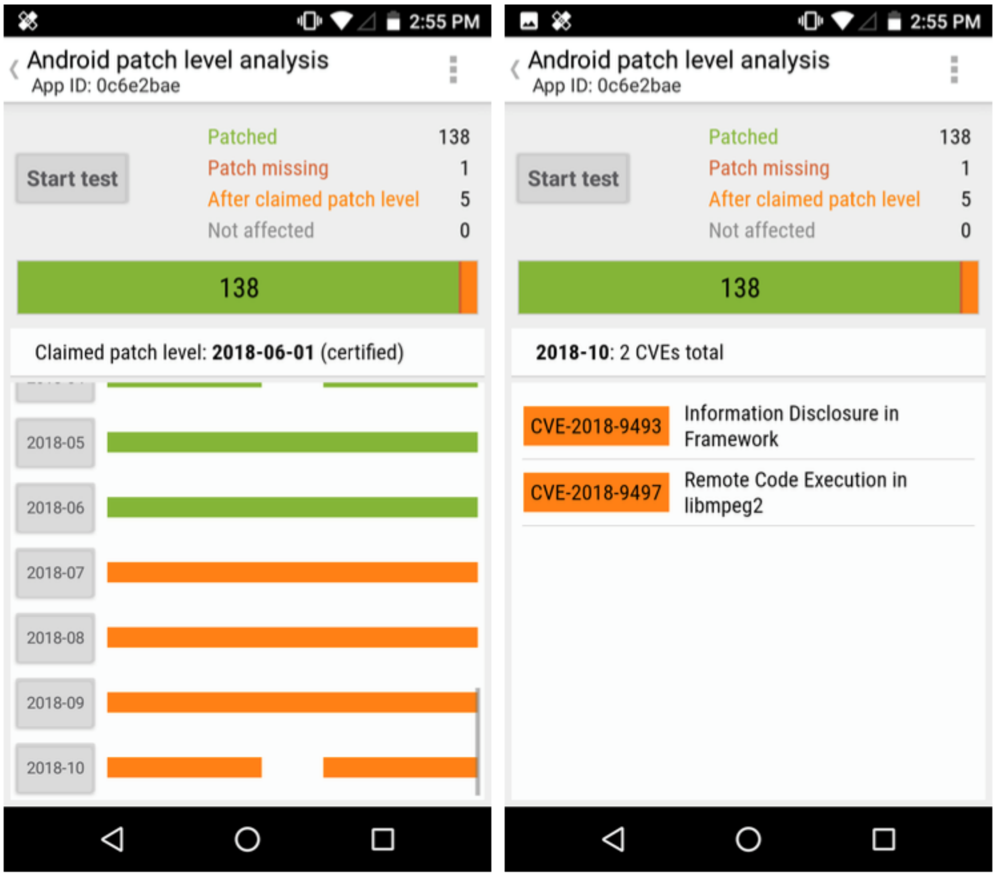


Figure 5 - Vulnerability scanner results

Vulnerability scanners, such as SnoopSnitch in Figure 5, can scan a device and provide patch analysis reports to inform the user of any potential vulnerabilities. The results in the above report, show two potential vulnerabilities. The device under test (DUT) is running Android version 7.1.1, patch level June 1st, 2018. No subsequent updates were available for this device, potentially putting the device at risk.

Analysis: CVE databases are easily accessible through online sources and patch level analysis tools are available for free use. Most CVEs can be mitigated through regular patching and updates. Those that can't be mitigated through patching must utilize alternative methods of protections, such as mobile threat defense and detection applications. While CVEs are easy to find and identify, the level of threat and user applicability may differ, depending on the device, OS and build. Some CVEs are listed as informal notifications that affect a large breadth of devices but may not directly affect the DUT.

Gaps: Individual patch levels may further be analyzed to determine if a specific software build contains vulnerabilities. Not all patch levels are publicly disclosed. Software builds may also be specific to a device, vendor, hardware platform and/or service provider. It may be difficult for a first responder to interpret what CVEs impact their device. The information presented is not always clear and concise for the average user and may require additional research. The requirement of additional time investment may not be feasible to most public safety groups.

Guidance: Enterprise administrators of public safety mobile devices should be aware of CVEs that pertain to current running versions. Since devices typically run under a common administration using a mobile device management (MDM) solution in enterprise scenarios, keeping devices up-to-date and patching CVEs is a cumulative task. Individually managed devices and personal devices are administered upon the discretion of the first responder and/or mobile ISP service provider. It is recommended to check for device software updates on a regular basis and apply those patches when available. Note that not all CVEs may be applicable to a specific device, nor may it be possible to address or patch the CVE. OS and patch-level information should be readily available to the device user at any time of inquiry.

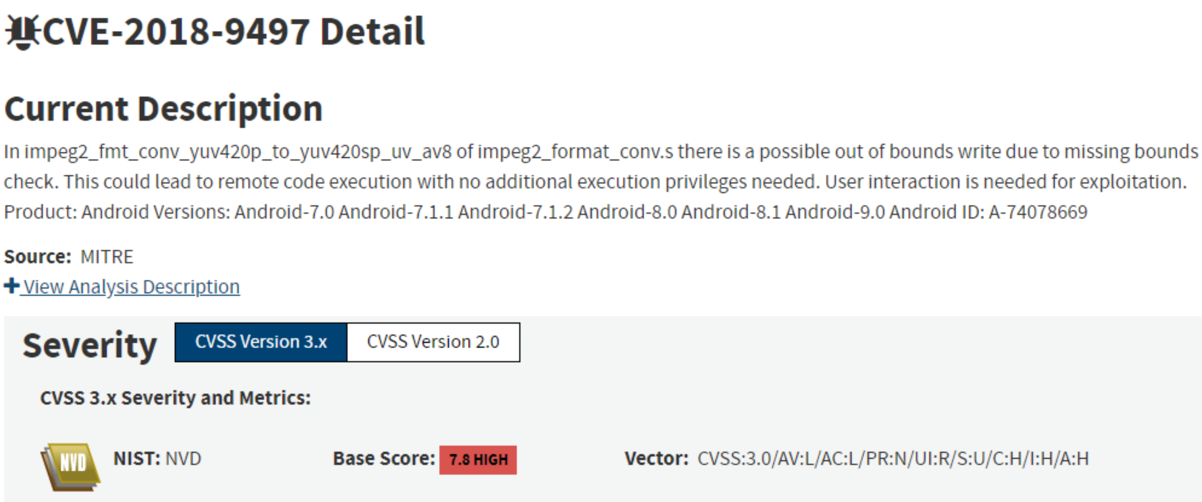


Figure 6 - CVE reference in National Vulnerability Database

Using one of the CVE's found in Figure 5, Figure 6, cross-references the CVE-2018-9497 ID in the NIST National Vulnerability Database to obtain more information about the unpatched vulnerability. Detailed information can be used to determine if a patch is available or if further action is needed to mitigate the risk.

Benefits: Analysis of known vulnerabilities informs the user of potential threats that the device may incur. This analysis allows the users to determine next steps to secure the device, if the device can be updated, if further protections are necessary or supplemental mitigation mechanisms must be employed.

B.1.5 Test 5: Vulnerability Scan via Mobile Threat Defense (MTD) Application

Security Objective(s): Device integrity, availability and health can be enhanced using a mobile threat defense application.

Test Description: Vulnerability scanning on a mobile device is commonly achieved using a 3rd party application downloaded from a mobile application store. Frequent use of an MTD ensures the integrity of both the mobile device operating system as well as any applications installed by the user, manufacturer or service provider. MTDs expedite and automate vulnerability scanning reducing time invested into searching for vulnerabilities. This test uses publicly available MTD applications to identify vulnerabilities within the mobile OS and applications shipped with the device. MTD information may be cross referenced with the results in Test 4 CVEs or via the manufacturers web site to ensure consistency among results. In most cases, the MTD will produce a report and prompt a notification of any potential threats to the mobile device.

Test Procedures: Download and install an MTD application that references CVE databases and provide applications ratings. Observe and compare the results, cross referencing patch databases.

Test Outcome: Overall, the 3rd party application found that all CVEs were patched at the current level (after the mobile device was updated) for three of the DUTs. The remaining devices contained less than five patched CVEs. The 3rd party application reported many “inconclusive” results for all the DUTs. Inconclusive indicates that the MTD could not find evidence of the patch related to the OS. The number of pre-installed/OEM apps and number of files analyzed by the MTD varied among all the devices tested. Only one false-positive result was reported among the OEM applications installed. The MTD reported a potential command and control application. The application in question was used for device remote provisioning and deployment. Referring to Test 2, due to the unique application of First Responder mobile devices, pre-installed applications represented less risk compared to consumer mobile devices.

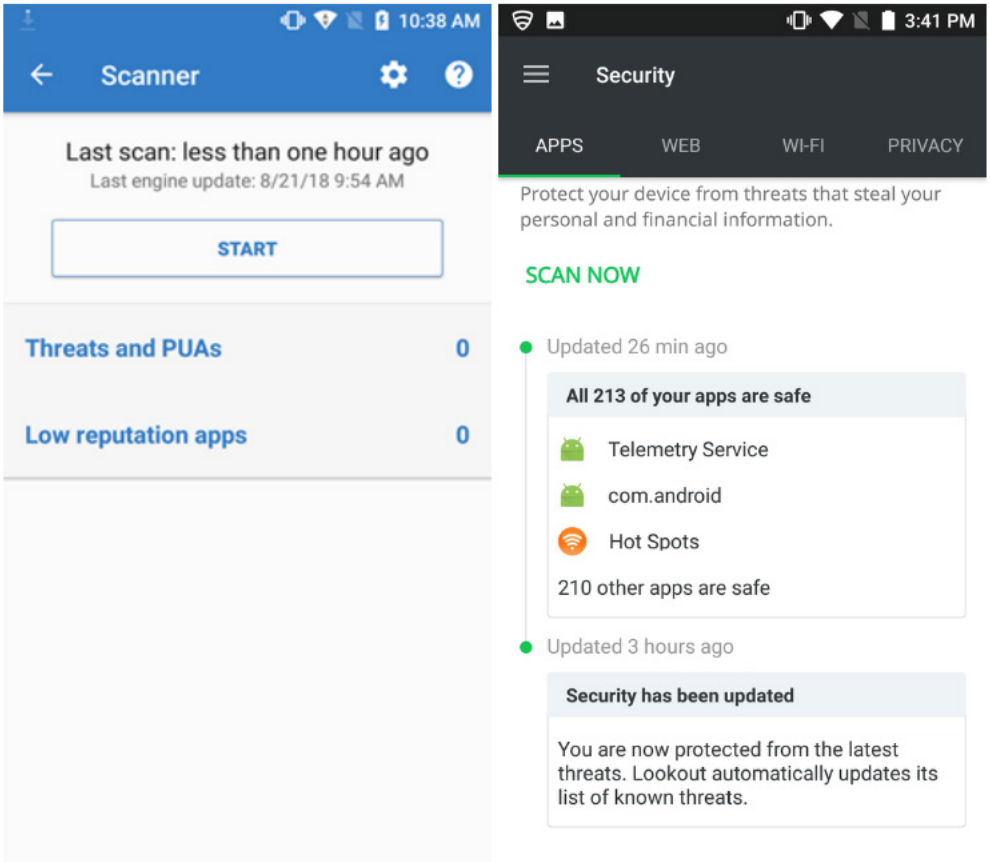


Figure 7 - MTD scan results

MTD software can scan device for app-based vulnerabilities in addition to systems scans (see Figure 7). Most MTD applications can be configured to run on a continuous or "active" basis to intercept malicious apps in real time. Regular, full-system scans should be running daily to ensure existing apps have not been compromised.

Analysis: MTD software is easily obtained through OS application stores and can be configured to scan the device automatically on a regular basis. Most MTD applications will also provide active application analysis, web browsing security, connection monitoring and privacy settings optimization. When a threat is detected, the application immediately informs the user of the threat and will take action to mitigate the problem. Full system scans give the user a detailed report and accounting log of executed actions. MTD application updates and definition updates occur upon installation of the MTD and check on a regularly preconfigured schedule.

Gaps: Results differ among MTD software providers. MTD definitions must be updated to ensure latest vulnerabilities are defined and discoverable. Users and administrators must be aware that malware on an infected device may alter results from MTD applications. The occurrence of false-positive results also varies among MTD software providers. MTDs are powerful tools to help the user secure their device, however human intervention and judgement must be made to determine if an unpatched CVE presents a risk to the device. Analysis of CVEs can be time consuming and requires familiarity with cybersecurity related technologies to determine if a CVE presents a risk.

Guidance: For both public safety enterprise administrators and individual first responder users, it is recommended to consider using mobile security tools, such as the MTD application tool used in this test. MTD applications can be used in conjunction with an EMM solution to ensure a complete device health ecosystem. An MTD tool scans the mobile device and alerts the user/administrator of potential vulnerabilities. In addition to EMM, MDM and MTD solutions, users can also consider Mobile Application Vetting Services. More information can be found in NIST SP 800-124 rev. 2 *Guidelines for Managing the Security of Mobile Devices in the Enterprise*[6]. Daily scans should be performed to ensure no new threats are present. User and/or administrators should be alerted if a threat is present. A log or summary of the scan information should be presented in the application or remote management software upon request. Most MTD applications offer both “free-to-use" and paid tier levels. Typically, the “paid” tier offers greater protections such as zero-day mitigations and enhanced device management optimizations. At the very least, first responders should install and run the free MTD application, however it is recommended to utilize a paid application service to ensure the greatest level of protection for the first responder device.

Benefits: Mobile security tools such as MTDs inform the user of potential vulnerabilities and low reputation applications installed on the mobile device. Information and awareness are beneficial to public safety device administrators by allowing them to take necessary action to address any potential vulnerabilities or concerns. By addressing these vulnerabilities, public safety officials can avoid any potential compromise of a mobile device and its capabilities. Scanned app information can be used to make decisions on an app trustworthiness or weigh the benefits of the app verses potential risk of using the app. This decision can prompt further investigation of the app in question and the data that it has access to. Maintaining logs or summary of information from the mobile security tools can assist with future policy analysis and risk considerations.

B.1.6 Test 6: External Fingerprinting

Security Objective(s): Device integrity and confidentiality can be determined through use of network-based scanning tools.

Test Description: Device integrity can be verified by performing external scanning and fingerprinting over a network connection. Most internet connected devices utilize application sockets to communicate using either Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) transport mechanisms. Open TCP or UDP sockets on a device may indicate a “listening” service or application on the mobile device. Network sockets are typically used for enhanced user experience and network operation/functionality. In some cases, an open socket may be used to exploit a device application or be indicative of malicious applications on the mobile device. Knowledge of open service ports may lead to further analysis of the application or services requesting the service port. Fingerprinting a device is often the initial stage of information gathering before it is attacked over a network.

Test Procedures: Identify the Wi-Fi IP address of the mobile device. Using a network-based scanning tool, such as nmap, scan the DUT. Determine which, if any network sockets are open, what services are running on the ports and if the device OS and/or hardware can be identified.

Test Outcome: Analyzed devices displayed open ports via Wi-Fi scanning with nmap. Open ports did not indicate a listening service to establish a session with the specified TCP/UDP socket. Of the devices tested, dhcps UDP/67, dhcpc UDP/68 and zeroconf were observed as common open ports. All three ports are typically used for device configuration and IP assignment. Although all three ports were “open” the scan indicated that the devices did not respond or actively closed the connection. One device indicated SIP TCP/5060 service port, commonly used for Voice over IP applications. Two of the devices scanned indicated open imap TCP/143 and TCP/993 and pop3 ports, TCP/110 and TCP/995 typically used for email services. Overall, potential findings indicate the presence of applications, such as pop and sip services, that could be further exploited. In order to minimize exposure, unnecessary applications and services should be disabled or removed. The scan could not indicate what applications used these open ports. Further investigation of running applications should be investigated to determine the need of the application. Device hardware could only be extrapolated by manufacturer due to the 24-bit Organizationally Unique Identifier (OUI) of the Wi-Fi MAC address.

```
:~$ sudo nmap -sS -sU -PN 10.230.101.124

Starting Nmap 7.60 ( https://nmap.org ) at 2018-08-21 17:13 MDT
Nmap scan report for 10.230.101.124
Host is up (0.081s latency).
Not shown: 1996 closed ports
PORT     STATE         SERVICE
5060/tcp filtered      sip
67/udp   open|filtered dhcps
68/udp   open|filtered dhcpc
5353/udp open|filtered zeroconf
MAC Address:

Nmap done: 1 IP address (1 host up) scanned in 1010.91 seconds
```

Figure 8 - NMAP port scan

Network based scanning tools, such as NMAP (see Figure 8), can provide insight of open ports, indicating a potential running service on the device. Other information can be extrapolated from in-depth scans, such as OS type, running applications and hardware information.

Analysis: Network based scanning tools utilized in this test returned results indicating that the devices filtered any open network ports. While this does indicate an active running service, the device actively mitigated any attempts to probe or exploit those ports. In general, mobile devices, in their default configuration, protect against network-based attacks using methods built-in to the devices’ OS. However, the manufacturer of the device can be easily obtained through the devices MAC OUI if the device does not support MAC address randomization. The device manufacturer of all of the tested devices was determined, however detailed information, such as device type and actual running applications, could not be determined.

Gaps: Network based port scanning does not provide information on the specific application using the open port. Host based tools may be used to determine the nature of the application and legitimacy of its presence on a device. Accordingly, if a device has multiple network interfaces, e.g. Wi-Fi, Bluetooth and/or LTE data connection, all interfaces must be analyzed to determine listening service ports. Depending on the network configuration, accurate results may be skewed due to intermediate network devices, filters, firewalls or other middleware boxes.

Guidance: Devices under a common administration should be routinely scanned over a managed local network for potential network vulnerabilities. Since most broadband mobile devices operate over LTE networks, the opportunity to externally scan the device on a locally controlled Wi-Fi network may not be possible. If a device cannot be regularly scanned over a locally controlled Wi-Fi network, an MTD should be used and a mobile management policy should be implemented to ensure the device can be periodically scanned. MDM solutions, as explained in Test 7, can perform detailed device scans if the mobile device can connect to the internet. Devices not under a common administration should run an MTD on a daily basis. Only applications required for mission critical operations should be present on the device.

Benefits: Network scanning allows the user to determine how network based or “outside” hosts may connect to the mobile device. Scanning reveals potential exploitable sources of entry as well as applications that allow external access to the device.

B.1.7 Test 7: External Vulnerability Scan

Security Objective(s): Mobile device availability, confidentiality and integrity.

Test Description: Vulnerability scanning is the next step beyond external fingerprinting and is often executed to ensure device integrity. Vulnerability scanning suites utilize scripts and automated methods to determine if an open network port or service can be exploited. This level of scanning is much more intrusive but can provide in depth analysis concerning a device’s network security posture. An external vulnerability scan is often part of an information gathering phase before it is attacked.

Test Procedure: Determine the Wi-Fi IP address of the DUT. Using a network-based vulnerability scanner, execute a scan to determine if the open ports in Test 6 are exploitable and if OS information can be enumerated.

Test Outcome: Test results indicated only informative level findings providing network enumeration values, such as hostname, IP address and network diameter information. No know vulnerabilities were discovered, indicating that the ports discovered in Test 6 were not active listening services. Overall indications reveal that external, network originated attacks on mobile OS services do not represent high risk for the DUT. Specific OS information could not be determined without an authenticated scan. The scanner could only determine that the mobile devices run a variant of Linux.

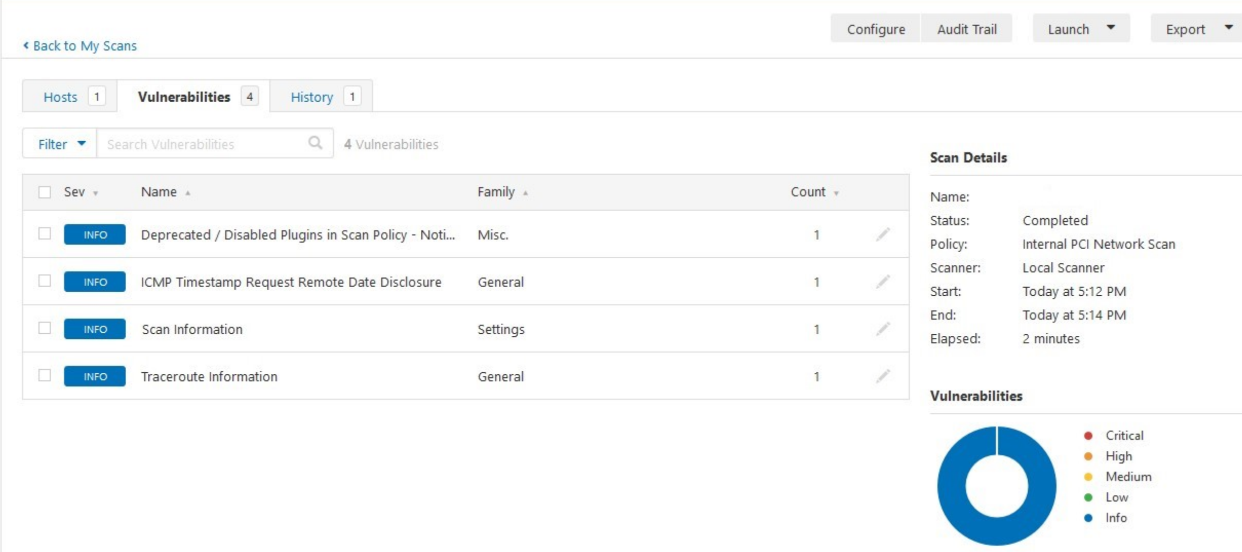


Figure 9 - External vulnerability scan results (1)

External vulnerability scanners can perform detailed analysis against networked hosts, including mobile devices (see Figure 9). Authenticated scans can also be performed to provide an administrative level scan against the device. Authenticated scans may require installation of additional apps and device policy modifications to maximize results. Scans should only be performed over Wi-Fi connections under locally controlled administration.

Analysis: Observed devices produce informational findings using unauthenticated scans. Authenticated scans using an MDM solution produced detailed analysis that included CVE checks against OS patch levels and application versions. Authenticated scans produced warnings concerning installed applications, including those requiring updating and potential low reputation apps.

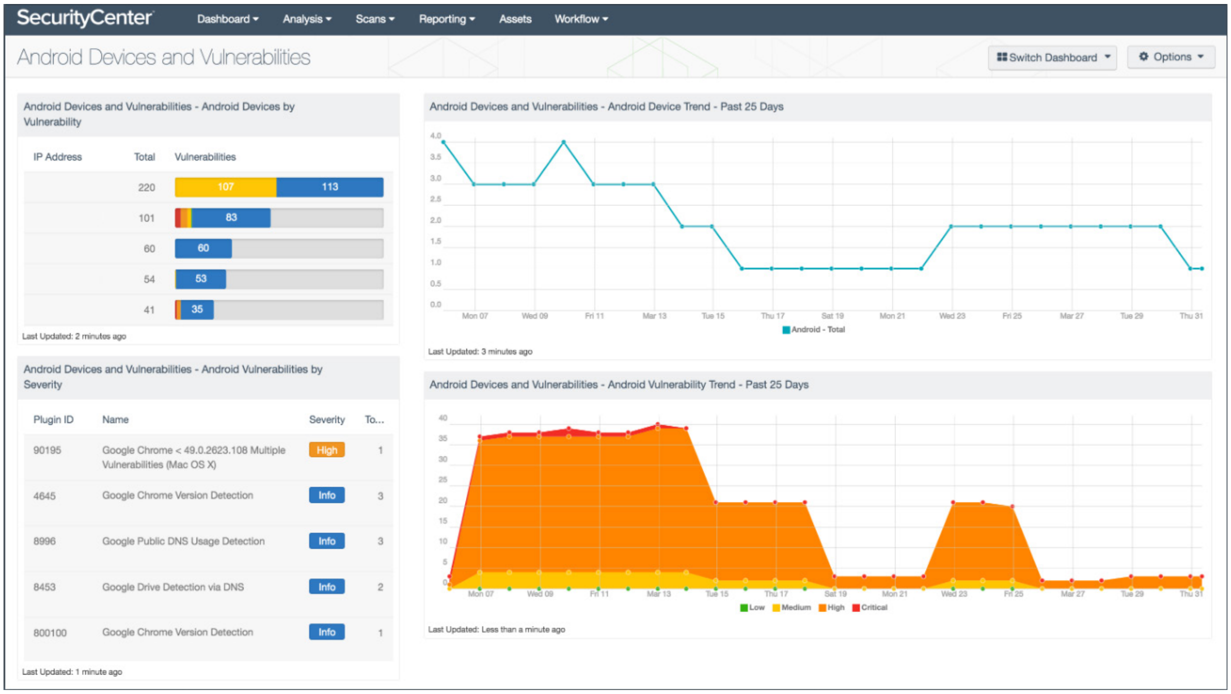


Figure 10 - External vulnerability scan results (2)

Another example of external vulnerability scanning can be found in Figure 10 which is a Nessus Android vulnerability report.[11]

Gaps: Authenticated scans provide enhanced scanning by remotely logging into the DUT. Most mobile devices do not allow authenticated scans without root account access, which is often restricted or prohibited by the manufacturer or service provider. Like Test 6, all network ports should be analyzed to determine a device's integrity.

Guidance: Like guidance in Test 6, devices under a common administration should be routinely scanned over a managed local network for potential network vulnerabilities. An MDM solution and mobile management policy should be implemented to ensure periodic scanning. Only applications required for mission critical operations should be present on the device. Non-essential applications should be removed to ensure no external network connections can be made to the device. Authenticated scans are typically performed on devices running an MDM and an associated scanner plugin. The scanner application works in conjunction with the MDM application to provide detailed analysis of device applications and patches. Devices that cannot be scanned or are scanned using unauthenticated methods should have a MTD installed and scheduled to run daily. For more information on MDM implementation, consult NIST SPECIAL PUBLICATION 1800-4, "Mobile Device Security Cloud and Hybrid Builds." This publication includes detailed procedures on how to architect enterprise-class protection for mobile devices accessing corporate resources.[14]

Benefits: External vulnerability scans allow the user to determine if the mobile device is exploitable. When possible, the scanning software will attempt to determine OS type, hardware platform, exploitable applications, services and exploit unpatched systems.

B.1.8 Test 8: MAC Address Randomization

Security Objective(s): Mobile device confidentiality

Test Description: Device confidentiality and autonomy can be maintained using MAC address randomization. Static MAC addresses can be used as a mechanism to track First Responders between networks and potentially build a profile of users, locations and network activity. Traditionally, IP networked devices do not randomize MAC address due to serviceability concerns, such as domain name resolution, MAC based authentication, access control, MAC-based billing. MAC address randomization may also be limited due to hardware, OS and device limitations.

Test Procedure: Check the device's MAC address under the Settings menu. Connect to a Wi-Fi network and compare the MAC address to the address in the settings menu. Perform the same analysis on different Wi-Fi networks. Using an external Wi-Fi network sniffer, capture traffic to and from the device. Analyze the packets and compare the MAC address in the capture with the MAC address under the Settings menu.

Test Outcome: Over the air packet captures confirmed that MAC address changed between different Wi-Fi networks. Only the devices running Android 8 and IOS 8 or greater performed the MAC address change. Older devices did not have a menu option to use MAC address randomization. Over-the-air captures confirmed that older devices did not change their MAC address.

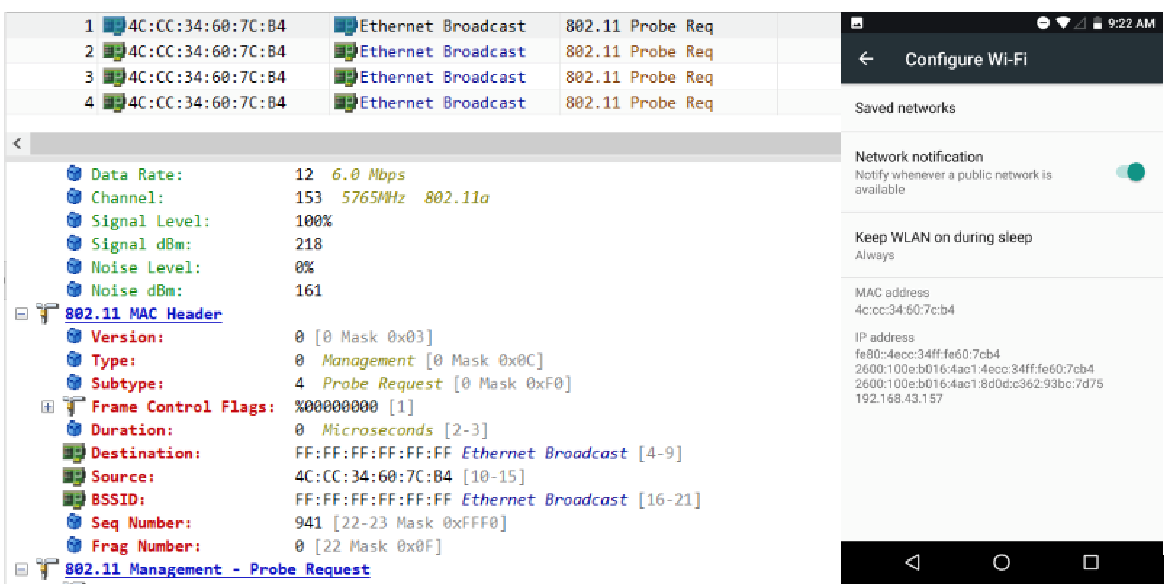


Figure 11 - Mac address randomization analysis

Figure 11's over-the-air capture shows MAC address on an Android device with MAC address unchanged. Note device MAC address in the 802.11 MAC Header Source (left), matches the device MAC address 4C:CC:34:60:7C:B4 (right)

Analysis: Starting in Android version 8, MAC address randomization can be implemented by the Wi-Fi chip vendor and the Android application developer can implement the IWifiStaIface.setMacAddress() HAL method to support this feature. Similarly, MAC address randomization was enabled starting in iOS version 8, but it is enabled only during specific user configurations. iOS will randomize the MAC address of the device when connecting to a new access point. The below figure displays an Android device running Android version 10, showing MAC address randomization enabled.

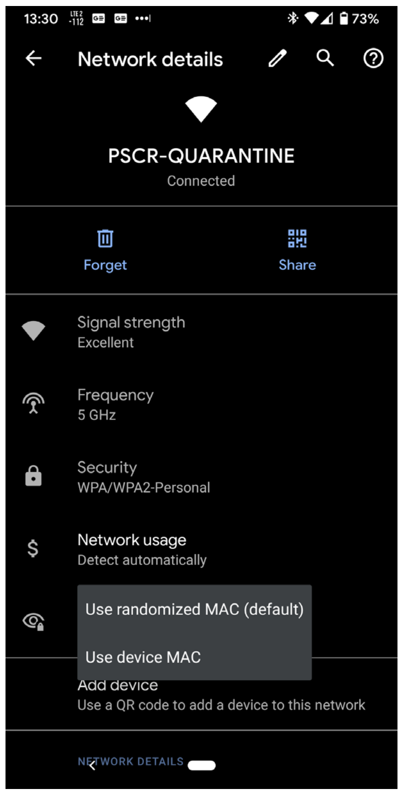


Figure 12 - Optional Mac address randomization setting

Figure 12 shows an Android device's Wi-Fi network settings where a randomized MAC address can be set under the specific Wi-Fi network. As shown in the figure, randomization is enabled by default.

Gaps: Network disruptions can occur due to MAC randomization. When a device is associated to a Wireless Basic Service Set (BSS) or Extended Service Set (ESS), changes in MAC address can temporarily disrupt service to the device. Depending on the network configuration and device implementation, it is possible to cause network disruptions, causing loss of device connectivity. For example, networks that use MAC addresses for network access control cannot support devices that utilize MAC address randomization.

Wi-Fi probe requests, device traffic patterns and frame sequence numbers from the mobile device may also be used to profile or fingerprint certain mobile devices, despite enabling MAC address randomization. MAC address randomization alone does not ensure device confidentiality due to advanced heuristic tracking methods.

Guidance: MAC address randomization should be enabled and used when possible. Network access control considerations should be given for devices that authenticate to enterprise wireless networks. The use of authentication methods that depend on static MAC addressing cannot be used. Additional device protections, as discussed in this document, are recommended in addition to MAC address randomization.

Only trusted Wi-Fi networks should be used while using a mission critical, first responder device. When outside of a trusted network, LTE broadband networks should be used.

Benefits: MAC address randomization ensures confidentiality by preventing the tracking of a device within or between networks. Similarly, randomized MAC address may prevent identification of the device hardware if the OUI portion of the address is randomized.

B.1.9 Test 9: Device Update Policy

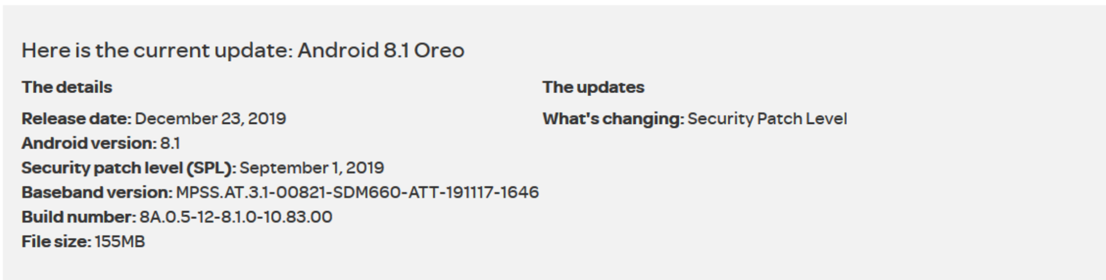
Security Objective(s): Device Ease of Management, Integrity and Healthy Ecosystem.

Test Description: Verifying the device update policy seeks to understand how often the device is scheduled to receive security updates and other software from the vendor. Specifically, the regularity / cadence, type, and reasons for updating the device and applying security patches are common policies contained in the update policy.

Test Outcome: Update procedures and implementation are clearly defined within device user guides, however specific information concerning frequency and scheduling of updates were not easily obtained. Both Android and Apple iOS have defined roadmaps for OS updates and releases at their respective web sites, but most mobile providers and smart phone vendors control the actual implementation and release of updates, patches and features. Since Apple iOS devices are sourced from a single vendor, roadmaps, release and patch notes can easily be found from the Apple support site. Specific versions can be found on the Apple web site and release notes have specific, clear sections for features that received updates. A specific section for privacy and security contained high level descriptions for specific security updates or features.

For Android devices, none of the vendor/platform specific user guides or web sites contained information concerning security update roadmaps. Some of the mobile device vendors have software update histories and change reports freely available, while others required support account logins to view update information. Overall, the information for security related updates are difficult to find for Android devices in vendor specific handsets. Vendor produced documentation does not include detailed information concerning security patches. More detailed information can be found through the Android support and developer web sites; however, the information only refers to the general Android OS and not the vendor specific, OEM version of the mission critical device.

Analysis: Specific device software and security patching roadmaps are not readability available. Device manufacturers did not contain specific information regarding patching but did contain update procedure documentation. The web site of cellular providers supporting the device contained the most recent information for device updates. Update information didn't contain road mapping information to address outstanding patch fixes for security vulnerabilities.


Here is the current update: Android 8.1 Oreo

The details

Release date: December 23, 2019
Android version: 8.1
Security patch level (SPL): September 1, 2019
Baseband version: MPSS.AT.3.1-00821-SDM660-ATT-191117-1646
Build number: 8A.0.5-12-8.1.0-10.83.00
File size: 155MB

The updates

What's changing: Security Patch Level

Figure 13 - Example update information

Most cellular service providers implement and control the distribution of software and security patch updates. This information can be found for specific devices on the cellular service provider's web site (see Figure 13).

Gaps: Update policies are either non-existent or not consistent among the Android devices tested. Update policies are difficult to find and often do not contain detailed information to make formal decisions.

Guidance: End users and administrators should configure devices to receive notifications when patches and updates are available. This configuration is commonly the default for both Android and Apple iOS devices but should be verified before initial deployment. Both Android and iOS devices are set to automatically check for updates and notify the user when updates are available. Users and administrators should be aware of the vendors current support for respective devices. Software versioning and patch levels can be found under the device's "About" menu on both iOS and Android devices. The specific version and patch level for a device can be cross referenced with on-line documentation to ensure the latest software is in use. As discussed in Test 4, OS versions and patch levels can be referenced in CVE databases to check existing vulnerabilities.

End users and administrators should also consider the schedule/timing of applying software updates. Applying a patch/update during an emergency incident can impact a First Responder's ability to perform their public safety activities. Device administrators should also ensure that all public safety applications are compatible with the software before performing an update. Lack of compatibility can prevent a First Responder from accessing public safety resources.

Benefits: A defined device update policy informs the user of ensured continuity of device support. It notifies the user of any potential vulnerabilities or enhancements made to the device OS. Applying patches assist in protecting a first responders' mobile device from known vulnerabilities.

B.1.10 Test 10: Rogue Base station Detection

Security Objective(s): Availability, Confidentiality, Integrity and Authentication

Test Description: Long-Term Evolution (LTE) is commonly known as 4G in the 3GPP specification. This test serves to identify the known LTE vulnerabilities and how public safety and first responder groups can protect against these attacks. Analysis will include settings that can be configured by first responders, conditions to observe during an LTE service attack and appropriate response actions.

There are three general attack methods that bad actors will use when targeting mobile devices utilizing LTE networks.

1. Denial of Service
2. Man-in-the-middle or rogue base station
3. Location Tracking

Denial of service attacks are the most successful because they can be performed multiple ways. Bad actors can “jam” the operating frequency, denying use of the mobile spectrum. Another way is to impersonate an LTE base station and send a fabricated network rejection message. Note that rogue basestations are also referred to as rogue eNodeBs or stingrays in some publications or articles.

Man-in-the-middle attacks involve both impersonating an eNodeB as well as causing a “downgrade attack.” In this method, the bad actor will send a rejection message, causing the mobile to disconnect from the trusted network as in the denial of service attack. Secondly, the bad actor will also run a 2G eNodeB that the mobile will believe is a valid service node. 2G services lack mutual authentication and weak encryption methods required in modern communications networks. Once the mobile device connects, the bad actor can intercept any and all traffic the user sends over the network.

Location tracking attacks utilize a weakness in how eNodeBs identify mobiles in each cell. In general, the information gathered from this attack cannot be detected by the user and is gathered by the bad actor using passive sniffing techniques.

Test Outcome: In the default configuration, mobile devices will attach to any “valid” eNodeB providing a mobile connection. The order of preference is to attach to the network providing the topmost tier connection within the provisioned “home” network. For example, if the mobile device’s provisioned network has an available 4G LTE signal, the phone will authenticate and connect to that network first. In the event of signal degradation or poor coverage, the handset will connect to the next best service tier. Fallback to 3G or 2G will occur when those services are available in absence of higher quality links and/or access to the mobile device’s “home” network. When a rogue eNodeB is introduced, the mobile handset will attach to the rogue base station in scenarios where legitimate services are lost or degraded to an unusable status. This will only occur if the rogue base station is configured to imitate an existing base station and to accept and authenticate with the handset.

Analysis: A tradeoff scenario occurs whilst determining greater protection versus reduce cell signal quality. Out of the box, most mobile devices are provisioned to connect to cellular services of any connection level, if available. This behavior is normal to ensure maximum coverage for cellular subscribers. Some mobile devices can be configured to only connect to specific quality connections, e.g. 5G, 4G, 3G, 2G or a combination of those services. Similarly, most devices allow the user to configure "home only" connections or disabling roaming when home networks are not available. All of the first responder specific mobile devices that were analyzed gave the user both the option to configure connection type as well as roaming options. However, many of the devices, not designed for first responder needs, only contained options for roaming configuration.

Gaps: Device types and OS may alter user configurable settings to control cellular connection parameters.

Most cellular vulnerabilities are inherent issues within the LTE standard and cannot be mitigated by the user. Ratifications within the 3GPP LTE standard would have to include methods to hide sensitive identifiers mobile providers use to authenticate and track handsets.

Some mitigations can only occur within the mobile provider network, including encryption of sensitive identifiers of mobile devices.

Guidance: Mobile providers should ensure baseline configurations of LTE network components include maximum security and encryption for public safety and first responder devices. Device users should be aware of the potential behaviors of LTE based attacks. Many of these attacks are localized, meaning the bad actor is specifically targeting a responder or group of responders with the intent of further mal intent. While targeted campaigns on mobile devices are rare, special events or circumstances may make an LTE based attack a viable method.

Denial of Service mitigations – Users should observe behaviors in signal drops and outages. A fabricated *Attach Reject* message from a rogue eNodeB causes a mobile device to go into an out-of-service state. *Attach Reject* messages are temporary blocks that can be removed by rebooting the mobile device or toggling off and on Airplane mode. The only way a first responder may know they have been affected by an *Attach Reject* attack is the loss of signal, "no bars" or inability to use network services. Another type of denial of service attack is using signal spectrum jamming. Jamming attacks can only be mitigated by moving into an area not affected by the jam or using alternative signaling channels. Localized controls, such as deployable LTE eNodeBs, may also counteract weaker jamming signals. Alternative protocols, such as LTE over Wi-Fi, or IMS over Wi-Fi can also be utilized if cellular service is unavailable.

Man-in-the-middle or rogue base station mitigations – like denial of service, observations in signal dropping and outages are inherent to these attacks. Users may also observe a downgrade in service from 4G/3G to 2G GSM. If the downgrade of service occurs in an area where 4G LTE service is inherent, this may be indicative of a downgrade attack. Users can mitigate these attacks by configuring the device to only attach to 4G LTE networks. However, the drawback is that coverage may be limited in areas where legitimate services are available. Configuring the device in 4G LTE only mode will prevent the device from connecting to mobile services in poor reception or coverage areas.

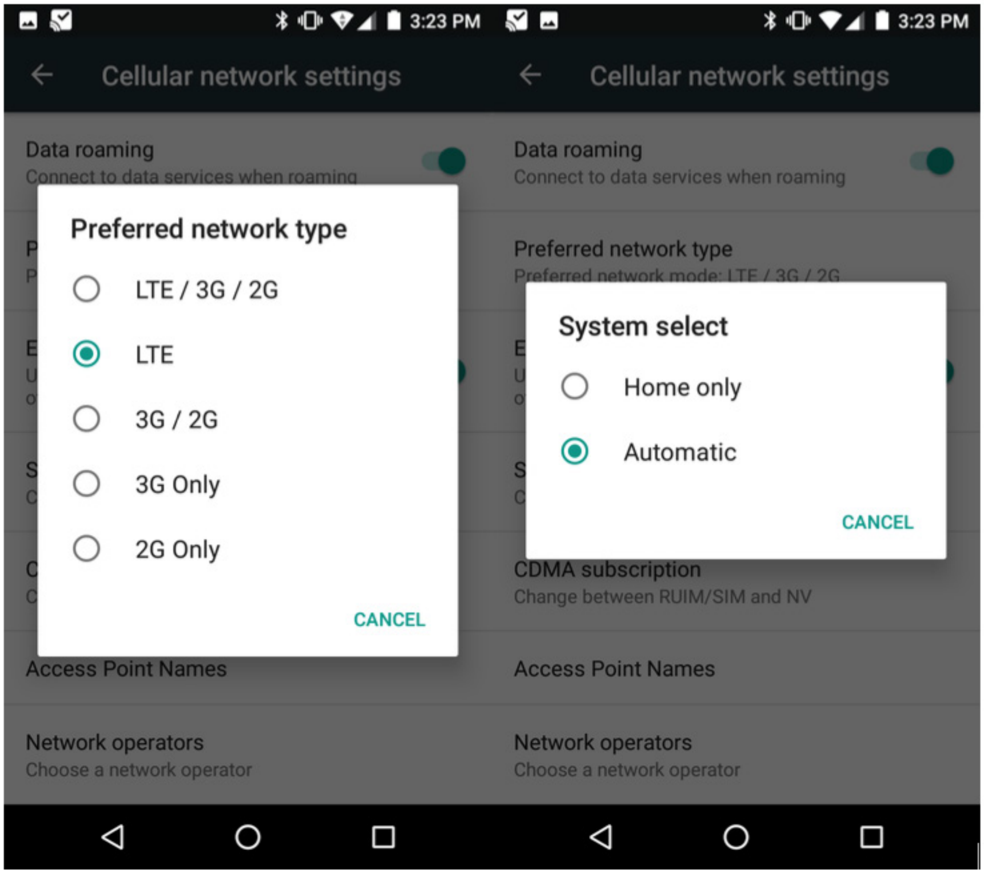


Figure 14 - Preferred network selection on an Android device

The preferred network can be configured to LTE only mode on some mobile devices (see Figure 14 - Preferred network selection on an Android device). Pictured on the right, configuration can set the mobile device to only connect to the home subscriber network. The home subscriber setting ensures the device only connects to a NPSBN. Be aware that both settings will effectively limit coverage for the device. These settings should only be used in situations where increased security is necessitated over mobile coverage requirements.

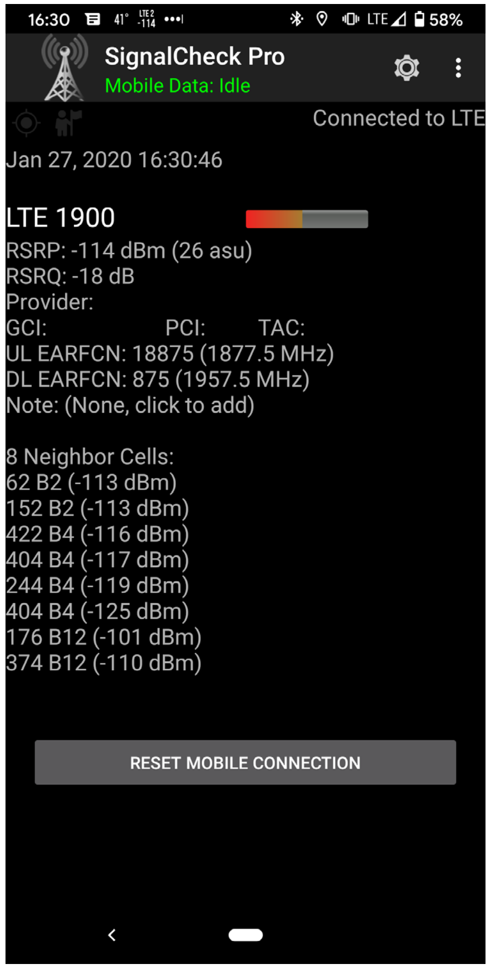


Figure 15 - Mobile network connection monitor

3rd Party applications, such as SignalCheck in Figure 15, can be used to monitor connected LTE networks. Savvy users and administrators may utilize these utilities to determine signal quality and legitimate LTE connections in special operations scenarios.

Location Tracking mitigations – Bad actors can utilize both passive monitoring and the man in the middle methods to track LTE users. First Responders should use the guidance for mitigating man in the middle attacks. However, since passive monitoring cannot be mitigated by the user, service providers should ensure that mission critical networks contain provisioning to prevent tracking of local mobile identifiers, such as international mobile subscriber identities (IMSI) or Cell Random Network Temporary Identifiers (C-RNTI.) These identifiers should be transmitted via encrypted methods to ensure passive monitoring attacks are mitigated.

Benefits: First Responders should have a general situational awareness of LTE mobile devices. While LTE based attacks are unlikely, they may be used in specific circumstances where the bad actor is savvy with communication technologies. Such circumstances may include investigative cases, SWAT scenarios or coordinated campaigns.

B.1.11 Test 11: Configuration Guidance

Security Objective(s): Integrity, Device & Ecosystem Health, Interoperability

Test Description: Mobile device configuration guidance provides the user instruction to configuring the device, ensuring integrity, device ecosystem health and interoperability. This test will review the type of guidance provided from the vendor to the public safety professionals. Analysis will determine if any of the contained information contains security guidance dedicated to properly owning, operating, and configuring the device for public safety use. The procedure of this test utilizes the outcome observed in Test 1; however, this test focuses specifically on user guidance after device unboxing and post-provisioning.

Test Outcome: Devices have specific user guidance in the user manual to secure the mobile device. Configuration settings include enabling/disabling of location tracking, account settings, user accounts, unlock settings and linked accounts. Detailed user guides can also be found online from both the device manufacturer and the cellular service provider.

Analysis: Out-of-the-box devices will go through a setup procedure to secure settings such as location tracking, encryption and lock screen settings. Application specific settings are configured after the device is initialized and in some cases after applications are installed. Configuration guidance is easily obtained through the device manufacturer's web site, accompanying documentation, and the cellular provider's web site. The most accurate guidance information is contained on the cellular service provider's web site for Android devices. Guidance for Apple iOS devices is best obtained through Apple's support web site. Specific app settings must be obtained through the application's vendor or developer web site. MDM solutions and local settings are also available for further device controls, such as camera access and app store access.

Gaps: OS updates and patches may alter the location of specific settings. Likewise, updates and patches can alter previously set configuration and/or add additional settings. Deviations from update and patches may require the user to either find new settings or search online for additional settings. MDM software can help mitigate settings induced risk among devices that are under common administration. App specific settings are variable, and users must refer to the specific app vendor for configuration guidance.

Guidance: It is recommended to perform post provisioning of devices, especially after installation of additional mission critical applications. Only the minimum services and permissions should be enabled to allow functionality of mission critical applications and perform routine duties. Configurations, such as location tracking should be turned off for non-essential applications, including OS provided tracking services. Application permissions are configured upon installation or can be changed post-installation in the settings menus.

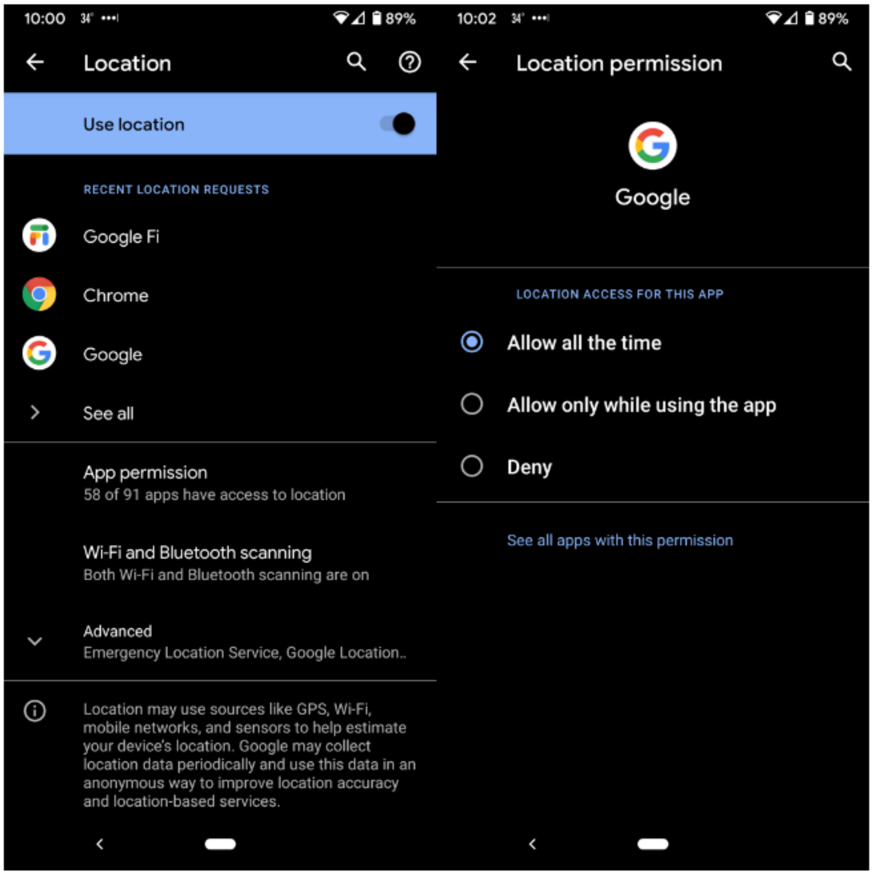


Figure 16 - Android device location permissions (1)

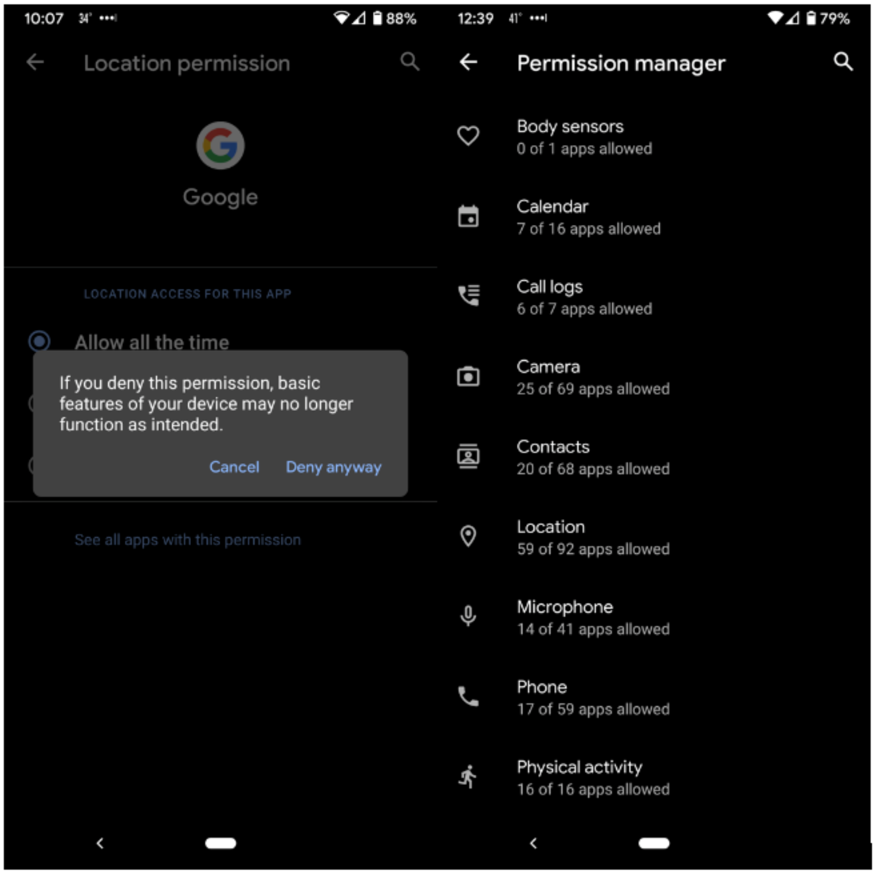


Figure 17 - Android device location permissions (2)

Android contains specific provisioning for location and permissions for each installed app. Figure 16 displays a system wide setting for location tracking as well as a log of recent tracking requests. The right image of Figure 16 shows specific settings for an individual application. Figure 17 shows a warning message notifying the user that disabling location services for certain apps may negatively affect basic device functionality and permissive variables for device functionality.

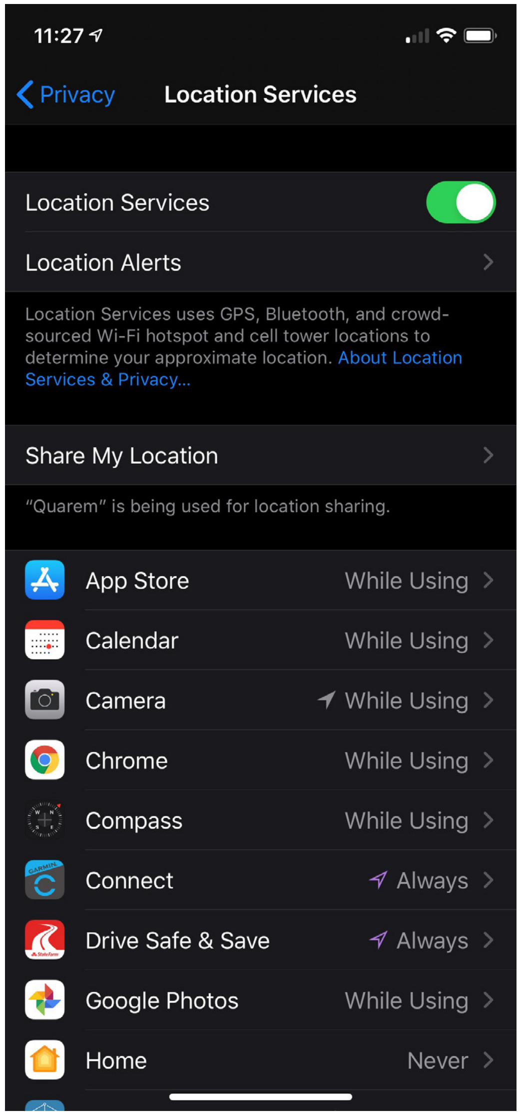


Figure 18 – iOS device location permissions

Figure 18 shows how Apple iOS devices contain a similar menu to control location permissions for the entire device or individual apps.

Mobile devices allow for application specific settings for various permission. Note that some permissions must be enabled for the device to operate properly. The application will typically re-prompt the user if an application requires additional permissions. Users and administrators should regularly review device permissions and services to ensure device integrity and prevent profile tracking of responders.

Since settings are subject to change with OS versions and device types, it is recommended to utilize web-based resources for configuration guidance for specific devices. Most mobile OSs provide detailed lists of apps and associated permissions as shown in the Android Permissions Manager in the figure above. It is recommended to regularly test applications, especially after updates or permission changes, to ensure that first responder applications remain operational. Policies applied through an MDM solution should be regularly tested to ensure proper policy implementation as well as adequate operation of the responder devices. Negligence in performing regression testing of security polices and operational functionality puts the first responder at risk. For example, a security policy that limits the use of the device's camera may impact the ability to collect incident evidence at a crime scene. In some reported cases, public safety personnel have resorted to use non-secure, personal devices to collect such evidence. These actions prevent the responder from completing their job, exposes their personal asset to external risk and may invalidate the evidence and chain-of-custody processes.

Benefits: Post provisioning of device security settings ensure device integrity by securing device permissions. Location services can allow profiling through apps and tracking of First Responder devices. Linked accounts may provide app access to mobile settings, cameras, haptic devices and databases. Linked accounts may present the potential for remote application execution or device exploitation through the installation of backdoor trojans or solicitation exploitation. Users should be aware of configuration and security settings to ensure continued health of the mobile device in post-provisioning situations. Post-provisioning, post-policy application regression testing should be performed on test devices before being applied to first responder devices in the field. Field users should be notified of changes and updates so that devices can be operationally verified in a non-emergency setting.

B.1.12 Test 12: Wi-Fi MitM and Rogue AP Detection

Security Objective(s): Integrity, Confidentiality

Test Description: This test checks to see if the mobile device can locally detect Evil Access Points and/or Man-in-the-Middle (MitM) attacks when using Wi-Fi.

> *Note:* While additional, advanced MitM attack methodologies exist, this test intends to test basic mobile device MitM detection using built-in OS defenses.

Test Procedure: The test configuration network consists of two Access Points (see Figure below.) One AP is the trusted Infrastructure AP utilizing secure methods of authentication and encryption. The second AP is the EvilAP used to mimic the Infrastructure APs SSID. This test consists of two parts. Part one tests if the Smartphone Device will connect to the EvilAP, part two tests interception of HTTP/HTTPS traffic and extraction of private data. For the tests to be "successful" the smartphone device must be able to locally distinguish between the trusted and untrusted Wi-Fi connections. Differentiation of trusted/untrusted connections are accomplished through association via a trusted 48-bit BSSID. If the first test is not successful, naturally the second MitM test cannot be tested. In a non-successful event, the second condition is tested by connecting the Smartphone Device to the EvilAP and the MitM test is performed.

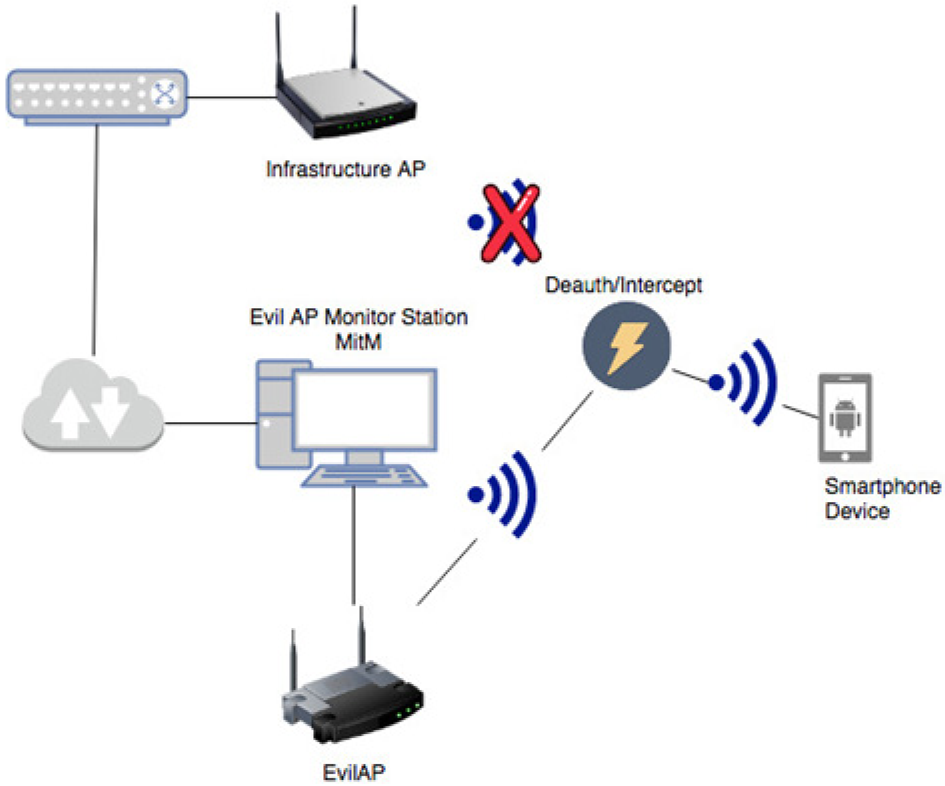


Figure 19 - EvilAP/MitM network configuration

Test Outcome: All DUTs successfully mitigated the Wi-Fi spoofing attack as well as the MitM attack. The mobile wireless client distinguishes the Wi-Fi connections by BSSID, even if the SSID contains the same network identifier. Mobile devices will not automatically connect to the rogue AP until manually subjected via user input. Additionally, if previous association is made to both APs, the mobile client would prefer the Infrastructure AP using advanced Wi-Fi security mechanisms over an AP using Open or no authentication.

All devices successfully mitigated the T attack. The devices tested claimed to be connected to the Rogue Wi-Fi network but reported "no internet." This factor indicates that the Wi-Fi client identified an untrusted connection. Further analysis with the mobile's web browser identified that the trusted destination web sited utilized a secure mechanism called HTTP Strict Transport Security (HSTS). HSTS prevents SSL downgrade attacks.

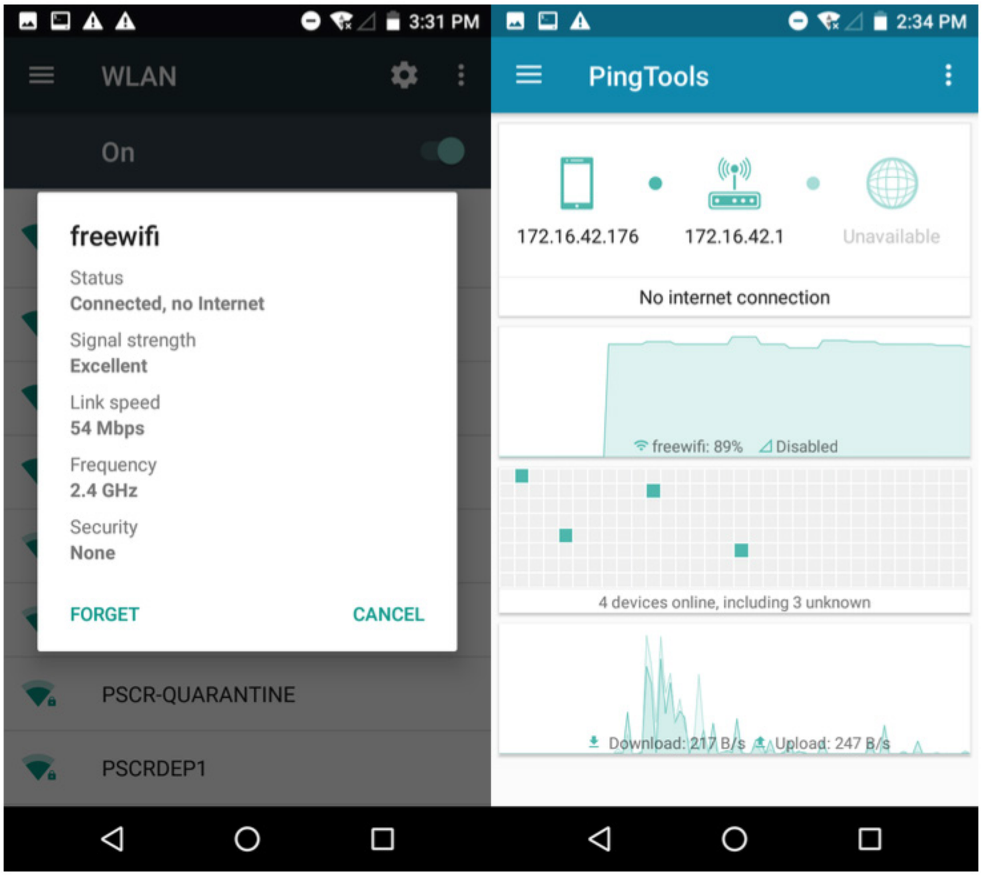


Figure 20 - Mobile device connection to AP with no Internet

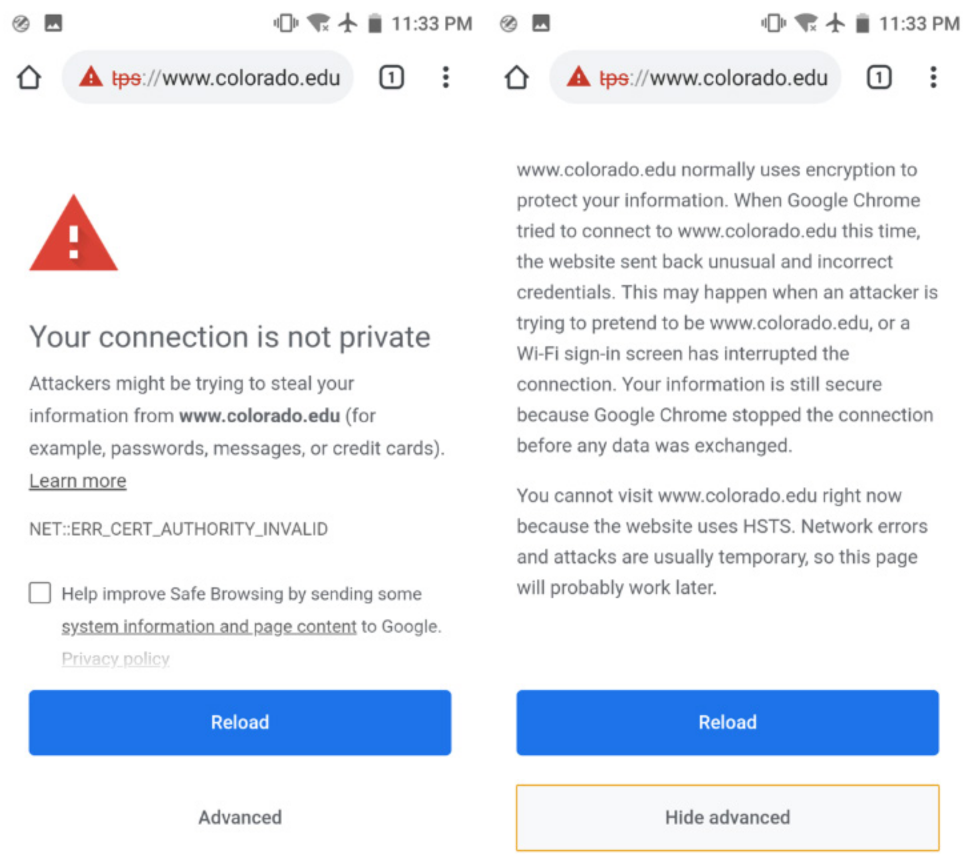


Figure 21 - Website detects MitM attack due invalid certificate response

Figure 20 (left) displays an Android Wi-Fi client that shows connection to AP, but no internet. On the right of Figure 20, Ping Tools (3rd party app) is shown to verify the connectivity status. Figure 21 shows a browser request detect MitM attack due to invalid certificate response and the advanced information explaining why connection was not established due to invalid certificate response.

Analysis: Mobile devices have built in mitigations to prevent Wi-Fi based attacks, both on the OS level as well as the browser level. Many indicators and warning messages are conveyed to the user to make them aware of a potential attack.

Gaps: HSTS is a server-side protocol feature that must be implemented in both the web server as well as the mobile browser.

The web browser is not locally tied to the OS, instead the OEM web browser was used in this experiment. Changes in browser technologies and protocols are typically interdependent of the OS. Therefore, it is important to keep browser applications up to date with latest revisions and patches in addition to the mobile OS.

We were unable to prevent the mobile device from connecting to the fake AP. This requires additional network configuration from a network and mobile device administrator.

Guidance: The device user should always check the network connection and access to network services. Awareness of network connectivity and availability is important to validate the Wi-Fi/LTE connection to ensure connection to the proper network.

To prevent connection to rogue or public access points, a device administrator should consider leveraging the VPN services on the mobile device. The device user should authenticate to the VPN services to ensure authorized access to public safety resources. VPNs ensure data confidentiality, especially when connecting to public Wi-Fi access points or other non-trusted networks.

Benefits: Detection mechanisms implemented in the mobile device's Wi-Fi client prevent basic MitM attacks by distinguishing trusted/untrusted connections. If a user accidently connects to an untrusted access point using the same SSID, multiple indicators are present to alert the user of a potential attack.

Configuring a mobile device to connect over first responder VPN services allows the device owner control over network access and secure transfer of public safety information. User data is encrypted and cannot be interpreted by any intermediate entities.

B.1.13 Test 13: Boot Integrity

Security Objective(s): Integrity

Test Description: This test will check to see if the mobile device is performing some form of boot validation. Boot validation are integrity checks on device boot files and processes to verify that the mobile OS has successfully executed into a valid state. Boot validation methods on mobile devices require executable kernels and code to be verified via digitally signed cryptographic hashes (of the kernel code). The exact location of the hashes varies between devices, but the operation and methodology are similar in all mobile devices. After the boot executable code is loaded into memory, validation occurs. If validation succeeds, the device will continue to load system executables and may perform additional validation. If validation fails, the device will stop the boot sequence, enter an error state and/or reboot.

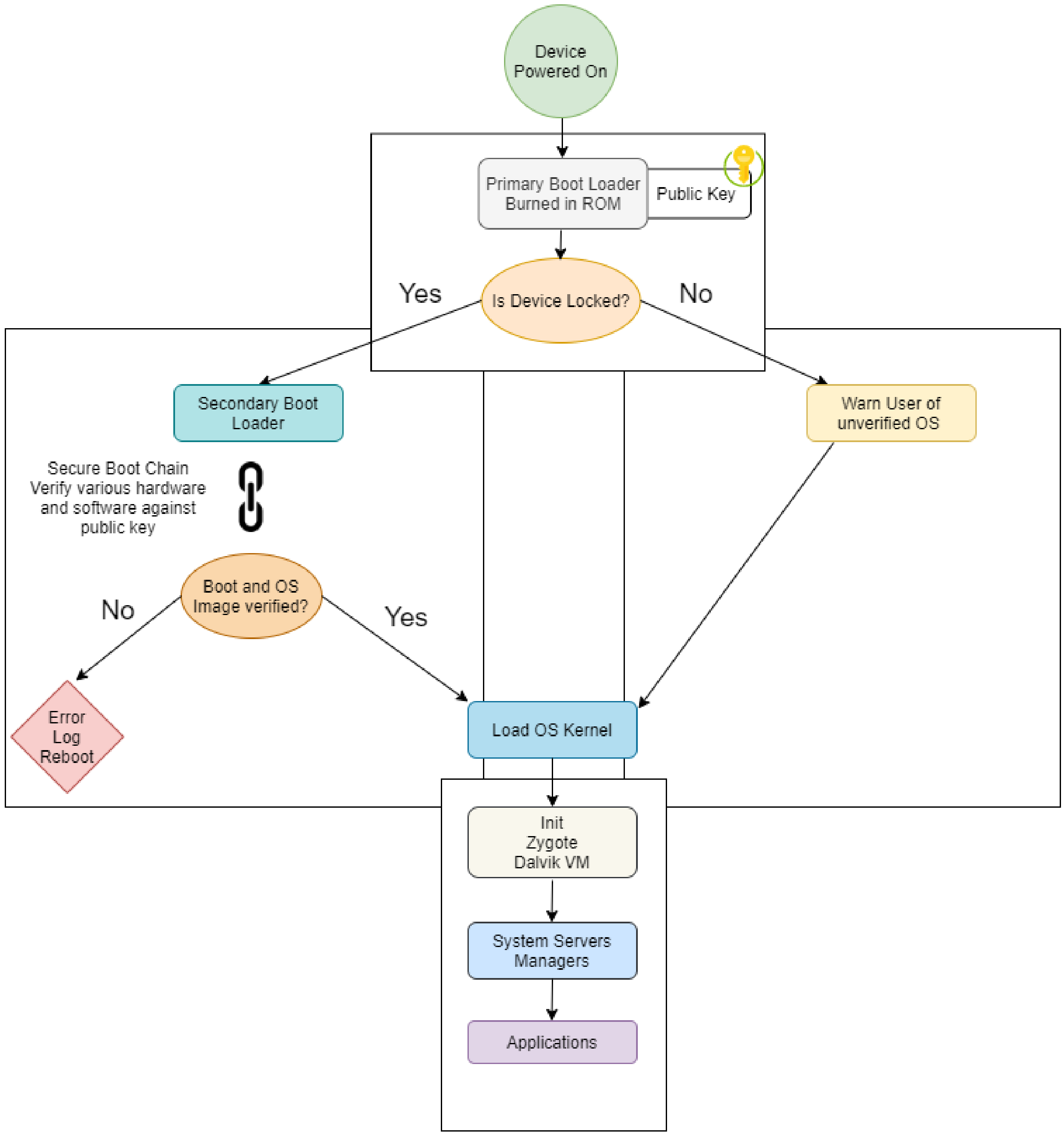


Figure 22 - Simplified schematic of the Android boot process

Secure boot operating systems utilize cryptographic public keys that are burned into system read-only memory (ROM) from the factory. The boot processes will use a burned in public key to verify hashes of boot loaders, hardware components, system images and the OS image in a "boot-chain". This methodology allows lower levels of the boot operation to verify the next operation in a "chain" of events. If any step in the chain verification fails, the device will stop the boot process, log the error, notify the user and reboot the device. While the boot procedure is like that of any other computer, verification occurs before any code is loaded into system memory or storage. Factory unlocked mobiles will bypass the secure chain verification, warn the user of booting an unverified OS, and load the OS.

When selecting mobile technology, the consumer needs to be aware of the differences and selections available between "factory unlocked" and "locked" phones. Starting in Android version 4.4, methods were added for kernel verification during boot and notified the user if deviations occurred. In Android version 7.0 boot verification was enforced to prevent data corruption and malicious compromise. Subsequent Android releases beyond 7.0 perform boot verification and in some cases have improved these methods to address known exploits or improve boot security methods. Apple iOS devices also cryptographically sign components involved in the booting and startup process in a similar method as Android. iOS boot code is immutable at the chip fabrication level and verified through Apple Root CA verification.[15]

Test Outcome: All tested devices contained some degree of boot verification. One of the tested devices contained the oldest Android version 4.4, however still contained kernel verification, but could be easily bypassed. Another device contained a special version of Android OS and therefore did not have specific information about boot integrity. Since this device also came factory unlocked, boot integrity methods can be bypassed by the user. All the remaining devices in the test contained an Android version greater than 7, contained enforced boot verification methods.

Analysis: Modern mobile devices contain some form of boot integrity verification. Like any technology, older devices may not have the latest protection mechanisms and are more likely to contain exploits to bypass boot verification. Newer devices also contain hardware level verification methods that check for digital signatures and cross reference these signatures with trusted manufacturer sources. Overall, factory "locked" devices provide the greatest boot integrity protections and should always be considered over "unlocked" devices.

Gaps: Many older handsets cannot be software upgraded to protect against new exploits. Like any other secure computing device, bootloaders typically run immutable code on read-only memory implemented at the factory. Future technologies and exploits may reveal weaknesses in current cryptographic algorithms. Since cryptographic keys are burned-in, they cannot be updated to support newer crypto algorithms that provide greater entropy. Typically, it is assumed that the lifecycle of the device is shorter than technological advances that may be used to exploit security controls.

Guidance: First responders and public safety organizations should only purchase mobile devices from trusted vendors. Devices should be factory locked to ensure device integrity and that only the mobile provider or device vendor can perform OS updates. Devices that are no longer software upgradable or hardware cannot support the latest boot integrity methods should be retired out-of-service.

Benefits: Boot integrity prevents loading of an unauthorized OS that could be used to compromise handset devices, potentially leading to data extraction or utilization as a remote attack platform. In Android Verified Boot Version 2.0, system prompts are implemented to warn the user in the event a custom or unverified OS is loaded. This warning occurs on both factory locked or unlocked Android devices. Apple iOS devices also provide similar protection mechanisms to prevent loading of unauthorized iOS boot code.

B.1.14 Test 14: Data Isolation

Objective: Isolation

Description: This test will check to understand if the mobile device is utilizing an isolation technology such as Android Security-Enhanced Linux (SELinux). Data isolation occurs on individual applications after the device is fully booted and operational. SELinux enforces access control over all device processes as well as their interaction with crucial Linux process, such as init, dmesg, cron and others. Data isolation provides device protection by confining and restricting system services and controls access between applications. These protections create sandboxes that allow applications to run within its own domain without risk of interfering with other applications or system services. Many mobile device systems run data isolation on a allowlisted basis where processes are denied unless explicitly allowed. However, for development purposes, it is possible to enable special modes that are more permissive. Permissive modes are disabled by default and must be manually enabled by the user or developer. While permissive modes allow greater access to system resources and processes, enabling this mode puts the device at greater risk. However, most modern mobile operating systems, such as Android, still allow sandboxing even while in permissive test modes. Android OS introduced SELinux sandboxing into its operating system in version 4.3. Version 7.0 and 8.0 added features to further restrict applications to sandboxes as well as boot level isolation for vendor specific images. Apple iOS uses a similar data integrity suite called System Integrity Protection (SIP) or rootless integrity protection. Much like SELinux, a combination of file system permissions as well as sandbox environments separate applications in user spaces to prevent unwanted system compromise. Accordingly, Apple further enhances application security by requiring code to be vetted through a digital signing process. Apple iOS also includes a specific development environment to allow unsigned applications, not yet vetted though the Apple App Store. Like Android, development environments include enhanced protections and sandboxing to prevent system compromise.[16]

Test Outcome: All observed devices contained a form of data isolation for applications. Most of the devices were factory locked and developer options were disabled by default. Of the devices that were not factory locked, developer options were disabled, and OEM OS images were used in testing. All devices ran in the respective enforced security policy to provide sandboxing of applications and file system protections.

Analysis: Data isolation methods are implemented on most modern devices. Like Boot Integrity methods, older hardware and software may not support the latest protections provided by data isolation methods like SELinux or Apple iOS SIP. Data isolation methods can be bypassed though user modification, however sandboxing of applications creates permissive restrictions for processes and applications. Most users are unaware of data isolation since there is an abstraction level between app operation and user interface (UI). Options for the user to interact with data isolation mechanism must be explicitly implemented by the application developer or through system settings.

Gaps: No vendor guidance is given regarding data isolation in the user documentation or web site resources from the vendor. Data isolation is considered a mandatory or common implementation on modern mobile devices, so it's often assumed that these features are enabled by default. Typical users would have no relocation of data integrity unless explicitly notified of its purpose or in the event of compromise.

Data isolation does not prevent administrative override to grant user or app permission to system resources. Out-of-the-box, the device owner has complete administrative control over the device to grant application permissions, which could potentially compromise the data integrity of the device. It is important to understand that data integrity does not influence administrative control, these two concepts are not analogous.

Guidance: Most modern handsets and mobile devices contain the latest features and enhancements regarding data integrity protections. Similarly, devices typically have data integrity mechanisms built in and enabled by default, requiring little or no user intervention. Older devices may lack features to protect against modern attacks, therefore it is important to keep devices up to date with latest OS patches and upgrades. Devices that are no longer supported by the hardware vendor or OS manufacturer should be retired out of service.

To guarantee data integrity, applications should only be downloaded though the OS app store. Apps must be digitally signed to ensure the contained code has been properly vetted for public use.

Users that install new applications from the app store should take note of any special permissions required for the application to run. Allowing application permissions grant use of protected system processes, which could compromise data integrity and put the system or user data at risk. Only applications required to perform first responder duties should be installed to mission critical handsets. By default, out-of-the-box, the device owner is considered the device administrator and can install apps or make system changes. While data integrity mechanisms are always in effect, the user can grant permissions to applications to bypass or allowlist access to system processes. Device administrators may consider using an application vetting service or working with an application provider that includes the information necessary to address any concerns (app permissions, data collection, privacy concerns, etc.). [21]

Devices that are under common administration should run supplemental device enrollment software to further enforce data integrity policies at the enterprise level. Device enrollment management systems are typically used to secure and manage enterprise mobile devices. These systems enforce device policies to ensure devices are up to date and prevent installation of unwanted or unnecessary applications. Device enrollment systems and software are not included in most factory handset configurations.

Handsets not used in software development environments should have developer and test modes disabled. This setting is commonly found within the device's setting menu, but may be hidden from the user, depending on the platform and OS version. By default, most factory distributions have developer or test modes disabled. This setting is typically not included within the normal user documentation but can be found though online web searches or vendor support web pages. Depending on the hardware platform, development environments may only be accessible using supplemental hardware interfaces and software development kits. Devices used for development purposes should not be used daily first responder use.

Application developers should only use software development kits offered from the OS developer. Applications should be vetted through the manufacturer and digitally signed for end user use and distribution. Any developed application should only request permissions necessary for the application to function. Requested permissions should be clearly explained as to why the permission is required within the app's description on the application store. During installation or application use the user should be prompted to allow special permissions. Allowing excessive or unnecessary permissions can allow an application to bypass data integrity protections, putting the device at risk.

Benefits: Data integrity protects OS processes and user data from potential compromise by enforcing access permission. Data integrity protection mechanisms are a combination of supervisory processes that prevent execution of code, access to system processes and critical OS file system areas. These supervisory processes prevent the deletion or alteration of critical system files, enforce user process separation, segregate application processes, and enforce application permission to system functions.

B.1.15 Test 15: Device Encryption

Objective: Confidentiality, Ease of Management

Description: This analyzes if the device is locally utilizing device-wide encryption, and how difficult it is to use. Device encryption encodes all user data using symmetric encryption keys. Once encrypted, the user must provide credentials upon boot to decrypt user data. Typically, the user only must provide credentials once and further encryption/decryption occurs automatically upon disk read and writes. Modern devices typically utilize dedicated, chip-based, encryption engines to support real-time processing as well as hardware level separation to physically separate encryption operations from systems processes. Physical separation of encryption activities creates isolated environments on-hardware to prevent compromise of encryption keys.

Two types of encryption are available for most mobile devices, depending on the mobile OS and hardware support. Device functionality behaves differently depending on the type of encryption used. One is not necessarily better or worse than the other regarding file system security but may alter the user experience. The type of encryption on mobile device is hardware dependent and typically not configurable by the user. For more information about Android encryption refer to Android's developer web documentation. [13]

- File-based encryption only encrypts user files, which allows for partial phone functionality before decryption. File-based encryption allows for the device to receive calls and/or make emergency calls before credentials are entered. Multiple keys can be used to provide independent encryption of files, which is useful in multi-user configurations or in high-confidential scenarios where additional protections are required.

- Full-disk encryption uses only a single key to protect the entire volume of the device. User data as well as system data is encrypted and can only be unlocked at boot. The device is not usable until the key is unlocked.

Test Outcome: All of the DUTs supported file-based encryption. Encryption options were prompted upon initial device setup, however configuration for encryption was present in the device's security settings.

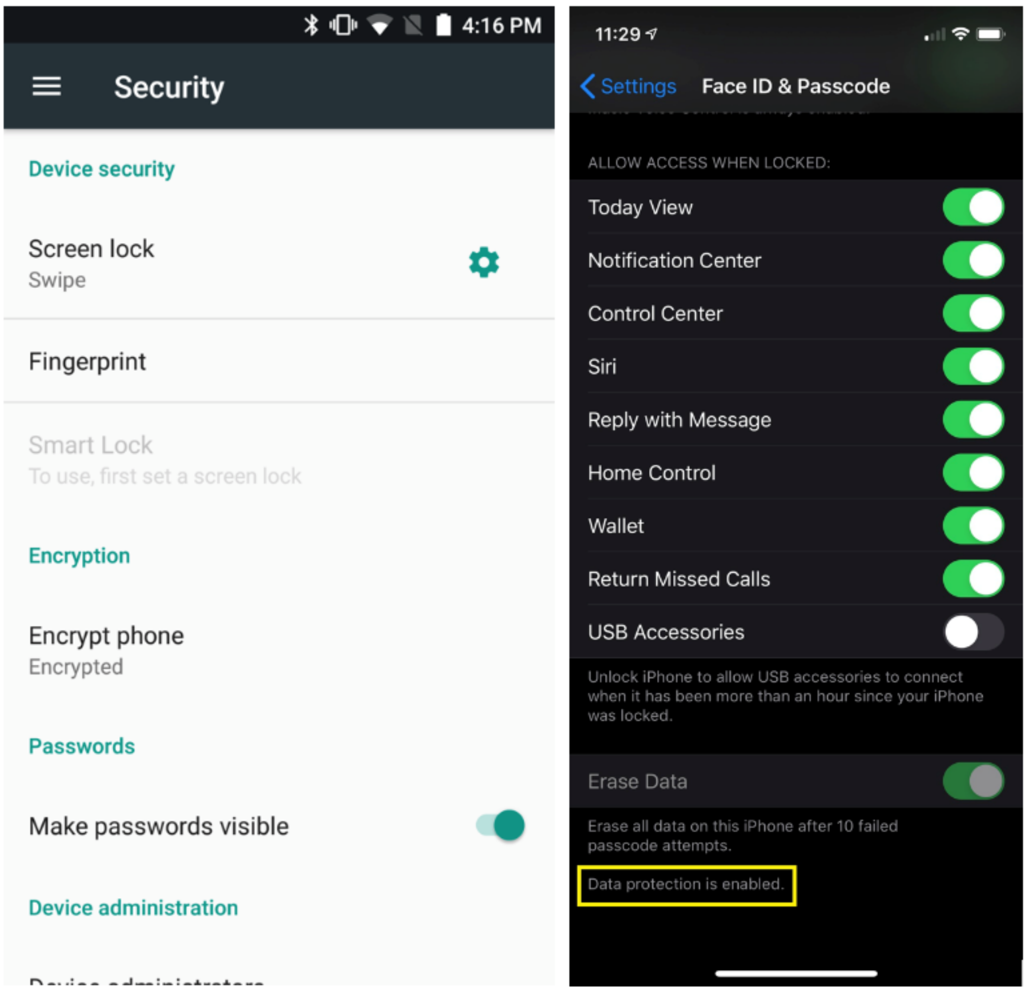


Figure 23 - (Left) Android device encryption settings. (Right) Apple iOS device data protection settings

Figure 23 shows an Android device's security settings confirming encryption and an Apple iOS device confirming encryption settings "Data protection is enabled." Neither device specifies what type of encryption is being used.

Analysis: All modern mobile handsets contain some form of device encryption. Apple iOS introduced forms of encryption and digital signing in early versions of its operating system. Digital signing of applications was mimicked after app store implementations were introduced in iPod devices. Encryption was introduced in iOS version 4, such as encryption on lock screen and application specific data protection. Android introduced encryption in Android version 4.4. Modern mobile devices include encryption as an initial deployment option and is recommended to the user on initial setup. Encryption is easy to set up, however it requires that the user implement stronger authentication methods. Stronger authentication ensures that encryption cannot be bypassed through brute force.

Gaps: No observable gaps were found concerning data and device encryption. Vendor guidance provided clear configuration instructions, where possible. Since encryption is offered during device setup, it is easily user configurable. On-line resources through the vendor or OS manufacturer offered clear instructions on how to set up encryption or where to check status of the device's encryption. App based encryption and configuration varies according to the app developer, this is not considered a notable gap for the device.

Guidance: Out-of-the-box most devices are not encrypted, however setup wizards provide the option to encrypt the device. It is recommended to enable encryption whenever possible, both on the OS/device level as well as within applications, wherever available. Device encryption can be enabled though the setup menu of the device, typically under the security configuration section. On Apple iOS devices, encryption configuration can be found under Settings, Touch ID & Passcode or Face ID & Passcode. When the device is encrypted, it will prompt the user for a passcode. It is important to recognize that this passcode is a separate passcode/key than the device "unlock" code. While these two passcodes can be the same or different, one will unencrypt the disk data, while the other allows access to the device's UI.

Disk encryption is only as good as the authentication methodology for access control. When possible, complex passwords should be used for encryption. It is important to remember that encryption passwords are generally only authenticated upon device start or bootup. This password should include complex alphanumeric passwords instead of the numeric pin. Passwords should contain special characters, both lower and capital letters, numbers and should not contain dictionary based, easily guessable words. Since digital identity guidelines change on a constant basis, it is recommended to use the latest NIST guidelines found at https://www.nist.gov. After the device is fully booted and decrypted, alternative authentication methods can be used to "unlock" the device screen during normal use. For public safety applications, users need to ensure that the device is fully booted and authenticated to ensure rapid access to the device is available.

On devices that support file-based encryption, applications can be "made aware" of encryption. Apps that require additional protections can utilize this feature by operating in separate protected disk space. When the protected app is started, it will prompt for a passcode to unencrypt app specific device data. This passcode is a separate key from the key used to encrypt user files but utilizes the same hardware level processing. Configuration of encryption for individual apps vary by app vendor and support for app-based encryption must be implemented by the app vendor. App based encryption is recommended where additional protections are required for app specific data. Examples include enterprise secret data, personal identifiable information or state secret data. Common first responder applications that utilize these mechanisms include enterprise email apps, document editors, forensic collections apps, and health monitoring collections apps.

Benefits: Data Encryption ensures confidentiality of user or system data if the device is physically compromised. If the device is lost or stolen, data on the device cannot be retrieved unless the proper passcode or key is presented to unencrypt the data. While the device may be reused, the data cannot be retrieved due to the data being encoded. If key passcodes are lost, data cannot be retrieved, and the device must either be factory defaulted or application reinstalled. Data encryption can also protect app specific data from other potential malicious apps on devices that support file-based encryption. Malicious apps and bad actors cannot access app specific encrypted data unless a key is presented to unlock data.

B.2 Wearable Devices

B.2.1 Test 1: Obtain General Hardware Information

Security Objective: Ease of Management

Test Description: This test will identify information about the device, and how easy it is to obtain that information.

Test Procedures: Search for online datasheets and technical documentation for each wearable device to obtain available hardware information and operating specifications. Most information was obtained using the device manufacturer's webpages and search engines if the information could not be found through the device manufacturer.

Test Outcome: All devices had specific online resources pertaining to the hardware and software specifications of each device. Some devices had specific datasheets that listed all the hardware components and manufacturer information while others listed the ranges of operating conditions that the device would be able to handle. Overall the information gathered about each device was sufficient to understand what sensors and components the device had as well as its hardware capabilities.

Analysis: Most of the information about devices was readily available. The information sheets varied in the amount of detail and types of data provided. The data ranged from specifications on the hardware and software to general marketing information about the product. Devices that were accompanied by technical datasheets could be more thoroughly examined since they often included important information about software versions and hardware components that may have been difficult to obtain through other means, since most wearable devices do not have an operating system to interact with.

Gaps: Some devices had more descriptive datasheets than others, so we were not able to get all the important information we would have liked to have about each device through reading these datasheets.

Guidance: Public Safety device administrators should have the device hardware information for asset management and resource awareness. Device manufacturers should ensure hardware information is readily available on the device, online, or in the device manual.

Benefits: Hardware data sheets allow public safety device administrators to be aware of the device information, such as the make and model. This information is important for general awareness, auditing inventory, and asset management. This information is also useful if any issues are identified with a specific make or model of device (e.g., recall or identify information about the device based on hardware datasheets that can give awareness to information (e.g., the device make and model).

B.2.2 Test 2: Obtain General Software Information

Security Objective: Ease of Management

Test Description: This test will identify the name and software version of operating system and major applications that are shipped with the device. Note that this is much more difficult on a wearable device than on a mobile device, and NIST engineers will not be performing firmware and binary extraction activities. This will also attempt to understand the protocol versions for the primary wireless protocols (i.e., Wi-Fi, Bluetooth, and Cellular). This test will also investigate the use of wearable specific protocols such as Near field communications (NFC), ZigBee, and Z-Wave.

Test Procedures: Software information about each wearable device was obtained using the device datasheets obtained from the device manufacturer or through packet captures. More recent versions of Bluetooth carry more comprehensive security capabilities, so identifying the version of Bluetooth used by the device is indicative of what security measures the device is capable of supporting. Some devices had the version of Bluetooth and Wi-Fi being used listed in their technical documentation. Other devices did not have this information readily available, so the information needed to be obtained through examining a packet capture for an attempted connection to the device using Bluetooth. Versions of Bluetooth past version 4.0 usually contain a packet that identifies the version of Bluetooth that the device is using even if a successful connection to the device cannot be made.

Test Outcome: All devices examined either used Bluetooth or Wi-Fi, with some devices using both for different purposes. The versions of Bluetooth being used by each device varied since Bluetooth is designed to be backwards compatible with earlier versions. All devices using Bluetooth exclusively used at least Bluetooth version 2.1 which was the first version of Bluetooth to enforce using encrypted key exchange between devices.

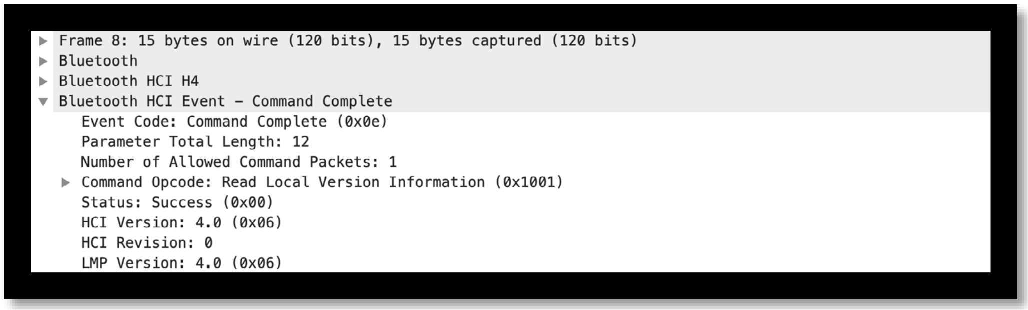


Figure 24 - Example packet capture used to identify Bluetooth version

Analysis: Most of the wearable devices examined do not contain an operating system since they were not designed to be interacted with directly. Therefore, to identify versions of Bluetooth being used you need to examine datasheets that accompany the device or identify the information through attempting to pair with the device. From examining device pairings, we could find the Bluetooth version directly if the exchange contained a 'Read Remote Version Information' packet sent by the controller or a 'Read Local Version Information' packet sent by the host. Both of these packets contain a "LMP version number" field that corresponds to the Link Manager Protocol (LMP) Version Number. This version number has a corresponding mapping to what version of Bluetooth is being used by the device. If the device pairing did not contain either of these packets, we could check the exchange to see if simple pairing mode was enabled, which indicates that the device is at least using Bluetooth version 2.1.

Gaps: Some older versions of Bluetooth do not require that the device list its version number when pairing, so we were not able to list a specific version of Bluetooth for all devices. However, if the devices were using Secure Simple Pairing, we could assume that the version being used was at least 2.1.

Guidance: Software information should be available to device owners to understand the device capabilities (e.g., available network protocols, compatible applications, operating system). For first responders, additional information about the specifics of the network protocols should be provided. For example, with Bluetooth, the device owner should have the information about what version of Bluetooth is being used and what security levels are enabled within the device.

Benefits: Devices that use newer versions of Bluetooth can utilize more security features that have been built into the pairing mechanisms between devices. Recognizing the differences between versions of Bluetooth can encourage public safety organizations to purchase devices that clearly state the software specifications for the devices they are using to ensure that they have the capabilities necessary to meet their security objectives (e.g., confidentiality, integrity, and availability).

B.2.3 Test 3: Device Ruggedization Ratings

Security Objective: Availability

Test Description: This will identify the IP ratings and any other information available for the device.

Test Procedures: Most devices were accompanied by datasheets and technical documentation that contained ruggedization information, specifically IP ratings and operating temperatures. Examining the IP ratings and operating temperatures in this documentation was sufficient to determine what physical limitations the device had.

Test Outcome: Most wearable devices were accompanied by IP ratings in their technical documentation, with varying capabilities when it came to dust and water protection. The least protected wearable devices had protection against limited dust ingress and low-pressure water jets, while the best protected wearables had protection for total dust ingress and were submersible up to 1 meter in water. Most wearable devices had relatively durable operating temperatures, with some allowing devices to operate at temperatures below 0° F and as high as 122°F. Some of the wearable devices examined contained drop tests as well and had varying results between 6 to 10 feet. Some devices did not contain significant technical documentation information like operating temperatures and IP ratings could not be obtained.

Analysis: Most wearable devices have significant durability because they were built for everyday use. Wearable devices that have little to no protection against dust and water are limited in where and how they can be used effectively, so most wearable devices are required to have a certain level of protection that allow for them to be used by consumers wherever possible. This makes them particularly useful for public safety professionals because wearable devices need to be durable and dependable for public safety professionals to incorporate them into their jobs. Devices that can withstand extreme operating temperatures and have significant protection against water are particularly useful since they can be used in most climates that a public safety professional will experience. It is important for device manufacturers to provide easy access to this information so consumers can evaluate the conditions that the wearable device can handle and decide whether the device will be capable of withstanding the environment that it will be in.

Gaps: Some devices did not contain IP ratings and operating temperature ranges in their technical documentation, so the durability of these wearable devices could not be evaluated. Providing these details in technical documents can be very important for public safety professionals to determine whether or not they can be used.

Guidance: Public safety device administrators should be aware of their ruggedization ratings for their wearable devices. These devices are typically worn on a first responder's body and may be more exposed to elements than other devices/sensors.

Benefits: Devices that have a wider range of operating temperatures, significant dust ingress protection, and water protection are more dependable for public safety professionals to use in their everyday tasks. Better protection also means that these devices can be used in more significant ways that could help public safety professionals have better tools to work with in situations with bad weather conditions or in unsafe environments.

B.2.4 Test 4: Obtaining Vulnerability Information from OS Information

Security Objective: Integrity, Device & Ecosystem Health

Test Description: This test will have NIST engineers manually check the software versions of the OS that shipped within the device against a list of vulnerabilities within public databases to understand the types of vulnerabilities already known within the OS. These will include the National Vulnerability Database (NVD), VulnDB, and the vulnerability bulletins from Apple, Google, and the public safety handset manufacturers. Engineers will look to understand the impact and criticality of all the known vulnerabilities.

Test Procedures: Researchers could extract version information pertaining to Bluetooth from each device by parsing packet captures using Python. Bluetooth versions earlier than 4.0 do not include the "Low Energy" and "Bluetooth Smart" additions to the protocol so devices that used these earlier versions were identified as having potential vulnerabilities.

Test Outcome: Most devices used versions of Bluetooth that supported Secure Simple Pairing, which would indicate that the device supported at least Bluetooth version 2.1. This version of Bluetooth allows for encryption key sizes to be negotiated, so an attacker can negotiate a smaller key size in an effort to help them break the encryption set up by Secure Simple Pairing. In addition, mutual authentication may not be required with this and versions of Bluetooth prior to 3.1. The "Just Works" pairing method was observed in most devices, since it requires the least number of security features to be enabled, however this method of pairing provides no man-in-the-middle protection. Devices that use this method for pairing, even in versions of Bluetooth up to 4.2, are susceptible to a man-in-the-middle attack where an attacker can obtain the authentication and encryption key(s) from each device and observe and inject Bluetooth packets between devices. Devices using Bluetooth versions prior to 4.0 also use the E0 stream cipher, which is relatively weak and is supplemented with FIPS approved algorithms in later versions of Bluetooth.

Analysis: Through observing packet captures, information about the version of Bluetooth being used by the device and security features that were enabled could be extracted to provide insight into what vulnerabilities the device was likely to have. Most devices using Secure Simple Pairing were using Security Mode 4 but did not have man-in-the-middle protection enabled. Wearable devices often do not have a method for a user to input anything like a display or text keyboard, so enabling man-in-the-middle protection would require the device to have a static pin number that it can use to set up this protection with the controlling device. Devices using a version of Bluetooth greater than 4.0 use the Bluetooth "Low Energy" pairing process that contains the same limitation, so device manufacturers need to ensure that man-in-the-middle-protection can be enabled through using a static pin number and the "Passkey" pairing method as opposed to the "Just Works" pairing method. This static pin number should not be obvious or included in technical documentation since attackers can easily find what the pin number is and disable the man-in-the-middle protection. Bluetooth was designed to be backwards compatible with earlier versions of itself, which means that devices will commonly try to connect using legacy methods that can possibly be less secure than more recent implementations.

Gaps: Prior to Bluetooth version 4.0, there was not an explicit packet that designated what version of Bluetooth was being used in the device's pairing process. Since Secure Simple Pairing was introduced in version 2.1, we can only assume that the devices are using at least version 2.1 when the "Read Remote Version Information" or "Read Local Version Information" packets are not present in a packet capture of a device's pairing process.

Guidance: Public safety device administrators should be aware of the Bluetooth version used on their wearable devices and the potential vulnerabilities with using a particular version. PSCR Engineers performed packet captures to obtain the Bluetooth version. It would be helpful if this information was provided by the manufacturer within the device manual. With this information, a device administrator can identify and assess the risk of using that device.

Attackers will often intentionally display or use an earlier version of Bluetooth to force the device to authenticate and pair using a less secure process, so device manufacturers need to take this into account when evaluating the security of their wearable devices. Device manufacturers need to carefully observe what "Security Mode" their device will downgrade to when the controlling device does not support a recent or commonly used version of Bluetooth, in order to make sure that there is no situation where the device can be connected to and used with low to no security measures.

Benefits: Identifying a device's Bluetooth version and pairing mechanisms gives an in-depth view on what security measures the device can support and what measures it has enabled. Earlier versions of Bluetooth have significant vulnerabilities that are somewhat addressed in more recent versions of Bluetooth but are not always enabled or enforced by default. Using packet captures also allows researchers to perform an unbiased analysis of the device and allows for providing additional information about the device's capabilities along with what may or may not be present in a device's technical documentation.

B.2.5 Test 5: Bluetooth Pairing

Security Objective: Authentication

Test Description: This test will identify how the wearable device pairs and authenticates to a mobile device, such as the use of an insecure pairing mechanism. Investigate any encryption, privacy protections, device names, and insecure pairing types.

Test Procedures: To examine authentication mechanisms packet captures were examined between wearable devices and the mobile devices that contained software to be able to interact with them. Many wearable devices are accompanied by third party applications, so capturing packets gave the opportunity to examine how the wearable device would attempt to authenticate when being used as intended. To facilitate identifying authentication information in packet captures, automation methods using Python were implemented to extract meaningful information related to device version information and flags that were enabled during pairing such as secure simple pairing, man in the middle protection, and out of band information. The presence of these fields in each packet determines the level of privacy protection that the wearable device will use and is an indicator for what kind of encryption the device will use as well.

Test Outcome: All of the wearable devices contained an authentication mechanism, although how this mechanism was implemented varied depending on what version of Bluetooth the device was using. Some devices did not use Bluetooth at all, since they contained a wireless networking interface that they could use to access all of their components over the local area network. In this case the devices used WPA2 passwords to handle authentication, but packet payload encryption was not available for all devices. Devices that primarily used Bluetooth to communicate enforced authentication through Bluetooth's simple pairing mode, which will set up a symmetric key between each device upon pairing. Before the symmetric key is established between the devices, the host device sends a user confirmation request packet to the controller device. The controller device then needs to respond with the corresponding link key to authenticate to the host device.

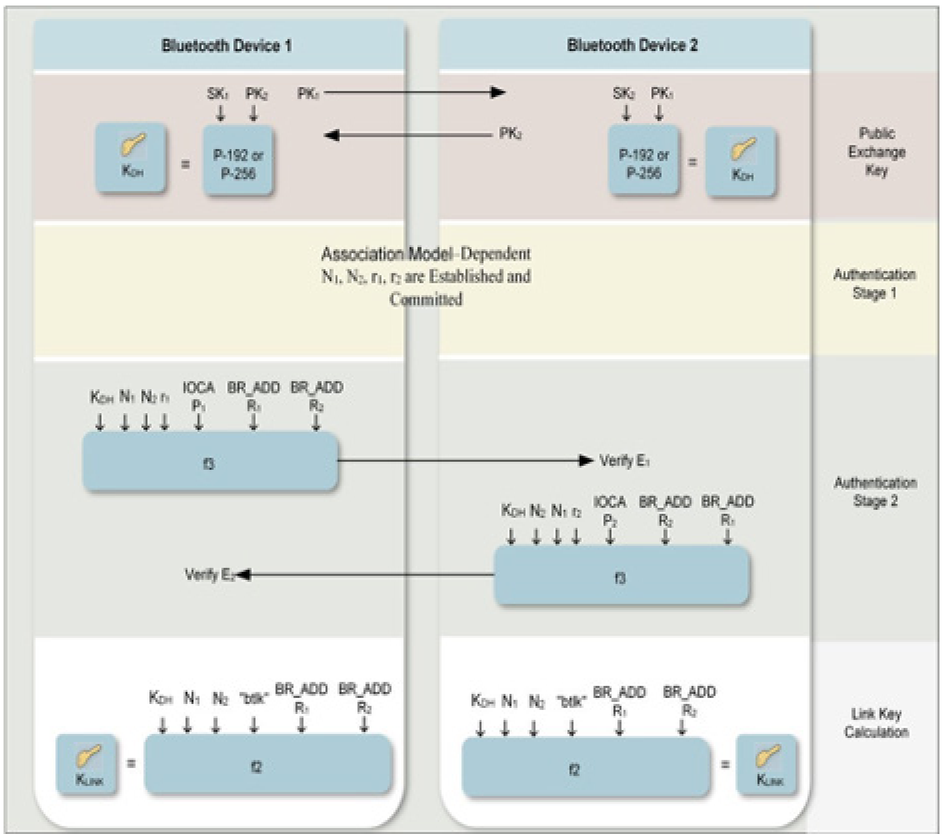


Figure 25 - Link Key Establishment for Secure Simple Pairing (NIST SP 800-121) [17]

If the link key is not provided, then the device will either set up a new connection or refuse to pair with the controller device depending on its authentication requirements Most of the devices used secure simple pairing to handle authentication, however some appeared to be using Bluetooth's Generic Attribute Profile (GATT) to only handle service level access restrictions. Devices that were compatible with Bluetooth Low Energy (BLE) handled authentication through the low energy pairing process, where identity keys for each device are used among a set of additional keys to calculate a long-term key that is used to verify each device's identity.

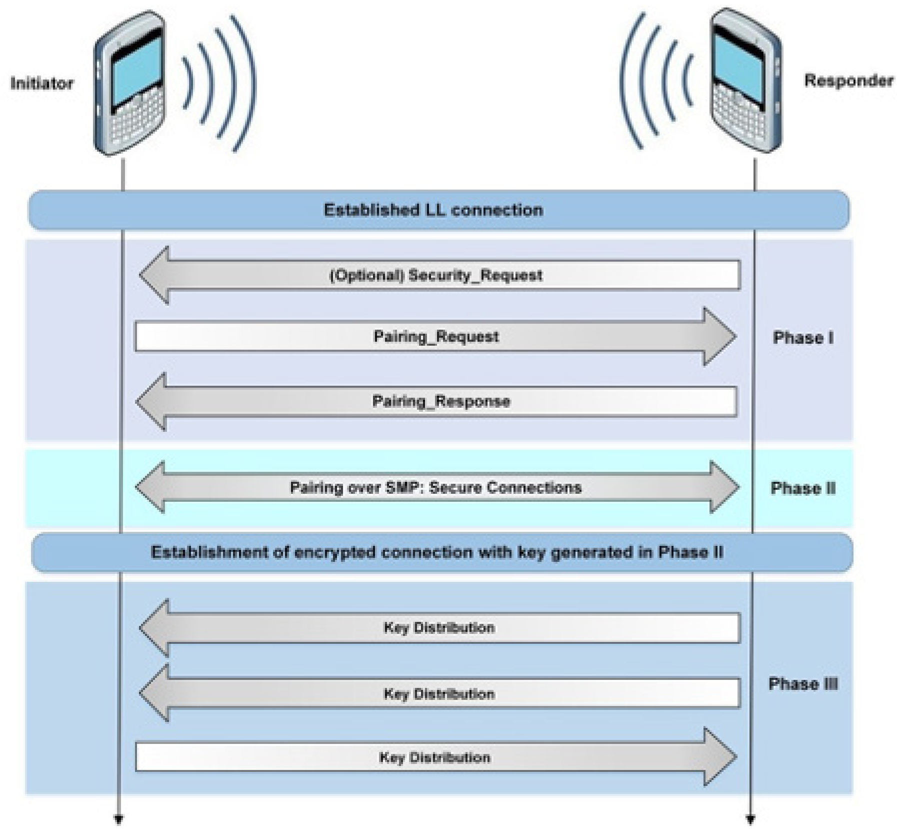


Figure 26 - Bluetooth Low Energy Secure Connections Pairing (NIST SP 800-121) [17]

Analysis: The pairing exchanges for every device could be observed and every device could be successfully paired with, however the version of Bluetooth being used by the device and its input capability determined what kind of authentication would be used. Devices that do not have an interface for a user to interact with cannot require the user to input a PIN number or passcode since there is no way to enter this information, so the device has to either take a predetermined pin code or use an alternative method for handling authentication. Wearable devices using secure simple pairing handle authentication through using a link key and a random number which is calculated during the pairing exchange, so when a host reconnects the controller device can verify its identity. However, the authentication requirements of the controller device can allow for varying restrictions on devices that do not authenticate correctly, from automatically accepting a new connection to refusing a connection with the host device. Secure simple pairing also does not provide man in the middle protection since a single link key is calculated between the devices, so Bluetooth version 4.0 and above have adapted a more robust pairing mechanism to authenticate devices. This pairing mechanism is referred to as “Bluetooth Smart” and “Bluetooth Smart Ready” for host and controller devices and involves creating a “long term key” from a series of key exchanges between the devices. These key exchanges allow the devices to handle authentication by securely sending keys from one device to the other, instead of the devices calculating them individually. Bluetooth Smart can provide man in the middle protection if both devices can input a six-digit code, but if the controller device has no input capability then no man in the middle protection is applied. One device examined used a static PIN code with

Bluetooth Smart, that provided man in the middle protection but was listed in their technical documentation and could be easily guessed to allow for a successful connection to the device.

Gaps: Bluetooth is designed to be able to successfully pair with devices using older versions of Bluetooth, so when examining the pairing between devices the wearable device may use an older method of pairing if the host device is using an older version of Bluetooth. In addition, the authentication requirements of the wearable device can be set to allow automatically accepting new connections. This is common in wearable devices since they do not have an interface to interact with, so some are built to constantly try to accept new connections without a set number of allowed attempts.

Guidance: Public safety device administrators should be aware of the device pairing process for their IoT devices. This pairing process is often based on the network protocols (discussed in Test B.1.2) available within the device (e.g., Wi-Fi, Bluetooth, NFC, etc.). Device manufacturers should include information about the pairing capabilities within the device manuals and also consider providing different pairing options. By providing information on different device pairing options, this allows public safety officials to enable the authentication process that meets their various needs.

Benefits: It is important that wearable devices used by Public Safety are appropriately authenticated to interact with other Public Safety devices (e.g., mobile devices) and/or public safety resources (e.g., computer-aided dispatch (CAD) systems). Evaluating the pairing between devices highlights the important information being passed between devices when the wearable device is being used, and what steps the device will take to protect the confidentiality, integrity, and availability of this information.

Depending on the emergency incident or scenario, a first responder may require immediate access to communications or resources. With this in mind, it is important for device administrators to understand the device authentication/pairing capabilities and consider the risk of implementing different levels of authentication. Certain authentication mechanisms may require more time and interaction from the user, which can negatively impact a first responders response time to an emergency incident.

Devices that use newer versions of Bluetooth have access to more robust security measures that provide better protection from common attacks on wearable devices. Examining the pairing between host devices and wearable devices can give specific information on what requirements for authentication and encryption wearable devices should have to make full use of the security options in newer versions of Bluetooth.

B.2.6 Test 6: Bluetooth Encryption

Security Objective: Confidentiality, Integrity

Test Description: This test will identify how the wearable device communicates with a mobile device, specifically using encryption. This will include the use of a secure algorithm, reasonable key sizes, and any man in the middle protection.

Test Procedures: Similar to the previous authentication testing, automated parsing of packet captures using Python was used to test for encryption mechanisms in wearable devices. When a wearable Bluetooth device pairs with a host device an encryption scheme is determined based on the corresponding versions of each device and the method for authentication. Encryption information could be extracted from packet captures if flags were set during the pairing process such as secure simple pairing, out of band pairing, or man in the middle protection enabled since a Bluetooth device will examine these flags and choose a certain encryption method in versions under 4.0. Later versions of Bluetooth use a more complicated process which uses multiple temporary encryption keys to calculate a long-term encryption key, so encryption information can be extracted from multiple packets that carry these encryption keys.

Test Outcome: All devices pairing using Secure Simple Pairing enforced link level encryption using a shared link key, with some devices explicitly setting an encryption key size when paired. The pairing exchanges between devices do not mention specific algorithms being used to generate keys but does indicate whether encryption is enabled and provides a code that indicates what type of encryption key was used to encrypt the data. Secure simple pairing uses elliptic-Curve Diffie Hellman (ECDH) public key cryptography to generate key pairs between devices starting with version 2.1 and includes four levels of link key authentication that services on Bluetooth devices can enforce (see Figure 27).

- Level 4: Authenticated link key using Secure Connections required
- Level 3: Authenticated link key required
- Level 2: Unauthenticated link key required
- Level 1: No security required
- Level 0: No security required. (Only allowed for SDP)

Figure 27 - Security Requirements for Services Protected by Security Mode 4 (NIST SP 800-121) [17]

All of the devices examined using Secure Simple Pairing enforced unauthenticated link keys, which would correspond to Security Level 2. Security Level 1 corresponds to no security at all, Security Level 3 enforces using authenticated link keys, and Security Level 4 enforces using Secure Connections. All devices examined used Bluetooth versions 2.1 to 4.0, which corresponds to using the Bluetooth E0 encryption algorithm, which uses the 128-bit link key, 128-bit random number, and an encryption key to encrypt packet data. Newer versions of Bluetooth do not use the E0 algorithm because it is not Federal Information Processing Standards (FIPS) approved and is considered a relatively weak algorithm for encryption. Bluetooth Low Energy (BLE) and versions of Bluetooth after 4.1 use a stronger encryption algorithm called Advanced Encryption Standard-Counter with Cipher Block Chaining Message Authentication Code (AES-CCM) which is FIPS approved and helps to resolve a lot of the shortcomings of the E0 algorithm. Man-in-the-middle protection was not enabled with most of the wearable devices since Bluetooth depends on the user being able to enter or verify a numerical PIN, and most wearable devices do not contain the ability to enter data through a keyboard. One device set a static PIN for use with the BLE Secure Connections pairing, which provides man in the middle protection but makes the static pin easy to guess through a brute force attack or easily identified in user manuals. Key sizes for devices ranged between 7 and 16 bytes for encryption keys, some of which were set by the controller device during pairing.

For Security Mode 4, the Bluetooth specification defines five levels of security for Bluetooth services for use during SSP. The service security levels are as follows:

- **Service Level 4** – Requires MITM protection and encryption using 128-bit equivalent strength for link and encryption keys; user interaction is acceptable.
- **Service Level 3**—Requires MITM protection and encryption; user interaction is acceptable.
- **Service Level 2**—Requires encryption only; MITM protection is not necessary.
- **Service Level 1**—MITM protection and encryption not required. Minimal user interaction.
- **Service Level 0**—No MITM protection, encryption, or user interaction required.

Figure 28 - Secure Simple Pairing Service Levels (NIST SP 800-121) [17]

Analysis: The strength and reliability of Bluetooth encryption algorithms is directly related to the pairing mechanisms being used between devices, and many of the inputs for encryption schemes come from outputs of authentication during pairing. With later versions of Bluetooth come more robust pairing schemes which lead to stronger and more reliable encryption algorithms, so keeping up to date with the latest versions of Bluetooth becomes vitally important for protecting the confidentiality of data passing between wearable and mobile devices. Even between the latest three versions of Bluetooth there have been significant improvements to the encryption algorithm being used as well as the authentication mechanisms that Bluetooth uses.

Using more recent versions of Bluetooth also provides additional capabilities when it comes to protecting data integrity. Devices using Secure Simple Pairing only generate a link key that is used to encrypt and decrypt data, but the ability to cryptographically sign packets to ensure they have not been altered in transit after the pairing process is complete did not become available until Bluetooth Smart and Bluetooth Low Energy was introduced in version 4.0. This updated version introduced a Connection Signature Resolving Key (CSRK) that is generated from the same pairing process that creates Long Term Key (LTK) that is used for authentication. This CSRK can be used by the device sending data packets to sign them and the signature can be verified by the receiving device to provide additional data integrity protection.

Gaps: If wearable devices do not have the ability to input a numeric PIN for Security Level 4 then they cannot provide man in the middle protection and have to fall back to using the "Just Works" pairing mechanism. In addition, the ability to have no limit on the attempts made to pair with a device means that an attacker can continually attempt to pair with a device to try to extract any information about encryption or authentication. To determine the Bluetooth encryption levels, PSCR Engineers performed network traffic analysis. This information was not easily available in the device documentation and would require public safety officials to inquire about the device encryption information.

Guidance: Wearable devices that use the classic implementation of Bluetooth should strive to use the latest version of Bluetooth since it includes significant updates to encryption and authentication that are available in Bluetooth Low Energy capable devices. Where applicable, wearable devices should also use Security Level 4 which implements secure connections for both BLE and BDR implementations but be mindful that using secure connections does not guarantee man in the middle protection.

Benefits: Strong encryption algorithms help to protect vital user data for wearable devices, such as devices that measure a user's vital signs or record what a user is doing while working as a public safety professional. First responders, such as law enforcement, may need to keep their location and activities confidential during an operation. Using robust pairing and strong encryption algorithms can help to prevent an attacker from being able to gain access to this data without proper authentication to the device.

B.2.7 Test 7: Configuration Guidance

Security Objective: Integrity, Device & Ecosystem Health, Interoperability

Test Description: This will review the type of guidance provided from the vendor to the public safety professionals, and if any of this is security guidance dedicated to properly owning, operating, and configuring the device for public safety use.

Test Procedures: To identify configuration guidance information, researchers examined user guides and manuals that were shipped with the device. Additionally, researchers examined the vendor's websites and any additional information that could be found through the vendor's documentation for each device.

Test Outcome: The wearable devices examined that used Bluetooth did not provide secure configurations guidance, while the wearable devices that included a networking component did. The quality of guidance varied between devices, with some containing simple instructions and suggestions to some devoting entire webpages and videos to secure configuration. The devices that used Bluetooth primarily did not provide secure configuration guidance since most of the configuration details are set within the Bluetooth firmware and could not be changed by the user.

Analysis: Most of the wearable devices that primarily use Bluetooth did not provide secure configuration guidance since most of the configuration is already established in the firmware. This highlights the fact that secure configuration and use has not been a major focus in the development of wearable devices since manufacturers place more emphasis on usability than security. However, secure configuration plays a major role in how Bluetooth devices can use the available security options present in the most recent versions of Bluetooth, so providing mechanisms for enforcing strict authentication and encryption requirements can help a great deal to close some of the security gaps present in wearable Bluetooth devices.

Gaps: Most wearable Bluetooth devices examined do not provide a mechanism for altering the authentication and encryption requirements present in the device from outside the device's firmware.

Guidance: Public safety device administrators should identify the necessary device configurations and apply them prior to providing the devices to their users.

Benefits: Secure configuration guidance can help users to become aware of the security capabilities of the wearable devices in use and can help users to extend enforcing security policies to wearable devices. By applying secure configurations prior to device deployment, this provides the first responder with a device that is secure whilst requiring minimal to no additional configuration that may interfere with their response to an emergency.

B.2.8 Test 8: Wearable Device MAC Address Randomization

Security Objective: Confidentiality

Test Description: This test will identify if the wearable device is utilizing MAC addresses randomization. This includes the Bluetooth MAC address.

Test Procedures: Bluetooth advertisement packets were collected using Python, which contained the Bluetooth MAC addresses of the devices sending advertisements within range of the capturing device. The specific Bluetooth address of the DUT was already known, so a program was developed that would check this known address against the addresses found in advertisement packets to determine if the device was sending its real Bluetooth MAC address in advertisement packets.

Test Outcome: Most devices do not utilize address randomization as their Bluetooth addresses can be found in advertising messages broadcasted to all devices in the local area network.

Analysis: Bluetooth devices with a version prior to 4.0 and not using Bluetooth Low Energy (BLE) do not have the option to randomize hardware addresses in advertising messages. Since most of the devices observed were using older versions of Bluetooth, MAC address randomization was not expected to be observed. Bluetooth devices that use version 4.0 or later have a feature called "LE Privacy" that will replace the hardware address with a random value that changes at a varying timing interval.

Gaps: Most devices examined were using a Bluetooth version earlier than 4.0, so devices in the future may be able to overcome this limitation through enabling the LE Privacy feature present in the latest versions of Bluetooth.

Guidance: Device address randomization is recommended for first responders that may be involved in situations where tracking their location is problematic and could put them in danger. Public safety device administrators should consider the use cases for each device and ensure it has the appropriate security capabilities. If a feature like LE Privacy is necessary, Public Safety device administrators should ensure they are using the appropriate version of Bluetooth with that capability enabled. This device information could be included with the device manual for easy awareness to the device owner. Additionally, it would useful for an IoT Management Solution to be able to easily extract the devices capabilities and present it to the device administrator through their console.

Benefits: Including this kind of randomization into future wearable devices will help to prevent problematic tracking of public safety wearable devices using the hardware address. With this information readily available, device administrators can make informed decisions when considering the use of a device.

B.2.9 Test 9: Device Update Policy

Security Objective: Device & Ecosystem Health

Test Description: This will seek to understand how often the device is scheduled to receive security updates and other software from the vendor. Specifically, the regularity / cadence, type, and reasons for updating the device and applying security patches will be reviewed.

Test Procedures: To identify update policy information, researchers examined the device vendor's user guides and manuals to see what steps they recommended taking to apply updates and upgrades to each device. When this information could not be found through the device's documentation the vendor's website and any additional information that vendor provided was examined.

Test Outcome: Most wearable devices examined do not contain update policies that schedule regular updates for security. The devices examined either did not contain any mechanism to update the device, required that the device be sent back in for updates to be applied, or could only be updated manually using additional applications and software packages that needed to be purchased separately. Since most devices primarily used Bluetooth, they did not contain a way to regularly check for updates through an online provider unless the user had access to an application or tool on a separate device that could check for updates.

Analysis: Wearable devices using Bluetooth cannot manage identifying updates on their own since they do not have a network connection, so scheduling security updates for these devices needs to be managed by another device. Many of the devices examined included applications or command line tools for a host device in the local piconet to handle updating the firmware on devices. While these applications could successfully update the firmware on the wearable devices, they rarely included information on what specific updates were being applied, so users could not be made aware of whether specific versions of components were being upgraded.

Gaps: Wearable devices cannot seek out updates on their own and need a separate application or tool to be able to install the newest versions of firmware available.

Guidance: Public safety device administrators should be aware of any devices update polices to be informed of the following:

- Device update schedule – to plan and ensure updates do not conflict with first responder daily work activities
- Device security updates – to patch vulnerabilities that may leave a first responder's device vulnerable to attack
- Device functionality updates – to address bug fixes and be aware of any new/removed capabilities provided within the device
- Device support period – to know how long a device is supported and prepare for end-of-life, device disposal, and device refresh.
- Device interoperability changes – to be aware if the update impacts the wearable devices compatibility with applications and different device platforms (e.g., Windows, MacOS, iOS, and Android)
- Applying device update – to understand how the device must be updated (e.g., automatically, manually, or through purchase of a new device)

Benefits: Device update policies can help keep wearable devices equipped with the latest versions of Bluetooth that implement the most robust and secure pairing and encryption mechanisms available.