Evaluating Mobile App Vetting Integration with Enterprise Mobility Management in the Enterprise

June 26, 2019

Homeland Security

Science and Technology

Executive Summary

Federal agencies increasingly use mobile devices and mobile applications (apps) to meet their mission and business needs and improve productivity and efficiency. The ubiquity of mobile apps and the increased reliance on their use has a counter side, however. Mobile apps pose substantial risk to federal enterprises because of their potential for exploitable vulnerabilities, malicious code, or privacy-violating behaviors and should be deployed with care. Even apps from the Google Play or Apple App Stores are not free of these risks. Mobile app vetting solutions can automate security analysis of mobile apps to help enterprises determine whether apps are safe to deploy on mobile devices. This generally takes time to review and act upon the findings from these solutions. Enterprise mobility management (EMM) provides the centralized capability to manage an enterprise's mobile devices, including provisioning security policies to the devices. Many EMM and mobile app vetting solutions advertise integration capabilities—the mobile app vetting solution can share an inventory of installed apps with the EMM, and the EMM can take action based on app vetting findings.

The Mobile Security Research and Development (R&D) program within the Department of Homeland Security (DHS) Science and Technology Directorate (S&T) promotes such adoption of safe and secure mobile technology within DHS and across the federal government, and encourages development and adoption of integrated cybersecurity solutions to improve mobile security for the federal government. To help promote this adoption and explore other solutions, the team solicited the Homeland Security Systems Engineering and Development Institute (HSSEDI) to perform an independent evaluation of the integration capabilities of mobile app vetting and EMM solutions. This would help assess maturity of the integrated solutions and provide guidance to federal government users and to industry on integration and interoperability of the solutions. Integrated solutions can inform automated updates to agency mobile app blacklists or whitelists, reducing exposure of federal users' devices and federal systems to mobile app threats.

This report provides the results of HSSEDI's evaluation of the integration of app vetting capabilities with EMM, conducted between November 2018 and May 2019. It builds on prior work described in MITRE Technical Report 160242, *Analyzing the Effectiveness of App Vetting Tools in the Enterprise* [3], which focuses on the ability of mobile app vetting tools to find vulnerabilities and privacy violating and malicious behaviors in mobile apps. This report presents background information on EMM, app vetting, and MTD tools and describes the requirements and evaluation criteria generated for the assessment.

EMM provides protection at the device level but does not have an ultimate view into the behavior of the apps that execute on the device. By creating complementary solutions that integrate app vetting tools with EMM, federal enterprises can strengthen their security posture. Further, Mobile Threat Detection (MTD) solutions, much like the endpoint detection and response solutions for desktop or laptop systems, can detect and defend against runtime security threats, often containing app vetting or similar services in conjunction with device- and network-

level protections. MTD solutions can be similarly integrated with the EMM to provide a more holistic security approach than either one working alone.

An EMM solution can take a variety of actions in response to app vetting findings, to include: notifying an administrator to perform further inspection; notifying the user that an app violates enterprise policy; restricting use of the app; or restricting access to enterprise resources until the issue is mitigated. This continuous app vetting approach leverages the analysis capabilities of app vetting tools to periodically inspect apps installed on enterprise devices for security issues, and shepherd those results to the EMM solution for potential action. Such integration can enable enterprises to seamlessly make use of both EMM and mobile app vetting solutions.

For this task, HSSEDI evaluated two market-leading EMM solutions and compared the integration capabilities of each to those of six leading mobile app vetting solutions.[1] The team performed 43 test cases involving a select set of commercial and custom-developed apps. The findings, as depicted in Figures ES1 (Android) and ES2 (iOS), show little difference between the integration capabilities of the EMM solutions, although some EMM solutions have a tighter integration with a particular mobile app vetting or MTD solution. The integrated EMM and app vetting solutions exhibit some common strengths; however, multiple strengths and drawbacks differentiated each app vetting solution's capabilities.

[1] As used herein, "mobile app vetting solutions" refers to both standalone mobile app vetting tools and MTD products that include mobile app vetting capabilities.

Table ES 1 Overall Android Test Results

Assessment Criteria	Test Case (Android)	Solution 1 + EMM A	Solution 1 + EMM B	Solution 2 + EMM A	Solution 2 + EMM B	Solution 3 + EMM A	Solution 4 + EMM A	Solution 4 + EMM B	Solution 5 + EMM A	Solution 5 + EMM B	Solution 6 + EMM A	Solution 6 + EMM B
1A. Platforms supported		1	1	2	3	2	2	1	2	1	2	2
1B. Obtain inventory of apps		1	1	1	1	1	1	1	1	1	1	1
1C Uniquely identify apps	1 Rename app	1	1	2	0	1	1	1	1	1	1	1
	2 Repackage app	3	3	2	0	3	1	1	1	1	1	1
	3 Modify version number	3	3	2	0	3	1	1	1	1	1	1
	4 Enterprise app store	2	2	2	0	3	2	2	3	3	2	2
1D Periodic app security analysis	1 Schedule/frequency/event.	0	0	0	0	1	0	0	0	0	1	1
	2 Verify schedule	1	1	3	0	1	1	1	1	1	3	3
	3 App updates	1	1	3	0	1	1	1	1	1	1	1
	4 Wait three days	1	1	3	0	1	1	1	1	1	1	1
	5 Apply custom rule	3	3	2	0	2	2	2	1	1	0	0
1E Detect out-of-date apps	1 Verify app is up-to-date	3	3	3	0	3	3	3	3	3	1	1
	2 Flag out-of-date apps	3	3	3	0	3	3	3	3	3	1	1
1F Identify sideloaded apps	1 Sideloading	2	2	2	0	1	1	3	2	1	1	1
	2 Enterprise app store sideloading	3	3	3	0	3	3	1	3	3	3	3
1G Analyze sideloaded applications		1	1	3	0	2	1	1	1	1	2	2
1H Compare apps in app stores		3	3	3	0	1	1	1	3	3	3	3
1I Act on detection of sideloaded app	Tested with 1F, admin notification	3	3	3	0	2	1	3	1	1	1	1
	Tested with 1C, not an app store app	3	3	2	0	1	1	1	1	1	1	1
2A App exist in mainstream app stores. App popularity.	Tested with 1C, app from app store	1	1	2	0	2	2	2	1	1	1	1
	Tested with 1C, app popularity	3	3	3	0	2	3	3	3	3	1	1
2B Other apps from developer, popularity and issues identified.	1 Multiple apps from developer	1	1	3	0	3	3	3	3	3	1	1
	2 Popularity of other apps	2	2	3	0	3	3	3	3	3	1	1
	3 Other app issues identified	3	3	3	0	3	3	3	3	3	2	2
2C Repackaged or counterfeit	Tested with Test 2, detect repackage	3	3	2	0	3	1	1	3	3	1	1
3A Dynamic code execution		3	3	1	0		1	1	1	1	1	1
3B Report network connections	1 Report use of ad network	1	1	1	0	2	1	1	1	1	1	1
	2 Reports necessary communication	1	1	1	0	1	1	1	1	1	1	1
	1 Identify secure protocols	2	2	1	0	1	1	1	1	1	1	1

Assessment Criteria	Test Case (Android)	Solution 1 + EMM A	Solution 1 + EMM B	Solution 2 + EMM A	Solution 2 + EMM B	Solution 3 + EMM A	Solution 4 + EMM A	Solution 4 + EMM B	Solution 5 + EMM A	Solution 5 + EMM B	Solution 6 + EMM A	Solution 6 + EMM B
3C Improper use of networking protocols	2 NIAP communication requirements	3	3	2	0	1	1	1	2	2	1	1
3D Detect anomalous behavior	1 Detect anomalous behavior	2	2	1	0	1	2	1	1	1	1	1
	2 Ad network anomalous behavior	2	2	1	0	3	2	1	3	3	1	1
	3 Anomalous behavior false positive	1	1	1	0	1	1	1	1	1	1	1
4A EMM app vetting results/response		1	1	3	0	1	2	2	1	1	3	3
4B Quantifiable measure of risk		2	2		0	1	2	2	2	2	2	2
4C Configurable risk measures		1	1	1	0	2	2	2	2	2	1	1
4D Blacklist/whitelist		2	2	2	0	1	1	1	2	2	2	2
4E EMM and app vetting solution availability	1 EMM/app vetting solution availability	1	3	1	0	1	1	2	1	3	1	3
	2 Solution availability, modification	1	1	1	0	1	1	1	1	1	1	1
5A Solution secure connection	1 (part 1) Solution secure connection	1	1	1	1	1	1	1	1	1	1	1
	1 (part 2): NIAP tests	1	1	1	1	2	1	1	2	2	2	2
	Test 2 (part 2): More NIAP tests	0	0	0	0	0	0	0	0	0	0	0
5B EMM administrator credentials		1	1	1	1	1	1	1	1	1	1	1

Table ES 2 Overall iOS Test Results

Assessment Criteria	Test Case (iOS)	Solution 1 + EMM A	Solution 1 + EMM B	Solution 2 + EMM A	Solution 2 + EMM B	Solution 3 + EMM A	Solution 4 + EMM A	Solution 4 + EMM B	Solution 5 + EMM A	Solution 5 + EMM B	Solution 6 + EMM A	Solution 6 + EMM B
1A. Platforms supported		1	1	2	3	2	2	1	2	1	2	2
1B. Obtain inventory of apps		1	1	1	1	1	1	1	1	1	1	1
1C Uniquely identify apps	1 Rename app	3	3	2	0	1	3	3	2	1	1	1
	2 Repackage app	3	3	3	0	3	3	3	2	2	1	1
	3 Modify version number	3	3	3	0	3	3	3	3	3	1	1
	4 Enterprise app store	3	3	2	0	3	2	2	1	1	2	2
1D Periodic app security analysis	1 Schedule/frequency/event.	0	0	0	0	1	0	0	0	0	1	1
	2 Verify schedule	1	1	3	0	1	2	2	1	1		
	3 App updates	1	1	3	0	1	3	3	1	1	1	1
	4 Wait three days	1	1	3	0	1	3	3	1	1	1	1
	5 Apply custom rule	1	1	3	0	2	2	2	1	1	0	0
1E Detect out-of-date apps	1 Verify app is up-to-date	3	3	3	0	1	3	3	3	3	1	1
	2 Flag out-of-date apps	3	3	3	0	2	3	3	3	3	1	1
1F Identify sideloaded apps	1 Sideloading	1	1	3	0	1	3	3	3	1	1	1
	2 Enterprise app store sideloading	3	3	3	0		1	1	1	3	3	3
1G Analyze sideloaded applications		3	3	3	0	2	3	3	3	3	2	2
1H Compare apps in app stores		3	3	3	0	1	1	1	3	3	3	3
1I Act on detection of sideloaded app	Tested with 1F, admin notification	1	1	3	0	2	3	3	1	1	1	1
	Tested with 1C, not an app store app	1	1	2	0	3	3	3	3	2	1	1
2A App exist in mainstream app stores. App popularity.	Tested with 1C, app from app store	1	1	2	0	2	2	2	2	2	1	1
	Tested with 1C, app popularity	3	3	3	0	2	3	3	3	3	1	1
2B Other apps from developer, popularity and issues identified.	1 Multiple apps from developer	1	1	3	0	3	3	3	3	3	1	1
	2 Popularity of other apps	2	2	3	0	3	3	3	3	3	1	1
	3 Other app issues identified	3	3	3	0	3	3	3	3	3	2	2
2C Repackaged or counterfeit	Tested with Test 2, detect repackage	3	3	3	0	3	3	3	3	3	1	1

Assessment Criteria	Test Case (iOS)	Solution 1 + EMM A	Solution 1 + EMM B	Solution 2 + EMM A	Solution 2 + EMM B	Solution 3 + EMM A	Solution 4 + EMM A	Solution 4 + EMM B	Solution 5 + EMM A	Solution 5 + EMM B	Solution 6 + EMM A	Solution 6 + EMM B
3A Dynamic code execution		1	1	1	0	3	1	1	1	1	3	3
3B Report network connections	1 Report use of ad network	1	1	1	0	2	1	1	3	3	1	1
	2 Reports necessary communication	1	1	1	0	1	1	1	3	3	1	1
3C Improper use of networking protocols	1 Identify secure protocols	3	3	1	0	1	1	1	1	1	1	1
	2 NIAP communication requirements	3	3	2	0		2	2	2	2	2	2
3D Detect anomalous behavior	1 Detect anomalous behavior	1	1	1	0	1	1	1	1	1	1	1
	2 Ad network anomalous behavior	2	2	1	0		1	1	3	3	1	1
	3 Anomalous behavior false positive	1	1	1	0	1	1	1	1	1	1	1
4A EMM app vetting results/response		1	1	3	0	1	2	2	1	1	3	3
4B Quantifiable measure of risk		2	2	1	0	1	2	2	2	2	2	2
4C Configurable risk measures		1	1	1	0		2	2	1	1	1	1
4D Blacklist/whitelist		2	2	2	0	1	1	1	2	2	2	2
4E EMM and app vetting solution availability	1 EMM/app vetting solution availability	1	3	1	0	1	1	1	1	3	1	3
	2 Solution availability modification	1	1	1	0		3	3	3	1	1	1
5A Solution secure connection	1 (part 1) Solution secure connection	1	1	1	1	1	1	1	1	1	1	1
	1 (part 2): NIAP tests	1	1	1	1	2	1	1	2	2	2	2
	Test 2 (part 2): More NIAP tests	0	0	0	0	0	0	0	0	0	0	0
5B EMM administrator credentials		1	1	1	1	1	1	1	1	1	1	1

All the offerings were able to adequately satisfy the app vetting tests with varying levels of detail in the analysis. They were also able to analyze network communication by the app and output a comprehensive, easy-to-read app threat report. Most of the tested solutions could obtain an inventory of apps installed on devices and could re-scan updated apps in a timely fashion.

The services also generally performed well in test cases involving suspicious network traffic. However, most services could not perform reputation analysis, and all offerings either incorrectly labeled custom, non-market apps downloaded from the enterprise app store as sideloaded or

failed to detect a sideloaded app in some way. Detection of spoofed and sideloaded iOS apps was a weak point, almost certainly due to iOS platform restrictions. The team also encountered difficulties in ensuring that the EMM solutions enforced compliance policies linked to threats detected by the app vetting solution. Lastly, few of the solutions were able to report the presence of out-of-date apps.

Integration of EMM and app vetting is still an emerging capability and vendors are actively developing new features and improving their offerings. HSSEDI found no single integrated product that implements all security-relevant capabilities well and recommends several actions vendors can take to improve integration support and other functional capabilities identified in this evaluation. The app vetting and EMM solutions exhibited varying strengths and weaknesses, which can greatly affect organizational decisions regarding the solution that best meets their needs. HSSEDI recommends that agencies review and understand the strengths and limitations of each tool combination and select the EMM and app vetting solution that fits their needs and desired capabilities.

While this report explores app vetting capabilties and their integration with EMM, it does not provide a recommended configuration of such tools to identify mobile app risks and their proposed mitigations. Further work is needed to explore the ability to apply appropriate mitigations on a per-app basis and how best to apply them.

Acknowledgements

The authors, Carlton Northern, Michael Peck, Wesley Jordan, Stelios Melachrinoudis, and Taylor McCorkill of The MITRE Corporation, wish to thank their colleagues who reviewed drafts of this report and contributed content, to include Terri Phillips, Tom Morrissey, Jasen Jacobsen and Margaret MacDonald. The authors also wish to thank Gema E. Howell and Michael A. Ogata of NIST for their review of this report. The authors especially acknowledge Vincent Sritapan of DHS S&T for his technical leadership and sponsorship of this work. Lastly, the authors thank the vendors who participated in this evaluation.

Table of Contents

List of Tables

List of Figures

1 Introduction

Mobile devices and their applications (hereafter "apps") have completely transformed the way enterprises work and conduct business. The ability to communicate, collaborate, and access data while not being tied to a physical location empowers workers to conduct business from anywhere. Apps however, pose substantial risk to enterprises because of their potential to contain exploitable vulnerabilities, malicious code, or privacy-violating behaviors and should be deployed with care. Even apps from the Google Play or Apple App Store are not free of these risks. Mobile app vetting solutions can automate analysis of mobile apps to help enterprises determine whether particular apps are safe to deploy on mobile devices. Traditionally, app vetting takes place before deployment or at update time. Not all of the vetting process can be automated, however, and the review of findings as well as the ultimate decision to accept/deny deployment of the app can result in considerable delay from the time the app is requested until it is approved for use.

Enterprise mobility management (EMM) solutions, also referred to as mobile device management (MDM) systems, provide enterprises with a centralized capability to manage their mobile devices, including any security risks associated with them. Among other capabilities, EMM can provision security policies to devices, provision devices with credentials to access enterprise resources, and monitor aspects of device state, including gathering an inventory of installed applications. EMM inherently provides protection at the device level; as such, they do not have a view into the behavior of the apps that execute on the device. By creating complementary solutions that integrate app vetting tools with EMM solutions, enterprises can achieve a stronger security posture.

Effective integration of EMM and app vetting requires an understanding of the workflow and capabilities expected by end users. To gain this understanding, enterprises must first examine the capabilities of commercially available mobile app vetting tools, especially those that perform functions beyond the traditional static and dynamic analysis of binaries or source code.

Numerous vendors provide a type of automated app vetting tool (sometimes also known as app threat intelligence or threat protection services) that collects large amounts of publicly available mobile app binaries and runs static and/or dynamic analysis to detect security vulnerabilities, maliciousness, or privacy-violating behaviors. Many of these tools regularly crawl commercial app stores, automatically analyzing new app versions using the vendors' evolving knowledge of mobile threats and making the analysis results available to enterprise customers. Others perform similar analysis but may additionally obtain apps by using crowdsourcing techniques via an agent app installed directly on end-user devices.

App vetting may sometimes include reputational analysis of apps and their developers. App vetting solutions may also give enterprises the ability to directly submit in-house-developed apps for analysis. Leveraging these commercial offerings enables enterprises to streamline app vetting, thereby decreasing the cost and time associated with analysis while potentially improving security by staying up-to-date with emerging threats and app versions.

App scanning, intelligence, and reputational analysis services are often (but not always) features of a category of services called Mobile Threat Defense (MTD), sometimes also referred to as Mobile Threat Prevention (MTP). MTD solutions typically extend beyond the app and also analyze the device and its network communications for security issues. Thus, app vetting may be a standalone solution or integrated as part of an MTD or a mobile security platform.

1.1 Purpose

Many EMM and mobile app vetting solutions advertise capabilities to integrate with each other, or at the very least publish interfaces that could be used to enable integration. EMM solutions can share the inventory of installed apps with an app vetting solution and can perform actions based on app vetting results. Such integration can enable enterprises to seamlessly make use of both EMM and mobile app vetting solutions to automate the entire app vetting process, referred to in this report as "continuous app vetting".

The Mobile Security Research and Development (R&D) program within the Department of Homeland Security (DHS) Science and Technology Directorate (S&T) promotes the adoption of safe and secure mobile technology within DHS and across the federal government. To accomplish program goals, the S&T program manager collaborates with DHS programs such as Continuous Diagnostics and Mitigation (CDM) and the .gov Cybersecurity Architecture Review (.govCAR). Their joint efforts seek to encourage development and adoption of integrated cybersecurity solutions to improve an agency's mobile security posture over standalone solutions.

The DHS S&T Program Manager asked the Homeland Security Systems Engineering and Development Institute (HSSEDI) to perform an independent evaluation of the integration capabilities of mobile app vetting and EMM solutions to assess maturity of the integrated solutions and provide guidance to federal government users and to industry on integration and interoperability of the solutions. Integrated solutions can inform automated updates to agency mobile app blacklists or whitelists, reducing exposure of federal users' devices and federal systems to mobile app threats.

This report presents the results of that evaluation. It outlines how enterprises can use the integration of app vetting tools with EMM to improve the overall security posture of a mobile deployment while simultaneously putting more apps in the hands of end-users. It presents use cases, risks, market analysis, requirements, an evaluation of commercially available products, and finally recommendations to the enterprise.

1.2 Background

This section describes sources of threats the Homeland Security Systems Engineering and Development Institute (HSSEDI) testing team consulted when assembling the evaluation criteria for the assessment [6].

In MITRE Technical Report 160242, titled *Analyzing the Effectiveness of App Vetting Tools in the Enterprise* [3], the authors provided an in-depth evaluation of mobile app vetting tools. The study focused on the ability of the tools to find vulnerabilities and privacy-violating and malicious behaviors in mobile apps. It did not describe the practicalities of how such tools would

be employed, merely their ability to detect these problems given a mobile app. The study cited a need for future work to examine app intelligence services that gather app/developer reputation and known threat information. It also took into account the ability of the app vetting solution to integrate with EMM solutions but did not test or identify what capabilities should be present in such a configuration. This evaluation is, in part, follow-on work to this previous report.

MITRE's Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) framework details specific techniques that adversaries can use to obtain unauthorized access to enterprise resources or achieve other adversarial objectives [10]. ATT&CK for Mobile describes techniques that adversaries can use to gain access to mobile devices and then make use of that access to achieve their objectives. The entry for each technique typically includes a detailed technical description, mitigations, detection analytics, examples of use by adversaries, and references. The techniques are organized into tactic categories, which represent higher-level adversarial objectives.

As described in ATT&CK, mobile apps are a significant attack vector against mobile devices, as they can provide an adversary with the ability to execute arbitrary code on the devices. Access to mobile devices could give adversaries access to enterprise data stored either on the device itself or on backend enterprise systems. It could also provide the adversary with access to other valuable information, for example data obtained from the variety of sensors present on mobile devices (e.g., microphone, camera, Global Positioning System [GPS]). App vetting provides some form of mitigation against 34 out of the 76 techniques currently included in ATT&CK for Mobile.[2]

The techniques mitigated by app vetting are spread throughout ATT&CK for Mobile's tactic categories. Many of the techniques fall into the "Collection," "Credential Access," and "Discovery" tactic categories. Adversaries can use "Collection" techniques to gather data stored on the mobile device, transmitted through the mobile device, or obtained through the mobile device's sensors. "Credential Access" techniques can be used to gather passwords, tokens, cryptographic keys, or other values that the adversary could use to gain unauthorized access to resources. "Discovery" techniques allow an adversary to gain knowledge about the characteristics of the mobile device and potentially other networked systems.

By default, mobile devices are configured to only allow apps to be installed from the device platform's official app store (e.g., Google Play Store or Apple App Store). This default behavior provides significant protection, as these app stores perform some level of vetting (of both the developer's identity and of the apps themselves), and requiring the apps to be published openly increases the potential for detecting an adversary's actions. However, examples still exist of malicious apps that have been distributed through the official app stores [1, 5, 9]. This implies that relying on this default behavior of only installing apps from authorized app sources is not by itself sufficient to ensure device security, although the presence of sideloaded apps (installed from sources other than the official app store or an authorized enterprise app store) should be treated as an indicator of higher potential risk.

[2] ATT&CK for Mobile provides a capability to query a specific mitigation such as application vetting and list all of the techniques that list that mitigation.

In the United States, the National Information Assurance Partnership (NIAP), managed by the National Security Agency (NSA), oversees the evaluation of commercial off-the-shelf (COTS) information technology (IT) products under the international Common Criteria (CC) [16]. Earning a NIAP certificate signifies that the IT product has met the relevant Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) of the corresponding Protection Profiles (PPs), Extended Packages (EPs), PP-Modules, or a combination of these. It further signifies that the COTS IT product is suitable for procurement by U.S. government agencies as well as other international entities in countries that are included in the Common Criteria Recognition Agreement (CCRA).

Each NIAP PP or EP contains a list of threats it is designed to address. For the evaluation of mobile app vetting and EMM integration, this report refers to several NIAP PPs and EPs with the evaluation criteria. The testing performed by HSSEDI included addressing the threats described in specific PPs and EPs. The PPs and EPs referenced in this report include:

- Collaborative Protection Profile for Network Devices Version 2.0
- Extended Package for Mobile Device Management Agents Version 3.0
- Protection Profile for Application Software Version 1.2
- Protection Profile for Enterprise Security Management – Access Control Version 2.1
- Protection Profile for Enterprise Security Management – Identity and Credential Management Version 2.1
- Protection Profile for Enterprise Security Management – Policy Management Version 2.1
- Protection Profile for Mobile Device Management Version 3.0

1.3 Alignment to National Institute of Standards and Technology Mobile Security Guidance

HSSEDI's evaluation aligns with recommendations contained in National Institute of Standards and Technology (NIST) guidance for mobile app vetting (NIST Special Publication 800-163, *Vetting the Security of Mobile Applications, Revision 1)* and mobile device security (NIST Special Publication (SP) 800-124, *Guidelines for Managing the Security of Mobile Devices in the Enterprise*.

SP 800-163 provides organizations with guidance on mobile app vetting that includes developing security requirements, identifying tools, and determining risk/reward tradeoffs when deploying mobile apps. The document defines a four-phase (*app intake*, *app testing*, *app approval/rejection*, and *results submission*) process for vetting mobile apps. While it may appear the process espoused is a manual one, it does not have to be; indeed, SP 800-163 mentions the ability to automate in multiple places. In the sub-process *app intake,* it states "This process [app intake] is typically performed manually by an organization administrator or automatically by an app vetting system." Section 5.1 provides guidance on the use of managed and unmanaged apps via EMM/MDM. Section 5.2 provides guidance for app whitelisting and blacklisting via mobile application management (MAM)/MDM. Last, Section 5.5, *Automated Approval/Rejection* states in its entirety:

> In some cases, the activities conducted by analysts to derive recommendations for approving or rejecting an app can be automated, particularly if no organization-specific

policies, regulation, etc. are required. Here, an app vetting system used to support the specification of rules can be configured to automatically approve or reject an app based on risk assessments from multiple tools. For example, an app vetting system could be configured to automatically recommend an app if all test tools deem the app as having "LOW" risk. Similarly, an app vetting system could be configured to automatically enforce organization-specific requirements. For example, using metadata extracted during the preprocessing of an app, an app vetting system could automatically reject an app from a specific vendor.

SP 800-124 provides guidance to organizations for the employment of mobile devices utilizing a threat-based, mitigation approach. It presents enterprise security building blocks for mobile device usage that include the use of EMM, virtual private networks (VPNs), app vetting, secure containers, and authentication and recommendations for their policy configuration settings. Appendix A calls out the major controls in the NIST Special Publication 800-53 Revision 4, *Security and Privacy Controls for Federal Information Systems and Organizations* control catalog that affect enterprise mobility. From a Continuous Diagnostics and Mitigation (CDM) perspective, these are the controls to be monitored by an organization with a mobile device deployment. Some of these controls would be satisfied via mobile app vetting solutions and/or EMM integration. For example, RA-5 *Vulnerability Scanning* describes the scanning of information systems and their apps for vulnerabilities with the ability to report findings. Employment of a mobile app vetting tool that has the ability to obtain app lists from every device in the enterprise, scan each occurrence of an app and report vulnerabilities will provide the ability to satisfy this control in an automated fashion.

1.4 Report Structure

This report presents the results of HSSEDI's evaluation of mobile app vetting solution (standalone solution or integrated as part of an MTD or a Mobile Security Platform) integration with EMM solutions. The remainder of the report is structured as follows:

- Section 2 contains information on use cases and solutions to frame the evaluation
- Section 3 addresses the threats and evaluation criteria selected from the NIAP PPs and EPs
- Section 4 discusses the Android and iOS mobile apps used to conduct the testing and the rationale for selecting those apps
- Section 5 summarizes the market analysis conducted to select app vetting and MTD products for the assessment and to inform development of the evaluation criteria
- Section 6 describes the EMM and mobile app vetting products examined for the evaluation.
- Section 7 presents major findings and gaps, with recommendations to address the gaps. The recommendations are provided in two parts: 1) for vendors to improve integration and/or functionality of their tools; and 2) for enterprises considering implementing an integrated app vetting product with their existing EMM solution

2 Mobile App Vetting and EMM Use Cases and Solutions

In MITRE Technical Report 160242 [3], the authors described a process of inspecting apps for both potentially exploitable vulnerabilities and potentially malicious or privacy-violating behaviors. Traditionally, app vetting takes place prior to deployment or at update time as described in NIST SP 800-163. Not all of the vetting process can be automated, and the review of findings as well as the ultimate decision to accept/deny deployment of the app can result in considerable delay from the time the app is requested until it is approved for use. For custom-developed apps this delay may be tolerated, but for third-party apps (weather, mapping, productivity, etc.) where uses are wide and varied, this process is time consuming and can result in long wait times for users. This may even have the unintended side effect of prompting users to find ways of bypassing security measures, thus effectively lowering the overall security of the enterprise. Figure 1 depicts the traditional app vetting process.

Figure 1: Traditional App Vetting Workflow

To streamline this process, some organizations have implemented a continuous app vetting approach by integrating app vetting tools with EMM/MDM. This continuous approach aims to strike a balance between security and the freedom to use apps that employees need to conduct business and accomplish the organization's mission. This approach uses the analysis capabilities of app vetting tools to periodically inspect apps installed on enterprise devices for security issues and relay the results to the EMM solution for potential action. In some cases, the app vetting tool categorizes or scores (configurable by the organization) the findings and acts upon them. Among the actions that the EMM solution can take are notifying an administrator to perform further inspection, notifying the user that an app violates enterprise policy, restricting use of the app (blacklisting), restricting access to enterprise resources, restricting access to specific device profiles (e.g., Bluetooth), or prohibiting use of the employee's device completely until the issue is mitigated. Figure 2 depicts this process.

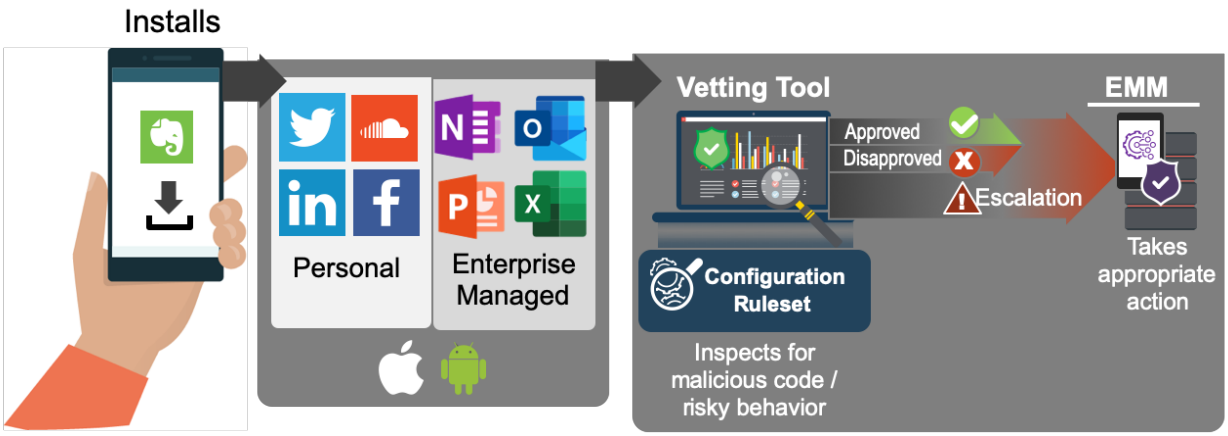

Figure 2: Continuous App Vetting Process

Apps that process sensitive enterprise data can be further separated from those that process only personal or non-sensitive data using device- or operating system-level features such as Android Enterprise, Samsung KNOX Workspace, and Apple iOS-managed apps. These separation capabilities are explored in MITRE Technical Report 150360, titled *Secure Enterprise Access and Personal Enablement of Mobile Devices* [8]. These features, to some extent, defend enterprise apps against vulnerabilities or malicious activities introduced by apps that do not process sensitive enterprise data. Security policies managed by an EMM solution enforce the separation capabilities. This can help to alleviate some of the risk that results from not performing a manual app vetting process for every app installed on users' devices and also has the benefit of enabling the *Bring Your Own Device* (BYOD) or *Corporately Owned-Personally Enabled* (COPE) use cases described in NIST SP 800-124.

2.1 Mobile App Vetting Solutions

Each enterprise must determine its acceptable level of risk when deciding on the necessary scope of app security analysis. Enterprises should also consider the security features provided by the mobile platform (operating system and other underlying on-device technologies as well as broader ecosystem capabilities).

Apps developed in-house are less likely than external apps to contain intentionally malicious or privacy-violating functionality. Vetting of these apps can therefore focus primarily on searching for security vulnerabilities. However, even these apps may present some level of risk of malicious or privacy-violating behavior: for example, third-party software libraries included in the app may include privacy-violating behaviors to enable targeted advertising that are not known to the app developer. Similarly, if the enterprise outsourced all or part of development, it may not be aware of the full behavior of the app yet might still be held responsible for the app's behavior.

Personal-use apps are not intended to process enterprise data. Vetting of these apps can primarily focus on searching for malicious or privacy-violating functionality; in these cases, a search for security vulnerabilities is less critical to the enterprise, because mobile device app sandboxes generally isolate the impact that exploitation of a single app would have on other apps on the

device. However, some risk still exists that an attacker could use a vulnerable app as a vector to exploit the mobile device itself and gain access to enterprise data. Additionally, individual mobile device users would certainly be interested in the potential impact on their personal data of any vulnerabilities in personal-use apps.

Enterprises should consider the properties of modern mobile platforms when determining the required scope of app vetting. The NIAP Protection Profile for Application Software [11] takes many of these properties into account in its operating system-specific tests for each requirement. Mobile operating systems contain built-in security features designed to provide protection from malicious behaviors, decrease the likelihood of vulnerabilities, and decrease the impact that would result from exploitation of vulnerabilities.

2.2 Enterprise Mobility Management

EMM solutions provide enterprises with a centralized capability to manage their mobile devices, including any security risks associated with them. These solutions (among other capabilities) can provision security policies to devices, provision devices with credentials to access enterprise resources, and monitor aspects of device state, including gathering an inventory of installed applications. In more recent years, EMM has also taken on the role of personal enablement with BYOD or COPE deployment options.

"Enterprise Mobility Management" is an umbrella term that describes the many services an EMM solution provides in addition to the security of the device (typically referred to as "Mobile Device Management"). These other services are often described as Mobile App Store (MAS), Mobile App Management (MAM), Mobile Content Management (MCM), Containerization, and MTD. The solutions vary widely depending on the EMM vendor's offerings and choice of terms. Typically, MAS refers to the ability to catalog and provision apps to an employee's device. MAM refers to the ability to control usage of apps on a device, whether this involves creating and enforcing whitelists/blacklists or limiting the hardware resources available to apps. MCM refers to the ability to host, edit, and restrict access to content available on mobile devices.

2.3 Mobile Threat Defense

MTD generally refers to the functions performed by endpoint detection and response (EDR) products that run on mobile devices in the form of a mobile app designed to detect and defend against security threats. MTD products often provide the ability to integrate with EMM systems. Integration enables MTD products to obtain device and app inventory information from EMM solutions, supplement the existing security capabilities of EMM systems, and utilize EMM systems for compliance enforcement as necessary.

MTD products generally advertise the ability to defend against device threats, app threats, and network threats. This report focuses on evaluating only the mobile app vetting capabilities of the products; analysis of the app threat capabilities of 'pure play' MTD systems and other MTD capabilities could be performed as future work.

Device threats detected by MTD products could include misconfigurations that increase the device's susceptibility to exploitation, such as enablement of Universal Serial Bus (USB) debugging capabilities or disablement of device lock screen authentication. Threats could also

include devices that are running out-of-date operating system versions that are susceptible to publicly known vulnerabilities. MTD products additionally may include capabilities to detect device exploitation attempts, whether successful or unsuccessful, or other anomalous behavior. The products may make use of built-in device integrity attestation capabilities (e.g., Android SafetyNet and/or Android keystore attestations) and security audit capabilities.

App threats detected by MTD products may include the types of threats discussed in section 3.1 and section 4 of this report, including detection of vulnerable, malicious, or privacy-violating apps. MTD products may also be able to detect sideloaded apps or other characteristics of apps that indicate elevated risk.

MTD products often include the ability to detect man-in-the-middle attack attempts on device network traffic, use of unprotected Wi-Fi networks, and use of Wi-Fi networks with known malicious activity (e.g., using data obtained through crowdsourcing). MTD products may also include Uniform Resource Locator (URL) filtering capabilities to prevent phishing attacks (not only through email or the web browser, but also through short message service (SMS) text messaging and even third-party text messaging apps).

Unlike EDR products running in traditional Windows PC environments, MTD products are generally subject to the same sandbox protections and restrictions as other third-party mobile apps. These sandbox restrictions may impair the ability of MTD products to examine some aspects of system state.

As discussed later in this report, HSSEDI found no one way to uniquely characterize the solutions tested based on the complete set of capabilities they possess. Some solutions only perform app vetting, while others can be characterized as either mobile security platforms with MTD and app vetting capabilities or as MTD solutions with app vetting capabilities. Moreover, some vendors may have more experience with implementing MTD than app vetting (or vice versa) because they originally created a platform with greater emphasis on one or the other. For the purposes of this report, all solutions tested are assumed to have app vetting capabilities and will be referred to as an "app vetting solution."

2.4 Continuous App Vetting and the Mobile Ecosystem

A continuous mobile app vetting system is part of the larger mobile ecosystem comprising on-premises and cloud services. Some of these components are outside the direct control of an organization, such as Mobile Network Operator and Wi-Fi networks that a device owner may use for apps that require connectivity to resources inside the enterprise security boundary. However, mobile enterprise and supporting cloud services from mobile operating systems vendor infrastructure add potential threat vectors for attackers to distribute malware or other harmful software to end users [2, 16].

An app vetting solution is used to its full capability when integrated with an EMM system. Also, as noted earlier in this report, some app vetting and MTD solutions have converged into a single platform. Taking these factors into account, Figure 3 depicts the larger mobile security ecosystem created in (Draft) NISTIR 8144, a NIST publication that outlines a catalogue of threats to mobile devices. The figure is modified from the original with the addition of a

representational app vetting and MTD system within an enterprise's access and mobility management systems.

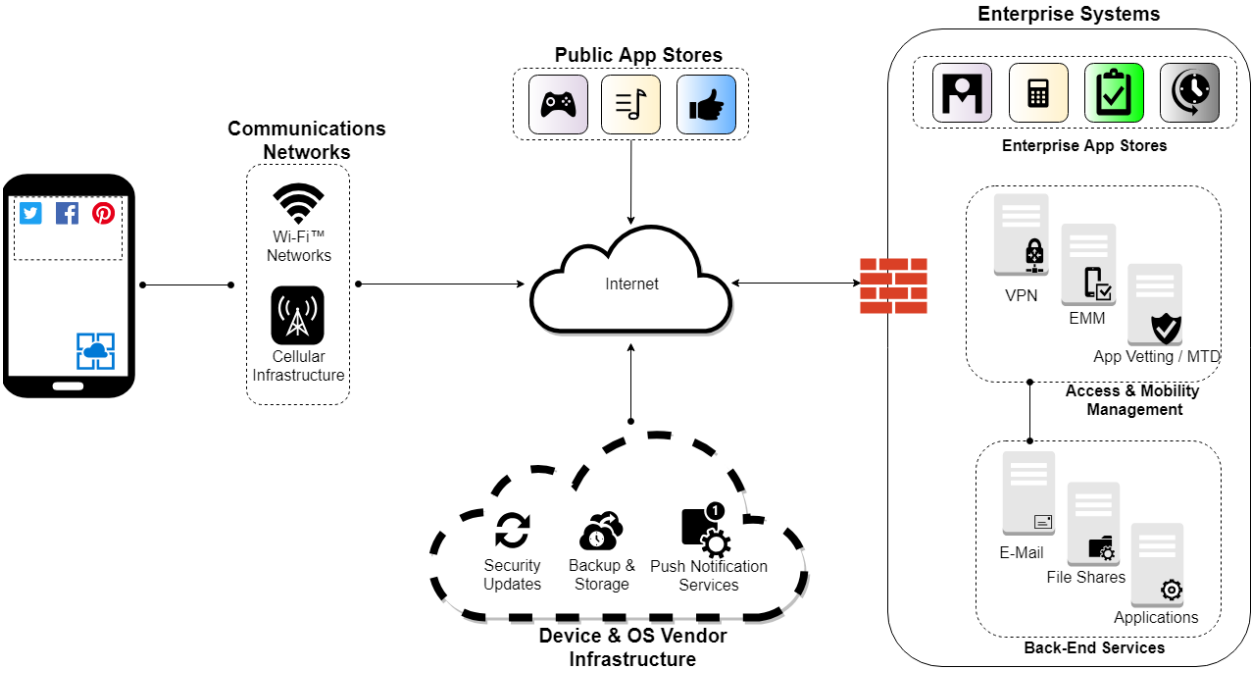

Figure 3: Mobile Ecosystem

Building on this architecture, in Figure 4 the general continuous app vetting process described earlier in this section is put into the context of the mobile security ecosystem. Note that the continuous app vetting process may have slight variations depending on the mobile operating system and the client that is used to gather app inventory on the endpoint device. In Step 2, the connection is secured and treats the communication network as untrusted, ensuring the confidentiality of the app inventory as it is communicated back to the enterprise. Further, in Step 4 an example remediation action could be restricting mobile VPN access to the back-end services until the security issue has been resolved.

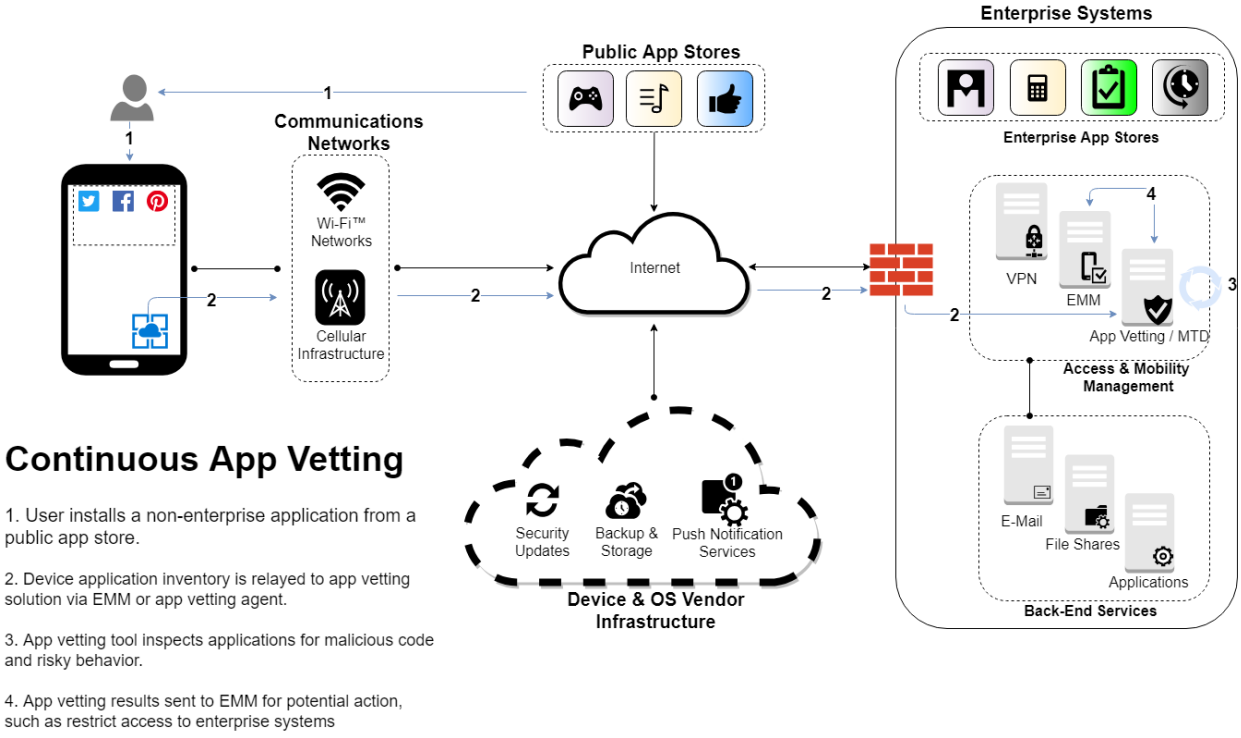

Figure 4: Continuous App Vetting - Mobile Ecosystem

3 Threat Rationale, Requirements, and Evaluation Criteria

This section identifies the rationale for the requirements and evaluation criteria using a threat-based assessment and mappings to security requirements where applicable.

3.1 Threat Rationale

The requirements and criteria that HSSEDI developed to evaluate the integration of EMM-app vetting solutions address threats related to the app security and EMM components independently, as well as threats to the integrated solution. Some threats are included in the most recent version of the MDM Agent EP v3.0, dated November 21, 2016; the Application Software PP v1.2, dated April 22, 2016; and Enterprise Security Management PPs for Identity and Credential Management (ICM), Policy Management (PM), and Access Control (AC), all dated October 24, 2013 [15, 11, 12, 13, 14].

1) Apps loaded onto a mobile device may include malicious or exploitable code, rendering them malicious or flawed. Administrators of an EMM or app vetting suite, or a mobile device user, may either inadvertently (through sideloading[3]) or as an adversary insert malicious code to compromise any of these systems [15].

   This threat applies to app security on the mobile devices themselves, which is addressed by the app vetting component of the EMM-app vetting solution, with policies enforced by the EMM component. The evaluation described in this report extends to analysis of the apps themselves by the app vetting component to detect malicious or exploitable code or vulnerabilities.

2) Unauthorized entities may intercept communications between the EMM solution and mobile devices to monitor, gain access to, disclose, or alter remote management commands. These entities may also intercept unprotected wireless communications between the mobile device and the enterprise to monitor, gain access to, disclose, or alter EMM and app vetting data [15].

   This threat applies to the integrated EMM-app vetting solution.

3) An attacker positioned on a communications channel or elsewhere on the network infrastructure may engage in communications with the app software or alter communications between the app software and other endpoints [11]. The app software might be running on the mobile device but can also refer to components of the EMM-app vetting solution as well as the entire solution. Attackers may monitor and gain access to data exchanged between the app and other endpoints [11]. An attacker can act through unprivileged software on the same computing platform on which the app executes. Attackers may provide maliciously formatted input to the app in the form of files or other local communications [11]. Additionally, an attacker may try to access sensitive data at rest [11].

[3] Sideloaded means the app was installed neither from the platform's sanctioned app store (e.g., Google Play Store, Apple App Store) nor from the EMM system that manages the device.

This threat applies to mobile apps, to the individual EMM and app vetting components of the EMM-app vetting solution, and to the integrated solution as a whole.

4) A careless administrator may create a policy that contains contradictory rules for enforcement by the integrated EMM-app vetting solution [14].

   This threat applies to policies created within the EMM and app vetting components of the integrated solution.

5) A malicious or careless user may suspend or terminate operation of the EMM and/or app vetting server, thereby preventing the integrated solution from enforcing its access controls or policies upon the environment or protected data [12]. In addition, a malicious or careless user may cause the mobile device and/or EMM-app vetting solution to lose connection to the source of its enforcement policies, adversely affecting access control behaviors [12].

   This threat applies to the individual EMM and app vetting components of the integrated solution as well as the fully integrated solution.

6) A malicious user could eavesdrop on network traffic to gain unauthorized access to the integrated EMM-app vetting solution's data, as well as to mobile device data [10].

   This threat applies to the EMM and app vetting components of the integrated EMM-app vetting solution as well as the full integrated solution.

7) A malicious user may falsify the identity of servers within the integrated EMM-app vetting solution and transmit false data that purports to originate from the solution to provide invalid data for EMM/app vetting deployment. Examples include issuing EMM commands to modify device configuration settings and weaken security or causing the app vetting solution to identify an app containing malware as a low-risk app and indicate that the EMM can deploy the app to managed devices. A malicious user could also falsify the identity of servers within the EMM/app vetting solution, giving the EMM false assurance that the solution is enforcing a policy on devices [12, 13].

   This threat applies to the EMM and app vetting components of the integrated EMM-app vetting solution.

8) The Google Play Store and Apple App Store perform their own app vetting and prevent malware from being hosted in their stores or remove apps identified as malicious. Although the vetting performed by Google and Apple does not satisfy government security criteria, it does provide a first level of defense against deliberate malware. However, apps found outside the platforms' stores are not vetted. Adversaries carrying out targeted attacks may seek to avoid detection by not publishing their apps in the mainstream app stores. If the EMM or app vetting solution detects an app installed on enterprise devices and does not find the app's binary in the official Android or iOS app store, this is a significant indicator of risk.

   This threat applies to app security on the mobile devices themselves, which is addressed by the app vetting component of the EMM-app vetting solution, with policies enforced by the EMM component.

9) App vetting systems commonly obtain and analyze binaries from the major app stores, but they may not have the ability to obtain binaries of sideloaded apps. Hence, those apps would not be analyzed, and risks associated with the app would not be detected or reported to the EMM solution or the app vetting analyst. The app vetting solution should and could report the presence of a sideloaded app and the failure to locate the app in the platform's official app store.

   This threat applies to the applications analyzed by the app vetting component of the EMM-app vetting solution.

10) If apps are not identified by some form of unforgeable identity (e.g., a cryptographic hash), then a malicious app may be able to disguise itself by impersonating the identifier of a legitimate app and appear as the legitimate app to the app vetting system and/or EMM system.

   This threat applies to the applications analyzed by the app vetting component of the EMM-app vetting solution.

3.2 Requirements

The solution requirements established by HSSEDI and addressed in this report are organized into five general categories, as listed below.

1. Solution characteristics:
   a. The solution shall support Android and iOS mobile platforms.
   b. The solution shall obtain an inventory of installed apps of the enterprise's user devices.
   c. The app vetting system and/or EMM solution shall uniquely identify each app installed on enterprise devices in such a way that the app identity cannot be spoofed. The solution shall correctly detect:
      i. Detect version numbers of apps.
      ii. A hash, certificate, or Software Identification (SWID) tag that is then compared to a known developer certificate or SWID.
   d. The solution shall perform periodic app security analysis either after installation or rule change or on a given schedule. On-demand scanning shall be performed within 1 hour (objective) and no more than 12 hours.
   e. The app vetting solution shall detect and report out-of-date apps.
   f. The solution shall identify and report the presence of sideloaded apps and report their detection to the EMM or EMM administrator for action.
   g. The app vetting solution shall obtain and analyze sideloaded app binaries. If on iOS, it shall flag sideloaded apps and provide a means to upload the app for analysis, as iOS does not allow for app binaries to be pulled off the device.
   h. The app vetting solution shall compare apps found on the device to known apps available in Google Play, the Apple App Store, and/or an enterprise app store that the organization uses.
2. Ability to assess general risks related to the reputation of the app and its developer:

a. The solution shall have the ability to determine if an app exists in the mainstream app store, and provide a measure of popularity determined by:
    i. Number of downloads
    ii. Average rating
    iii. Update history.
b. The solution shall have the ability to determine the developer of an app, as well as the number of other apps from that developer that are in a mainstream app store. The solution should provide a:
    i. Measure of popularity.
    ii. Listing of security issues found in other apps from the same developer.
c. The solution shall be able to identify and provide indications that the app is repackaged or counterfeit.

3. Ability to detect potentially exploitable security vulnerabilities or malicious/privacy-violating behaviors:
    a. The app vetting solution shall be able to satisfy the requirements identified in the NIAP Protection Profile for Application Software Version 1.2.
    b. The app vetting solution shall possess the ability to detect anomalous behavior of the app by some means of characterizing abnormal behavior (e.g., through machine learning or other techniques).
4. Ability to integrate seamlessly into an EMM solution:
    a. The app vetting solution shall interface with the EMM solution to deliver app vetting results.
        i. The app vetting solution shall interface with the EMM solution to trigger automated response actions based on these results.
    b. The app vetting solution shall provide a quantifiable measure of risk. It shall:
        i. Enumerate individual risks.
        ii. Specify their degree of risk.
        iii. Provide a consistent and well-documented methodology for calculating cumulative risk score.
    c. The app vetting solution shall provide risk measures that can be configured to allow for organization-specific risks or scores.
    d. The app vetting solution shall provide consistent app/user blacklist and whitelist policies where applicable.
    e. The app vetting solution shall provide policies that continue to be applied to devices even when there is no availability of the EMM or app vetting components of the integrated solution.
5. Security of the integrated app vetting/EMM solution itself:
    a. The integrated solution shall use secure protocols in any communications between the EMM and app vetting components (and vice versa).
    b. The integrated solution shall support mutual authentication.

c. If EMM administrator credentials are stored by the app vetting component, the app vetting solution shall provide a secure means of storing these credentials.
    i. The vendor shall document whether the app vetting system must store EMM administrator credentials and, if so, where these credentials are stored.
d. The EMM shall be able to satisfy the requirements in NIAP Mobile Device Management Version 3.0 as well as the Extended Package for MDM Agents.

3.3 Evaluation Criteria

The evaluation criteria developed by HSSEDI to assess solutions against the requirements include references to NIAP PP requirements, as appropriate.

1. Solution characteristics:
    a. Which mobile platforms are supported (e.g., Android, iOS, Windows)?
    b. How does the app vetting system obtain an inventory of installed apps? Does it use its own agent app to obtain the inventory or does it use the EMM agent, or both?[4]
    c. Can the app vetting system and/or EMM solution uniquely identify each app installed on devices in such a way that the app identity cannot be spoofed? For example, can the solution identify each app by a cryptographic hash or other unforgeable identity so that a malicious app cannot be disguised as a legitimate app and appear to be legitimate to an EMM or app vetting system?
        i. NIAP [PP_APP_V1.2]: FPT_TUD_EXT.1.6 (signed updates), FPT_IDV_EXT.1 (SWID tags, objective requirement) [11]

        NOTE: The Apple Mobile Device Management Protocol may limit the ability of the app vetting system and/or EMM solution to uniquely identify apps. It may therefore be possible for sideloaded apps to impersonate identifiers associated with legitimate apps to avoid detection.
    d. Does the app perform periodic app security analysis? Identify frequency or events for which this is performed.
    e. Can the app vetting system detect and report out-of-date apps (i.e., is there a newer version available)?
        i. NIAP [PP_APP_V1.2]: FPT_TUD_EXT.1.1 (application/platform shall check for application updates and patches) [11].
    f. Can the solution identify and report the presence of sideloaded applications?
        i. For sideloaded apps from the wrong app store, NIAP [PP_APP_V1.2]: FMT_MEC_EXT.1.1 [11].

        NOTE: Both Android and iOS provide the ability to sideload apps. Malicious apps are commonly sideloaded to avoid potential detection by an app store.

[4] On Apple iOS 11 and higher, apps (including an app vetting system's agent app) appear to be blocked from getting a list of other installed apps, even when sideloaded. Instead, Apple's MDM interface, which is used by EMM systems, must be used.

Sideloaded apps may also be more likely to contain vulnerabilities, as the app stores perform checks for common vulnerabilities and malicious content.

g. Can the app vetting system obtain and analyze sideloaded app binaries?
h. Does the app vetting system compare apps found on the device to known apps available in Google Play, the Apple App Store, and/or an enterprise app store that the organization uses?
i. If the app vetting solution detects a sideloaded app, can it report detection to the EMM or EMM administrator for action?

2. Ability to assess general risks related to the reputation of the app and its developer:
    a. Does the app exist in mainstream app stores? How popular is it (determined by number of downloads and ratings)?
    b. How many apps from the same developer exist in mainstream app stores? How popular are they? Have security issues been found in other apps from the same developer?
        i. NIAP/Common Criteria: AVA_VAN.1 (public vulnerability analysis) [11]
    c. Are there indications that the app is repackaged/counterfeit? For example, does another app with the same name but from a different developer exist in mainstream app stores and/or is it installed on a large number of mobile devices?
        i. NIAP [PP_APP_V1.2]: FPT_TUD_EXT.1.6 (signed updates), FPT_IDV_EXT.1 (SWID tags, objective requirement) [11]
3. Ability to detect potentially exploitable security vulnerabilities or malicious/privacy-violating behaviors:
    a. Can the app vetting solution assess whether the app attempts to update executable code after installation?
        i. NIAP [PP_APP_V1.2]: FPT_TUD_EXT.1.4 [11]
    b. Can the app vetting solution report that apps only establish network connections with sites that are absolutely essential for their purpose (i.e., no remote communications to malware sites)?
        i. NIAP [PP_APP_V1.2]: Parts of FDP_NET_EXT.1 are related [11]
    c. Can the app vetting solution report whether network communications use secure protocols (e.g., Hypertext Transfer Protocol Secure (HTTPS) vs. Hypertext Transfer Protocol (HTTP)) and any related security issues such as improper HTTPS/Transport Layer Security (TLS) certificate validation or hostname checking?
        i. NIAP [PP_APP_V1.2]: FTP_DIT_EXT.1, FCS_TLSC_EXT.1, FCS_TLSS_EXT.1, FCS_DTLS_EXT.1, FCS_HTTPS_EXT.1, FIA_X509_EXT.1, FIA_X509_EXT.2 [11]
        ii. NIAP [CPP_ND_V2.0]: FCS_IPSEC_EXT.1 [4]
    d. Can the app vetting solution detect anomalous behavior of the app by some means of characterizing abnormal behavior (e.g., through machine learning or other techniques)?
4. Ability to integrate seamlessly into an EMM solution:

a. Can the app vetting solution interface with the EMM system to deliver app vetting results and/or trigger automated response actions based on the results?
b. Does the app vetting and/or EMM solution provide a quantifiable measure of risk? Does it enumerate individual risks and their degree of risk as well as providing a cumulative risk measure score?
c. Are risk measures configurable to allow for organization-specific risks or scores?
d. How are app/user blacklist and whitelist policies enforced and, where applicable, how does the app vetting component interact with these policies?
    i. NIAP [PP_ESM_PM_v2.1]: FMT_MSA_EXT.5 (consistent security attributes) [14]
    ii. NIAP [PP_ESM_AC_V2.1]: FCO_NRR.2 (enforced proof of receipt) [12]
    iii. NIAP [EP_MDM_AGENT_V3.0]: FMT_POL_EXT.2 (trusted policy update) [7]
e. Do policies continue to be applied to devices even when there is no availability of the EMM or app vetting components of the integrated solution?
    i. NIAP [PP_ESM_AC_V2.1]: FPT_FLS_EXT.1 (failure of communications), FPT_FLS.1 (failure with preservation of secure state), FRU_FLT.1 (degraded fault tolerance), FPT_RPL.1 (replay detection) [12]

5. Security of the app vetting/EMM integration solution itself:
    a. How is the connection secured between the app vetting system and EMM system? Is a secure, encrypted protocol used (e.g., HTTPS/TLS with proper authentication of the server certificate)? Which entity connects to which? Is mutual authentication used?
        i. NIAP [EP_MDM_AGENT_V3.0]: FTP_ITC_EXT.1, FMT_POL_EXT.2 (trusted policy update), FAU_ALT_EXT.2 (agent alerts) [7]
        ii. NIAP [PP_MDM_V3.0]: FPT_ITT.1 (transfer between EMM and App Vetting), FIA_X509_EXT.1, FCS_TLSC_EXT.1, FCS_TLSS_EXT.1, FCS_HTTPS_EXT.1, FCS_DTLS_EXT.1 [15]
        iii. NIAP [CPP_ND_V2.0]: FCS_IPSEC_EXT.1 [4]
    b. Does the app vetting system have to store EMM administrator credentials? Where are these credentials stored?

4 Test Apps

HSSEDI used various Android and iOS apps to test solutions against the evaluation criteria listed above. Some of the apps were developed or modified by the testing team, and others were available in the Google Play Store or the Apple App Store. The following subsections describe each app, how it was used for testing, and the rationale behind its use.

4.1 Android Apps

4.1.1 Custom Apps

- "Custom Class Loader" – An app that attempts to download and update executable code after installation in order to simulate a vector for a potential zero-day attack. HSSEDI used it to test if the app vetting tools could detect if the executable code was updated after installation.
- The European Institute for Computer Anti-Virus Research "EICAR Anti-Virus Test (uk.co.extorian.EICARAntiVirusTest)" – An app that has a string/signature that can be detected by anti-virus software. This was used to test whether app vetting solutions can label apps as malicious based on anti-virus signatures.
- "Hello World (dev.trotter.android.GaC.1.0.apk)" – This app was used for testing whether a non-malicious non-market app uploaded to the enterprise app store is detected as sideloaded when downloaded by an Android device. *NOTE*: This app was originally published to the Play Store in 2010, but is currently not being offered, making it a nonmarket app.
- "Not The Blue Alliance" – A decompiled, modified, and repackaged version of The Blue Alliance app that is available in the Play Store, with the version number changed from 4.3.1 to 5.0.0 and several additional strings modified. The modified app was installed alongside the original app and was used to test if the app vetting solution detected the differences and to determine if the solution detected the modified version as potentially spoofed.
- "UploadDataApp" – An app that could be considered malicious. It uploads sensitive information to a remote server, including the user's location, microphone audio, received SMS messages, contact list entries, call log entries, inventory of all installed apps, and the device's International Mobile Subscriber Identity (IMSI), International Mobile Equipment Identifier (IMEI), and phone number. The app should register as malicious or unwanted when scanned by the app vetting tools. For one of the tests, the app was renamed to have the same package ID, display name, or both as those used in a popular market app (i.e., Facebook) to test if the tool could detect spoofing of the package ID or name.
- "VLC (modified)" – A decompiled, modified, and repackaged version of the VLC app that is available in the Play Store, with the version number changed and the light theme changed to the dark theme. The modified app was installed alongside the original app and was used to test if the app vetting solution detected the differences and to determine if the solution detected the modified version as potentially spoofed.
- "VLC (modified, vlc-original-mod-9-14-2018-1032p.apk)" – An alternate decompiled, modified, and repackaged version of the VLC app available in the Play Store. This version of

the app used the same package name as the original. Minor functionality was changed. Strings were modified from resources/res/values/strings.xml and resources/res/values/plurals.xml. The "about_text" value was changed in strings.xml, and both "albums_quantity" and "songs_quantity" values from plurals.xml were changed to "%d allowance" and "%d songs to sing". An if-statement for checking mobile platforms within the smali functions was removed. The modified app was installed alongside the original app and was used to test if the app vetting solution detected the differences and to determine if the solution detected the modified version as potentially spoofed.

- "VLC (modified, vlc-original-new-versionNumber.apk)" – An alternate decompiled, modified, and repackaged version of the VLC app available in the Play Store, with the version number changed. The modified app was installed alongside the original app and was used to test if the app vetting solution detected the differences and to determine if the solution detected the modified version as potentially spoofed.

4.1.2 Apps from the Play Store

- "Amazon Kindle" – An app that is known to communicate with ad servers. It was used to test if the app vetting tools could detect if apps were only communicating with necessary URLs.
- "Angry Birds" – An app that is known to send "Analytics Data to Flurry." This app was used to test whether custom app and remediation policies for specific app threats can be deployed in the app vetting solution, with policy violation notifications sent to the user and EMM administrator.
- "Chrome" – An app that runs Google's flagship Internet browser. It was used in multiple tests, including comparing apps by the same developer.
- "Chrome Canary" – An app that is updated nightly by Google. It was used to test if the app vetting tools would detect if apps installed on a device were out of date and could perform a timely scan of updated apps, including market apps.
- "Dropbox" – An app that uploads to the internet, but only communicates with URLs that are necessary for its purposes. It was used to test if the app vetting tools would mistakenly flag the app for communicating with unnecessary URLs.
- "Firefox" – An app that was used to test if the app vetting tool would detect an app downloaded from the flagship app store as a legitimate app.
- "Flipboard" – An app that was identified to use an ad network. It was used for multiple anomalous detection tests.
- "Forcepoint Trusted Access Mobile Client" – A NIAP-approved Android app used for testing an app's support of secure protocols and certificate standards.
- "F-Test" – An antivirus testing app that adapts the European Institute for Computer Anti-Virus Research (EICAR) file standard for testing antivirus software. It was used in certain tests to identify how solutions reacted to the installation of a malicious app.

- "Google Keep" – An app that uploads to the internet, but only communicates with URLs that are necessary for its purposes. It was used to test if the app vetting tools would mistakenly flag the app for communicating with unnecessary URLs.
- "Google+" – An app intended for interacting with community boards within the Google platform. It was used for checking software updates.
- "Hypori Client" – A NIAP-approved Android app used for testing an app's support of secure protocols and certificate standards.
- "IMDB" – The official app for the web service, identified as being a potential security risk due to extensive use of advertising protocols. It was used to test if app vetting services were able to detect usage of advertising Application Programming Interfaces (APIs).
- "Messenger" – An app used for messaging within Facebook. It was used in multiple tests and characterized as exhibiting a high privacy risk.
- "MITRE@Work" – An app intended for MITRE employees that transmits sensitive data and has been certified to be secure. It was used to test the app vetting services' ability to detect use of secure protocols and was also used as a negative test to ensure that the app vetting services did not over-flag safe apps.
- "NASA" – An app that was used for determining whether lower risk apps would be indicated as exhibiting little or no anomalous behavior.
- "Official NFL" – An app that was identified as posing a medium or high risk due to use of advertising libraries. It was used to test if the app vetting tool would return a quantifiable measure of risk.
- "SHAREIt" – An app that allows users to share various media files with other users. It is associated with high security and privacy risks as shown in multiple test cases.
- "Slack" – An app that uses secure protocols to communicate over the internet to perform messaging. It was used to test if the app vetting tools could identify if an app is using secure protocols. It was also uploaded to the EMM as an in-house app and was used to determine if an app downloaded via an EMM or means other than the flagship app store would be identified as sideloaded by the app vetting tool.
- "Tresorit" – An app intended for cloud-based storage. This app was used in several tests because of its emphasis on enhanced security and data encryption. It is expected to have a lower threat rating.

4.2 iOS Apps

4.2.1 Custom Apps

- "AcmeAirlines" – A MITRE-developed app that demonstrates a multitude of vulnerable or potentially malicious activities. For one of the tests, the app was renamed to have the same package ID, display name, or both as those used in a popular market app (i.e., Facebook) to

test if the app vetting tool could detect spoofing of the package ID or name. The AcmeAirlines app uses method swizzling[5] to modify its codebase at runtime and was used to test if the app vetting tools could detect if the executable code was updated after installation.

- "Facebook++" – Similar to Twitter++, an official app that has been decompiled, modified by a third party, and then recompiled. This app is distributed on the iOS hacking site iosninja.io, and can be installed on any non-jailbroken iOS device, where it purports to add functionality to the official Facebook app. This app uses a similar but not exact bundle ID as the official app and was used to check the abilities of app vetting services to detect spoofed or repackaged apps.
- "HelloWorld" – A blank app with no functionality. It was uploaded to and distributed by the EMM solutions. It was used to determine if an app downloaded via an EMM solution or means other than the flagship app store would be identified as sideloaded by the app vetting tool.
- "Twitter++" – An "improved" version of Twitter. It was downloaded from iosninja.io and appears to be a repackaged version of the Twitter app, signed with an enterprise developer certificate. It was sideloaded onto the test device alongside the official Twitter app, installed via the App Store. This was used to test if the app vetting solution detected the differences between the apps and to determine if the solution detected the modified version as potentially spoofed. It was also used to determine if the app vetting tools would identify it as being sideloaded.
- "YouTube (modified)" – A decompiled, modified, and repackaged version of the YouTube app that includes a popup to show that it has been modified. HSSEDI created this app by extracting a decrypted version of the official YouTube iOS app and using an open-source tool known as IPAPatch [18] to mix in custom code and install the app on a test device. It was installed alongside the original YouTube app and was used to test if the app vetting solution detected the differences and to determine if the solution detected the modified version as potentially spoofed.

4.2.2 Apps from the App Store

- "Amazon Kindle" – An app that is known to communicate with ad servers. It was used to test if the app vetting tools could detect if apps were only communicating with necessary URLs.
- "Chrome" – An app that is updated often. It was used to test if the app vetting tools would detect if apps installed on a device were out of date.
- "DuckDuckGo" – An app that was used to test whether new updates would be scanned in a timely manner (within 3 days).
- "Dropbox" – An app that uploads to the internet, but only communicates with URLs that are necessary for its purposes. It was used to test if the app vetting tools would mistakenly flag the app for communicating with unnecessary URLs.

[5] https://www.fireeye.com/blog/threat-research/2016/01/hot_or_not_the_bene.html

- "Facebook" – A social media app that was used to check if app updates were scanned in a timely manner.
- "Firefox" – An app that was used to test if apps downloaded from the flagship app store would be detected as legitimate apps by the app vetting tool.
- "Flipboard" – A news app that heavily uses advertising protocols and could have potential security risks. It was used to check if app vetting services were able to identify the use of advertising protocols in tested apps.
- "Forcepoint Trusted Access Mobile Client" – A NIAP-approved Android app used to test the support of secure protocols and certificate standards.
- "Google Docs" – An app that is updated often. It was used to test if the app vetting tools would detect if apps installed on a device were out of date.
- "Kdan PDF Reader – Document Expert" – An app that is known to use the AdMob Software Development Kit (SDK) and communicate with an ad server. It was used to test if the app vetting tools could detect if apps were only communicating with necessary URLs. This app is also known to send "Analytics Data to Flurry". The app was also used for testing whether custom app and remediation policies via specific app threats can be deployed in the app vetting solution, with policy violation notifications sent to the user and EMM administrator.
- "MITRE@Work" – An app intended for MITRE employees that transmits sensitive data and has been certified to be secure. It was used to test app vetting services' ability to detect use of secure protocols and was also used as a negative test to ensure that these services are not over-flagging safe apps.
- "Slack" – An app that uses secure protocols to communicate over the internet to perform messaging. It was used to test if the app vetting tools could identify if an app is using secure protocols.
- "SoundCloud" – An app used for streaming audio online. It was used to check if app updates were scanned in a timely manner.
- "Twitter" – An app that is known to "Connect to Twitter Social Network". This app was used to test whether custom app and remediation policies for specific app threats can be deployed in the app vetting solution, with policy violation notifications sent to the user and EMM administrator.

5 Tools Examined

To identify potential app vetting solutions for the evaluation, HSSEDI conducted a market analysis of continuous app vetting and MTD products. The market analysis involved a cursory examination of literature found on the websites of 27 products. The analysis tracked the vendors' claims of the products' ability to provide:

- Static and dynamic analysis
- Behavioral anomaly detection
- App threat intelligence
- App reputation analysis
- Mobile threat defense, including app, network, and device threats.

The analysis also covered the following solution characteristics:

- Works with Android, iOS and Windows
- Focuses on malicious behaviors, vulnerabilities, or both
- Uses open source or commercial software/services
- Integrates with EMM solutions
- Can be installed on-premises.

Based on the products' ability to perform in these categories, in the evaluation criteria deliverable [6] HSSEDI recommended further evaluation of 6 vendors' products to determine which app vetting and MTD products to include in the EMM-app vetting integration assessment.

HSSEDI integrated all the app vetting/MTD solutions with EMM A and B, with the exception of Solution 3, who does not integrate with EMM B and evaluated them according to the evaluation criteria and test cases described in Section 3.3.

6 Findings

This section summarizes the overall test results and overall results scoring.

6.1 Overall Test Results

HSSEDI developed and performed 43 test cases for each app vetting-EMM solution combination and each mobile platform over a test period of November 2018 thru May 2019. Table 1 shows a stoplight chart for Android test results comparing the test cases passing (green), partially passing (yellow/orange), failing (red), inconclusive (white), and not applicable (white).

Table 1: Overall Android Summary Test Results

Assessment Criteria	Test Case (Android)	Solution 1 + EMM A	Solution 1 + EMM B	Solution 2 + EMM A	Solution 2 + EMM B	Solution 3 + EMM A	Solution 4 + EMM A	Solution 4 + EMM B	Solution 5 + EMM A	Solution 5 + EMM B	Solution 6 + EMM A	Solution 6 + EMM B
1A. Platforms supported		1	1	2	3	2	2	1	2	1	2	2
1B. Obtain inventory of apps		1	1	1	1	1	1	1	1	1	1	1
1C Uniquely identify apps	1 Rename app	1	1	2	0	1	1	1	1	1	1	1
	2 Repackage app	3	3	2	0	3	1	1	1	1	1	1
	3 Modify version number	3	3	2	0	3	1	1	1	1	1	1
	4 Enterprise app store	2	2	2	0	3	2	2	3	3	2	2
1D Periodic app security analysis	1 Schedule/frequency/event.	0	0	0	0	1	0	0	0	0	1	1
	2 Verify schedule	1	1	3	0	1	1	1	1	1	3	3
	3 App updates	1	1	3	0	1	1	1	1	1	1	1
	4 Wait three days	1	1	3	0	1	1	1	1	1	1	1
	5 Apply custom rule	3	3	2	0	2	2	2	1	1	0	0
1E Detect out-of-date apps	1 Verify app is up-to-date	3	3	3	0	3	3	3	3	3	1	1
	2 Flag out-of-date apps	3	3	3	0	3	3	3	3	3	1	1
1F Identify sideloaded apps	1 Sideloading	2	2	2	0	1	1	3	2	1	1	1
	2 Enterprise app store sideloading	3	3	3	0	3	3	1	3	3	3	3
1G Analyze sideloaded applications		1	1	3	0	2	1	1	1	1	2	2
1H Compare apps in app stores		3	3	3	0	1	1	1	3	3	3	3
1I Act on detection of sideloaded app	Tested with 1F, admin notification	3	3	3	0	2	1	3	1	1	1	1
	Tested with 1C, not an app store app	3	3	2	0	1	1	1	1	1	1	1
	Tested with 1C, app from app store	1	1	2	0	2	2	2	1	1	1	1

Assessment Criteria	Test Case (Android)	Solution 1 + EMM A	Solution 1 + EMM B	Solution 2 + EMM A	Solution 2 + EMM B	Solution 3 + EMM A	Solution 4 + EMM A	Solution 4 + EMM B	Solution 5 + EMM A	Solution 5 + EMM B	Solution 6 + EMM A	Solution 6 + EMM B
2A App exist in mainstream app stores. App popularity.	Tested with 1C, app popularity	3	3	3	0	2	3	3	3	3	1	1
2B Other apps from developer, popularity and issues identified.	1 Multiple apps from developer	1	1	3	0	3	3	3	3	3	1	1
	2 Popularity of other apps	2	2	3	0	3	3	3	3	3	1	1
	3 Other app issues identified	3	3	3	0	3	3	3	3	3	2	2
2C Repackaged or counterfeit	Tested with Test 2, detect repackage	3	3	2	0	3	1	1	3	3	1	1
3A Dynamic code execution		3	3	1	0		1	1	1	1	1	1
3B Report network connections	1 Report use of ad network	1	1	1	0	2	1	1	1	1	1	1
	2 Reports necessary communication	1	1	1	0	1	1	1	1	1	1	1
3C Improper use of networking protocols	1 Identify secure protocols	2	2	1	0	1	1	1	1	1	1	1
	2 NIAP communication requirements	3	3	2	0	1	1	1	2	2	1	1
3D Detect anomalous behavior	1 Detect anomalous behavior	2	2	1	0	1	2	1	1	1	1	1
	2 Ad network anomalous behavior	2	2	1	0	3	2	1	3	3	1	1
	3 Anomalous behavior false positive	1	1	1	0	1	1	1	1	1	1	1
4A EMM app vetting results/response		1	1	3	0	1	2	2	1	1	3	3
4B Quantifiable measure of risk		2	2		0	1	2	2	2	2	2	2
4C Configurable risk measures		1	1	1	0	2	2	2	2	2	1	1
4D Blacklist/whitelist		2	2	2	0	1	1	1	2	2	2	2
4E EMM and app vetting solution availability	1 EMM/app vetting solution availability	1	3	1	0	1	1	2	1	3	1	3
	2 Solution availability, modification	1	1	1	0	1	1	1	1	1	1	1
5A Solution secure connection	1 (part 1) Solution secure connection	1	1	1	1	1	1	1	1	1	1	1
	1 (part 2): NIAP tests	1	1	1	1	2	1	1	2	2	2	2
	Test 2 (part 2): More NIAP tests	0	0	0	0	0	0	0	0	0	0	0
5B EMM administrator credentials		1	1	1	1	1	1	1	1	1	1	1

Table 2 shows the comparable results for iOS.

Table 2: Overall iOS Summary Test Results

Assessment Criteria	Test Case (iOS)	Solution 1 + EMM A	Solution 1 + EMM B	Solution 2 + EMM A	Solution 2 + EMM B	Solution 3 + EMM A	Solution 4 + EMM A	Solution 4 + EMM B	Solution 5 + EMM A	Solution 5 + EMM B	Solution 6 + EMM A	Solution 6 + EMM B
1A. Platforms supported		1	1	2	3	2	2	1	2	1	2	2
1B. Obtain inventory of apps		1	1	1	1	1	1	1	1	1	1	1
1C Uniquely identify apps	1 Rename app	3	3	2	0	1	3	3	2	1	1	1
	2 Repackage app	3	3	3	0	3	3	3	2	2	1	1
	3 Modify version number	3	3	3	0	3	3	3	3	3	1	1
	4 Enterprise app store	3	3	2	0	3	2	2	1	1	2	2
1D Periodic app security analysis	1 Schedule/frequency/event.	0	0	0	0	1	0	0	0	0	1	1
	2 Verify schedule	1	1	3	0	1	2	2	1	1		
	3 App updates	1	1	3	0	1	3	3	1	1	1	1
	4 Wait three days	1	1	3	0	1	3	3	1	1	1	1
	5 Apply custom rule	1	1	3	0	2	2	2	1	1	0	0
1E Detect out-of-date apps	1 Verify app is up-to-date	3	3	3	0	1	3	3	3	3	1	1
	2 Flag out-of-date apps	3	3	3	0	2	3	3	3	3	1	1
1F Identify sideloaded apps	1 Sideloading	1	1	3	0	1	3	3	3	1	1	1
	2 Enterprise app store sideloading	3	3	3	0		1	1	1	3	3	3
1G Analyze sideloaded applications		3	3	3	0	2	3	3	3	3	2	2
1H Compare apps in app stores		3	3	3	0	1	1	1	3	3	3	3
1I Act on detection of sideloaded app	Tested with 1F, admin notification	1	1	3	0	2	3	3	1	1	1	1
	Tested with 1C, not an app store app	1	1	2	0	3	3	3	3	2	1	1
2A App exist in mainstream app stores. App popularity.	Tested with 1C, app from app store	1	1	2	0	2	2	2	2	2	1	1
	Tested with 1C, app popularity	3	3	3	0	2	3	3	3	3	1	1
2B Other apps from developer, popularity and issues identified.	1 Multiple apps from developer	1	1	3	0	3	3	3	3	3	1	1
	2 Popularity of other apps	2	2	3	0	3	3	3	3	3	1	1
	3 Other app issues identified	3	3	3	0	3	3	3	3	3	2	2

Assessment Criteria	Test Case (iOS)	Solution 1 + EMM A	Solution 1 + EMM B	Solution 2 + EMM A	Solution 2 + EMM B	Solution 3 + EMM A	Solution 4 + EMM A	Solution 4 + EMM B	Solution 5 + EMM A	Solution 5 + EMM B	Solution 6 + EMM A	Solution 6 + EMM B
2C Repackaged or counterfeit	Tested with Test 2, detect repackage	3	3	3	0	3	3	3	3	3	1	1
3A Dynamic code execution		1	1	1	0	3	1	1	1	1	3	3
3B Report network connections	1 Report use of ad network	1	1	1	0	2	1	1	3	3	1	1
	2 Reports necessary communication	1	1	1	0	1	1	1	3	3	1	1
3C Improper use of networking protocols	1 Identify secure protocols	3	3	1	0	1	1	1	1	1	1	1
	2 NIAP communication requirements	3	3	2	0		2	2	2	2	2	2
3D Detect anomalous behavior	1 Detect anomalous behavior	1	1	1	0	1	1	1	1	1	1	1
	2 Ad network anomalous behavior	2	2	1	0		1	1	3	3	1	1
	3 Anomalous behavior false positive	1	1	1	0	1	1	1	1	1	1	1
4A EMM app vetting results/response		1	1	3	0	1	2	2	1	1	3	3
4B Quantifiable measure of risk		2	2	1	0	1	2	2	2	2	2	2
4C Configurable risk measures		1	1	1	0		2	2	1	1	1	1
4D Blacklist/whitelist		2	2	2	0	1	1	1	2	2	2	2
4E EMM and app vetting solution availability	1 EMM/app vetting solution availability	1	3	1	0	1	1	1	1	3	1	3
	2 Solution availability modification	1	1	1	0		3	3	3	1	1	1
5A Solution secure connection	1 (part 1) Solution secure connection	1	1	1	1	1	1	1	1	1	1	1
	1 (part 2): NIAP tests	1	1	1	1	2	1	1	2	2	2	2
	Test 2 (part 2): More NIAP tests	0	0	0	0	0	0	0	0	0	0	0
5B EMM administrator credentials		1	1	1	1	1	1	1	1	1	1	1

HSSEDI observed several strengths and drawbacks common to the offerings. All offerings were able to satisfy the app vetting tests adequately with varying levels of detail in the analysis and were able to analyze network communication by the app and output a comprehensive, easy-to-read app threat report. Most of the tested services were capable of obtaining an inventory of apps installed on devices and of re-scanning apps in a timely fashion when receiving updates, with high pass rates for tests 1B (obtain inventory of apps) and 1D.3 (periodic app security analysis/app updates). The services also generally performed well in test cases dealing with

suspicious network traffic, with a large majority passing test cases in 3B (report network connections), 3C (improper use of networking protocols), and 3D (detect anomalous behavior).

The services were mostly unable to perform reputation analysis, with most failing test cases 2A and 2B. All offerings either incorrectly labeled custom, non-market apps downloaded from the enterprise app store as sideloaded or failed to detect a sideloaded app in some way. Additionally, detection of spoofed and sideloaded iOS apps was a weak point, almost certainly due to iOS platform restrictions of such ability. The testing team also encountered difficulties in ensuring that the EMM solution enforced compliance policies (test case 4A) linked to threats detected by the app vetting solution, specifically when using EMM A. Lastly, few of the solutions were able to report the presence of out-of-date apps (test case 1E), a feature that is sorely lacking from most of these tools.

6.2 Overall Results Scoring

HSSEDI tested products in the first quarter of fiscal year 2019 and provided feedback to the vendors. Some responded by demonstrating new product features or upgrades, resulting in HSSEDI rerunning some test cases for those products.

HSSEDI quantified the testing results to better understand how the solutions performed across platforms and EMM solutions. The scoring awarded two points for each test passed and one point for each test passed partially. The points were tabulated and recorded by platform in Table 3. The combined EMM A scores had a range of 79 – 131 and an average score of 101. The combined EMM B scores had a range of 16 – 127 and an average score of 86.

Table 3: Overall Results Scoring

	Solution 1 + EMM A	Solution 1 + EMM B	Solution 2 + EMM A	Solution 2 + EMM B	Solution 3 + EMM A	Solution 4 + EMM A	Solution 4 + EMM B	Solution 5 + EMM A	Solution 5 + EMM B	Solution 6 + EMM A	Solution 6 + EMM B
Android	46	44	42	8	53	59	59	52	52	67	65
iOS	50	48	37	8	52	39	40	44	45	64	62
Combined by Platform	96	92	79	16	105	98	99	96	97	131	127

By this metric, Solution 6 outpaced the rest of the solutions with a top EMM A score of 131 and a top EMM B score of 127, followed by a tight group consisting of Solution 1, Solution 3, Solution 4, and Solution 5 in the ~100-point range. However, it is important to look beyond the

overall scores for each service and consider their individual strengths and weaknesses, as this scoring methodology is just one way to interpret results.

One interesting observation to note is that most of the solutions fared better with Android than iOS; whereas the opposite was true with vendor Solution 1 which performed better with iOS. There was also a slight favoring of EMM A over EMM B, but the difference appears negligible.

7 Recommendations and Conclusions

7.1 Improvements to Functionality

Based on the results of the testing and evaluation, HSSEDI recommends the following vendor actions to improve integration between current EMM and app vetting solutions:

- Implement comparison tools for reputation, popularity, and vulnerability analysis across apps from the same developer, including different versions of the same app. The app vetting tools in this evaluation did not provide an interface to perform reputation comparison between apps.
- Implement functionality for reporting potential threat violations from the app vetting solution to the EMM solution, notifying both the user of the device and the EMM administrator. Allow remediation actions to be enforced and implement additional remediation actions that correspond to relevant threats. For example, if an app is requesting access to the microphone, turn off the microphone.
- Implement automatic, timely scanning of previously unscanned, nonmarket apps as well as recent updates to market apps installed on mobile devices. App scans for both non-market and market apps should complete within hours, not days or weeks. Out-of-date apps should also be reported, with an ability to perform escalation actions to remediate the problem.
- Properly document the setup required to integrate EMM and app vetting solutions so that the process can be completed with minimal vendor input. Administrators should be able to initiate the EMM and app vetting integration according to vendor documentation without needing to contact technical support.
- Perform more robust testing of app vetting solutions to ensure that EMM and app vetting integration is fully functional when new versions of the EMM solution are released. App inventory lists reported by the app vetting solution should be accurate and consistent with the EMM reports. EMM vendors should perform similar testing to ensure that integration with app vetting solutions is fully functional when new versions of the app vetting tool are released. They should properly document any inconsistencies and limitations in functionality across different versions.
- Implement network protocol and certificate standard detection for apps and document the result in a separate, organized table. This includes, but is not limited to, the use of TLS, Datagram Transport Layer Security (DTLS), Internet Protocol Security (IPSec), HTTPS, and X.509 certificates, as well as any cryptography suites or algorithms used that can be reported.
- Ensure that app vetting solutions properly detect apps that are downloaded and installed on the device from sources other than market or enterprise app stores as sideloaded and report

them to the EMM solution. For example, market apps that are copied onto the device, rather than downloaded from an approved market or enterprise app store, should be labelled as sideloaded. Conversely, custom, non-market apps downloaded from the enterprise app store should not be labelled as sideloaded.

7.2 Organization-Specific Recommendations

HSSEDI found that no single integrated product implements all security-relevant capabilities well. The EMM and app vetting solutions in this evaluation exhibited different strengths and weaknesses, which can greatly affect organizational decisions on the solution that best meets their needs. For example, while Solution 6 holds a clear lead in overall score, it possesses a number of critical weaknesses, such as an inability to trigger responses in an EMM solution that prevent it from being a clear favorite. Therefore, HSSEDI recommends that departments/agencies review and understand the strengths and limitations of each tool combination and select the EMM and app vetting solution that best fits their needs and desired capabilities.

EMM and app vetting integration is still an emerging capability; the technology is still maturing. Vendors are actively developing new features and improving their offerings. Recommendations from the .gov Cybersecurity Architecture Review (.govCAR) of mobile cybersecurity architecture and the Continuous Diagnostics and Mitigation program will likely play a role in the maturation of this technology as well.

Despite their limitations, the top solutions from this evaluation showed an ability to mitigate many of the risks mobile apps present to the enterprise. Further, the vendors of the tools tested by HSSEDI have all indicated that they will improve integration support and other functional capabilities identified in this evaluation. Given the rapidly changing solution space, departments/agencies can still improve their overall security posture by employing these capabilities.

7.3 Conclusion

This report documented a continuous approach to mobile app vetting by integrating app vetting tools with EMM and exploring non-traditional app vetting approaches such as app threat intelligence. This approach can be used by enterprises to improve their overall security posture while allowing employees freedom to use apps to conduct business and accomplish the organization's mission. It provided market analysis of mobile app vetting tools and their capabilities to determine market leaders who were then evaluated via a threat-based assessment to determine their ability to identify risks to the enterprise and mitigate them via EMM. The findings suggest that there are solutions which possess the requisite app vetting capabilities (whether the solution identifies itself as an *app vetting* or *MTD* tool) to help secure the enterprise, however there is room for improvement. The individual solution evaluation results have been shared with their respective vendors and, in some cases, improvements have already been made, or identified in vendors' product road maps. It is HSSEDI's hope that this will result in a more seamless integration capability that will improve the overall security posture of departments' and agencies' mobile enterprises.

Future work is needed to evaluate the capabilities of MTD not included in this evaluation, namely, device and network security capabilities. Further exploration of how a continuous app vetting approach fits in with the CDM program is warranted as well. Lastly, while this report explored app vetting capabilties and their integration with EMM, it did not provide a recommended configuration of such tools to identify mobile app risks and their proposed mitigations. Further work is needed to explore the ability to apply appropriate mitigations on a per-app basis and how best to apply them.

List of Acronyms

Acronym	Definition
AC	Access Control
API	Application Programming Interface
ATT&CK	Adversarial Tactics, Techniques, and Common Knowledge
CC	Common Criteria
CCRA	Common Criteria Recognition Agreement
CDM	Continuous Diagnostics and Mitigation
COTS	Commercial Off-The-Shelf
CVSS	Common Vulnerability Scoring System
DHS	Department of Homeland Security
DTLS	Datagram Transport Layer Security
EDR	Endpoint Detection and Response
EICAR	European Institute for Computer Anti-Virus Research
EMM	Enterprise Mobility Management
EP	Extended Package
FFRDC	Federally Funded Research and Development Center
GPS	Global Positioning System
HSSEDI	Homeland Security Systems Engineering & Development Institute
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ICM	Identity and Credential Management
IMEI	International Mobile Equipment Identifier
IMSI	International Mobile Subscriber Identity
IPSec	Internet Protocol Security

Acronym	Definition
IT	Information Technology
JSON	JavaScript Object Notation
MAM	Mobile App Management
MAS	Mobile App Store
MCM	Mobile Content Management
MDM	Mobile Device Management
MTD	Mobile Threat Defense
MTP	Mobile Threat Protection
NIAP	National Information Assurance Partnership
NIST	National Institute of Standards and Technology
NSA	National Security Agency
OWASP	Open Web Application Security Project
PDF	Portable Document Format
PM	Policy Management
PP	Protection Profile
QA	Quality Assurance
R&D	Research and Development
S&T	Science and Technology Directorate
SAR	Security Assurance Requirement
SDK	Software Development Kit
SFR	Security Functional Requirement
SMS	Short Message Service
SSL	Secure Sockets Layer
SWID	Software Identification
TLS	Transport Layer Security

Acronym	Definition
UI	User Interface
URL	Uniform Resource Locator
USB	Universal Serial Bus
VPN	Virtual Private Network

List of References

1. "Banking Trojan attacks European users of Android devices." Dr. Web Anti-virus. November 16, 2018. [Accessed December 6, 2018]

2. C. Brown, S. Dog, J. Franklin, N McNab, S. Voss-Northrop, M. Peck, B. Stidham. *Assessing Threats to Mobile Devices & Infrastructure*. (DRAFT) NISTIR 8144. NIST, September 2016.

3. C. Northern, M. Peck. *Analyzing the Effectiveness of App Vetting Tools in the Enterprise: Recommendations to the Army Training and Doctrine Command*. MITRE Technical Report, MTR160242. The MITRE Corporation, August 22, 2016.

4. *Collaborative Protection Profile for Network Devices Version 2.0*. National Information Assurance Partnership. Protection Profile CPP_ND_V2.0. November 5, 2017. [Accessed March 29, 2018].

5. D. Goodin. "22 apps with 2 million+ Google Play downloads had a malicious backdoor." Ars Technica. December 6, 2018. [Accessed December 7, 2018]

6. *Evaluation Criteria for Evaluation of Enterprise Mobility Management and App Vetting Solution Integration.* HSSEDI Deliverable to DHS S&T. April 9, 2018.

7. *Extended Package for Mobile Device Management Agents Version 3.0.* National Information Assurance Partnership. Protection Profile EP_MDM_AGENT_V3.0. November 21, 2016. [Accessed March 29, 2018].

8. G. Bell, D. Keppler, M. Peck, C. Northern, C. Ryersen. *Secure Enterprise Access and Personal Enablement of Mobile Devices: Final Report*. MITRE Technical Report, MTR150360. The MITRE Corporation, September 2015.

9. Lukas Stefanko. "Scam iOS apps promise fitness, steal money instead." ESET. December 3, 2018. Available: [Accessed December 6, 2018]

10. MITRE. "Application Vetting," *ATT&CK*. The MITRE Corporation, 2018. [Accessed March 29, 2018].

11. *Protection Profile for Application Software Version 1.2.* National Information Assurance Partnership. Protection Profile PP_APP_V1.2. April 22, 2016. [Accessed March 29, 2018].

12. *Protection Profile for Enterprise Security Management–Access Control Version 2.1.* National Information Assurance Partnership. Protection Profile PP_ESM_AC_V2.1. November 21, 2013. [Accessed March 29, 2018]

13. *Protection Profile for Enterprise Security Management – Identity and Credential Management Version 2.1.* National Information Assurance Partnership. Protection Profile PP_ESM_ICM_V2.1. November 21, 2013. [Accessed March 29, 2018].

14. *Protection Profile for Enterprise Security Management – Policy Management Version 2.1.* National Information Assurance Partnership. Protection Profile PP_ESM_PM_V2.1. November 21, 2013. [Accessed March 29, 2018].

15. *Protection Profile for Mobile Device Management Version 3.0.* National Information Assurance Partnership. Protection Profile PP_MDM_V3.0. November 21, 2016. Available: [Accessed March 29, 2018].

16. *Study on Mobile Device Security.* Department of Homeland Security Science and Technology Directorate. April 2017.

17. "What Is NIAP/CCEVS?" *National Information Assurance Partnership.* National Information Assurance Partnership.[Accessed March 29, 2018].

18. Wu Tian. "IPAPatch." *GitHub*. 2017. [Accessed October 3, 2018]