March 2014 Federal Mobile Computing Summit Collaboration Session Summary

Dan Mintz
AMARC

Dr. Tamer Nadeem
Old Dominion University

Patrick Benito, Drew Buttner, Marie Collins, Bob Martin, Mike Peck, Ganu Kini
The MITRE Corporation

Approved for Public Release; Distribution Unlimited. 14-1455

Executive Summary

The Federal Mobile Computing Summit took place on March 5th and 6th, 2014. The Summit included a set of MITRE-Advanced Mobility Academic Research Center (AMARC) led Collaboration Sessions that allowed industry, government, academic, and federally funded research and development center (FFRDC) representatives an opportunity to collaborate, discuss the main challenge areas in mobile application security, and provide recommendations, and in some cases potential solutions, for key challenge areas. The discussions were government focused with the objective of refining gaps, and identifying features of potential solutions or frameworks.

The Collaboration Sessions covered four key mobile application security topics:

- Mobile Application Threats and Attack Patterns
- Mobile Application Vetting
- Mobile Application Protection Mechanisms
- Cyber Intelligence

Several key themes emerged as a result of the four Collaboration Sessions:

- There are a number of promising efforts ongoing within government mobile application security space that could be more effective if they were integrated.
- A robust, integrated Federal wide cyber intelligence program is critical to ensure agencies and commercial IT providers can adapt to new and emerging threats.
- The academic community is a ripe, largely untapped resource to use to help devote research into some key challenges.

This white paper summarizes the results of the Collaboration Sessions and provides detailed actionable recommendations for government and academia, which are summarized in the table below.

Extend Cyber Observable Expressions to Include Observables Associated with Mobile Threat Data
• The Federal CIO Council Information Security and Identity Management Committee's (ISIMC) Mobile Technology Tiger Team should work with key federal agencies to define processes to develop and disseminate cyber intelligence to key mobile ecosystem participants.
Improve the Mobile Application Information in Common Attack Pattern Enumeration and Classification (CAPEC)
• The Federal CIO Council ISIMC Mobile Technology Tiger Team should champion the improvement of CAPEC by adding in more mobile application related information, using the draft attack patterns developed during the Collaboration sessions as a start.
Develop Federal Level Mobile Application Security Requirements Guidelines
• The Federal CIO Council ISIMC Mobile Technology Tiger Team should champion and identify an owning government agency to create and maintain a Federal level mobile application SRG, based on DISA's SRG model.
Create a Central Clearing House for Mobile Apps
• The Federal CIO Council ISIMC Mobile Technology Tiger Team should work to identify a central government clearinghouse that can make the appropriate investments to specialize in application vetting.
Create an Integrated Advisory Capability Across FFRDC/Academia/Commercial
• The Federal CIO Council ISIMC Mobile Technology Tiger Team should champion a Integrated Advisory Capability to broker discussions between academia, industry, and government through targeted events and special initiatives.

Table of Contents

1 Introduction

The Federal Mobile Computing Summit took place on March 5th and 6th, 2014. The Summit included a set of MITRE-Advanced Mobility Academic Research Center (AMARC) led Collaboration Sessions that allowed industry, government, academic, and federally funded research and development center (FFRDC) representatives an opportunity to collaborate, discuss the main challenge areas in mobile application security, and provide recommendations, and in some cases potential solutions, for key challenge areas. The discussions were government focused and at a high-level, not addressing any specific solution and only identifying features of potential solutions or frameworks.

As part of the Collaboration Sessions, MITRE & AMARC invited academia to participate in each of the four challenge areas, and asked them to identify courses of action to be taken to enable improved government and industry collaboration with academic institutions. The academic participants focused on how they could build expertise in these technology areas, as well as address the specific challenge areas by changes in curricula, research, and outreach opportunities related to the challenges.

This white paper summarizes the results of the Collaboration Sessions and identifies suggestions and recommendations for government and academia while identifying crosscutting issues that tie between the different challenge areas. It also proposes a community built around government and industry collaboration with academia to leverage potentially previously untapped academic resources. The proposed community will be fostered by MITRE and the AMARC to enable communications between the different participating communities. The outcomes of this community include:

- Academia produces higher quality, better-prepared, and "industry-ready" graduates for hire.
- Government leverages graduate and undergraduate level research to help solve critical mobility challenges
- Government organizations have an integrated research and advisory capability made up of commercial companies, academic institutions, and federally funded research and development centers (FFRDC)

2 Collaboration Session Overview

MITRE and AMARC created an outreach and collaboration process to crowd-source the development of recommendations to key government challenges. This process focuses on soliciting input from a diverse group of participants and exploits their diverse backgrounds, points of view, and skillsets to create new, novel, and innovative recommendations to key problems.

Each MITRE-AMARC Collaboration Session was a focused and moderated discussion between government, industry, and academic representatives centered on a mobile application security challenge area.

The challenge areas for mobile application security as follows:

- Mobile Application Threats and Attack Patterns
- Mobile Application Vetting
- Mobile Application Protection Mechanisms
- Cyber Intelligence

Participants discussed current problems, gaps in current and planned work programs, potential solutions, and ways forward for each of the challenge areas. The following section of the whitepaper outlines the goals, outcomes and summary of each of the four collaboration sessions.

3 March 2014 Federal Mobile Computing Summit Collaboration Sessions on Mobile Application Security

3.1 "Mobile Application Threats and Attack Patterns" Session

The mobile application threats and attack patterns collaboration session aimed to identify a comprehensive list of known attack patterns pertaining to mobile applications and related sensitive data. This list is foundational in understanding useful protection mechanisms, preforming effective application vetting, and sharing cyber intelligence. In addition, the collaboration session looked to build a taxonomy to help group individual attack patterns to support a more efficient understanding of the relationships amongst the different patterns.

An attack pattern is an abstraction mechanism for helping describe how an attack is executed. Each pattern defines a challenge that an adversary may face, provides a description of the common technique(s) used to meet the challenge, and presents recommended methods for mitigating an actual attack. Attack patterns help categorize attacks to provide a coherent way of teaching designers and developers how their systems may be attacked and how they can effectively defend them. The Department of Homeland Security (DHS) Common Attack Pattern Enumeration and Classification (CAPEC™)[1] project provides a publicly available formal list of known attack patterns. CAPEC is a free, publicly available resource that provides attack pattern information to the community to enhance security throughout the software development lifecycle and to support the needs of developers, testers, and educators. An "attack pattern" is an abstraction mechanism for helping describe how an attack against vulnerable systems or networks is executed. Each pattern defines a challenge that an attacker may face, provides a description of the common technique(s) used to meet the challenge, and presents recommended methods for mitigating the attack.

[1] http://capec.mitre.org/

3.1.1 Session Goals

- Review the current list of known mobile application threats and attack patterns as defined in CAPEC.
- Propose additional and emerging items to be added to the list, focusing on those most important to consider when vetting an application.
- Develop an initial taxonomy that groups related mobile application threats and attack patterns.

3.1.2 Session Summary

The collaboration session focused on the different threats and attack patterns that are relevant to vetting mobile applications. Any available mobile hardware and operating systems was considered and were deemed appropriate for the discussion. An initial grouping was put together to help lead the discussion. The grouping was as follows:

1) Adversary has physical access to a device.
2) Adversary tricks user into executing a malicious application.
3) Adversary interacts with a target application through a malicious application.
4) Adversary interacts directly with a target application.
5) Adversary communicates with a target application's user.
6) Adversary has access to a target application's supply chain.
7) Adversary manipulates network traffic.

The above groupings made apparent a unique property in terms of application vetting. Some of the groups would be mitigated by vetting that an application had specific mitigations in place, while other groups would be mitigated by vetting that an application wasn't malware. A third type was also seen where vetting would make sure that certain weaknesses in the application did not exist. It is important to distinguish these three flavors of application as each may require a different approach used to vet the application.

Discussion within each grouping centered on the current list of known attack patterns. Each was reviewed and the associated negative technical impact was identified. Those that were already part of CAPEC were labeled as such, and those needing CAPEC entries were noted. Additional attack patterns were added to the list as appropriate.

Throughout the discussion, it became apparent that care has to be taken not to lose focus of the device level protections when talking about securing mobile applications. Today's mobile operating systems have a number of built in protections that applications can leverage and it is important to account for these when vetting mobile applications.

The full list of mobile attack patterns that came out of this session can be found in Attachment A. The CAPEC initiative was also identified as an appropriate forum to continue discussions about mobile attack patterns and to hold the list of patterns.

Discussion then shifted towards the taxonomy and how attack patterns can be related to each other in an attempt to enhance usability. CAPEC currently defines a view around

the method of attack. A second view should be considered that focuses on the countermeasures used to mitigate attacks. By grouping patterns according to countermeasures, an organization can easily see the potential trade-offs associated with certain choices. They can also focus their attention on those patterns with mitigations that they cannot implement and thus may need to put further attention on elimination or figure out how they can monitor for the actual attacks.

The ability to filter attack patterns by specific criteria was also discussed. Filtering is important to enable users to find the patterns that they care about. Implementing a tagging system where patterns could be tagged with specific metadata was an option mentioned and something that should be discussed in future sessions.

As a final note, attack patterns as they relate to the academic community were discussed. Patterns were seen as a potentially useful concept to help teach future engineers about the different types of attacks. Professors could work through the individual attack patterns and use each as a way to teach about the underlying weaknesses and potential mitigations. An entire course could probably be established using the attack patterns as the foundation.

3.2 "Mobile Application Vetting" Session

This session focused on identifying best practices and key criteria an organization should consider for assessing mobile applications. This criterion is critical to ensure your mobile apps have had as many weaknesses removed or mitigated by employing the proper protection mechanisms to defend against current and emerging cyber threats.

3.2.1 Session Goals

- Identify key criteria for assessing the security posture of a mobile application against mobile threats and attack patterns.
- Provide key recommendations on how criteria can be documented and shared across agencies in a common fashion.

3.2.2 Session Summary

The collaboration session discussion focused on identifying existing sources of vetting criteria to assess the security of mobile applications. Initially, questions were raised on the need for performing vetting at different stages of the SDLC vs. vetting after development. The discussion centered on data handling performed by mobile applications and the need for access control based on data sources. The mobile applications also tend to rely on several external software components and libraries that also need vetting.

The session shifted towards looking at methods to create and document criteria in a common way so they can be shared and understandable across government agencies, commercial companies and academic institutions. It was pointed out there is no single

source, or common documentation method at the Federal level. The following existing standards were identified:

- DISA Security Requirement Guides[2]
- OWASP Mobile Security Project[3]
- Common Weakness Enumerations (CWE)[4]
- Common Vulnerability Enumerations (CVE)[5]
- Traditional Secure Coding Guidelines (e.g., CERT[6])
- NIST SP800-53[7]

A single guide, with mapping between the existing guides to identify weaknesses, vulnerabilities and attacks was identified as a need. Once such a mapping exists, organizations can communicate in a common way about vetting practices. Once a common language exists, tool vendors can develop tools that align with government needs. An effort was made to discuss the potential for internal manual review of code; however this is probably not feasible due to time, labor, and complexity costs of performing such reviews.

Finally, the discussion looked at continuous feedback into the process where new attacks or weaknesses would make their way into policy, tools and procedures for vetting mobile applications. The academic community was identified as a good candidate to help validate the accuracy of tools/procedures that claim to identify weaknesses and help promote further research in that area by looking at the underlying techniques used by the tools and exploring the question of what those techniques are appropriate for and what they are not.

3.3 "Mobile Application Protection Mechanisms" Session

The mobile application protection mechanisms collaboration session aimed to identify key mechanisms used to protect mobile applications and their corresponding data.

3.3.1 Session Goals

- Review the current list of known protection mechanisms for mobile applications, including both native applications and web / HTML5 applications.
- Propose additional protection mechanisms to be added to the list.
- Output a set of initial recommendations around mechanisms that should be in place to protect mobile applications.

---

[2] http://iase.disa.mil/stigs/net_perimeter/wireless/smartphone.html
[3] https://www.owasp.org/index.php/OWASP_Mobile_Security_Project#tab.3DTop_Ten_Mobile_Risks
[4] https://cwe.mitre.org
[5] https://cve.mitre.org
[6] https://www.securecoding.cert.org/confluence/display/sci/Coding+Guidelines
[7] (http://csrc.nist.gov/publications/nistpubs/800-53-Rev3/sp800-53-rev3-final_updated-errata_05-01-2010.pdf

3.3.2 Session Summary

The collaboration session discussion focused on built-in protection mechanisms provided by mobile operating systems. Mobile operating systems typically provide sandboxing capabilities to prevent applications from interfering with each other or accessing another application's sensitive data. Mobile operating systems provide varying degrees of cryptographic capabilities that can be used to enable data-at-rest and data-in-transit protection. Some devices provide multi-persona solutions that can strongly separate personal applications and data from enterprise applications and data. Application vetting tools should perform checks to ensure that the OS security capabilities are being correctly leveraged.

The discussion shifted towards identifying how enterprises must also ensure that devices are appropriately configured. For example, a device screen lock passcode generally must be in place for the OS data-at-rest protection to be effective. Devices that are not kept up-to-date with security updates may be susceptible to publicly known vulnerabilities. Devices with USB debugging mode or similar developer modes enabled may be susceptible to exploitation. Mobile device management systems should be used to monitor device configuration and take action (such as denying access to enterprise resources) to insecurely configured devices. The Open Vulnerability and Assessment Language[8] (OVAL) standard has added experimental support for Android and iOS. OVAL can be used to standardize checking the state of Android and iOS devices against DISA Security Technical Implementation Guides (STIGs), other security configuration guides, or the Common Vulnerabilities and Exposures (CVE) dictionary of publicly known vulnerabilities, and reporting the results of assessments in a standard format that can be used by enterprise cyber security systems.

Session participants expressed concerns that in some environments, such as Bring Your Own Device (BYOD) environments, it may not be practical to require or ensure that the operating system is securely configured. Even if the OS is securely configured, new vulnerabilities are regularly discovered, and in some cases security patches may not be deployed in a timely manner (if ever). Application wrapping or application-level container technologies can be used to add application-level data-in-transit and data-at-rest protection, application-level authentication, or other security capabilities independent of what is provided by the operating system. However, security researchers have demonstrated that attackers who gain privileged access at the operating system level can subvert popular application-level-only solutions.

One practical recommendation to protecting sensitive data is to minimize its presence on mobile devices. Sensitive data should only be downloaded to devices when needed and retained on the device for the shortest necessary amount of time. Thin clients or non-persistent web browser sessions can be used when practical for accessing sensitive data.

Finally, the discussion shifted towards user authentication options. Smartcard readers for mobile devices have typically been expensive, have presented usability challenges, and

[8] http://oval.mitre.org

are generally not compatible with mainstream mobile applications. The federal government appears to me moving towards derived credential approaches, and away from smart card reads for mobility. The government can leverage strong built-in cryptographic capabilities now present in many mobile devices to support this shift towards derived credentials. Concerns were expressed that mitigations may be needed against physical loss or theft if there is no detachable authentication token. Wireless proximity-based authentication devices that need to stay near the mobile device for it to continue operating were suggested as a possible mitigation. Biometric authentication was also discussed as an option. The participants generally felt that voice solutions had the most promise, since this does not require any additional hardware such as a fingerprint reader.

3.4 "Cyber Intelligence" Session

The cyber intelligence collaboration session focused on the need for a framework to characterize and communicate mobile threat data and how this data can be shared, as well as ingested in the mobile application vetting process. There was discussion on the current approach for traditional applications that makes use of the Cyber Observable eXpression (CybOX).[9] CybOX is a standardized language for encoding and communicating high-fidelity information about cyber observables. The thought was that CybOX could be extended to mobile, and that the agile lifecycle of a mobile application provides an opportunity to apply knowledge of mobile threat data to the app vetting process, to the operations process through policy modifications (e.g. MDMs, MAMs) and more easily translated into application development guidance.

3.4.1 Session Goals

- Determine techniques and practices the Federal government can use to maintain good awareness of new and emerging cyber threats to mobile applications.
- Identify recommendations for keeping application evaluation criteria and security controls updated to assess the security posture of mobile apps.

3.4.2 Session Summary

The collaboration session discussed the life cycle of a mobile application to identify the uniqueness of a mobile application lifecycle vice traditional applications. The discussion focused around the following: a much looser, potentially less-controlled lifecycle of a mobile application where most anyone can develop an app to include non system developers; that we are getting away from a structured development environment; and that many apps may not be developed outside the controls of the government environment all resulting in the need for:

1. A hard and fast gate to examine apps before the application enters the ecosystem

[9] http://cybox.mitre.org

2. Means to automate most of the checking with an active "watch list" to keep up with the tempo of change e.g., DHS's Car Wash
3. Need to address known vulnerabilities

The discussion shifted towards sharing of threat data. Connected networks have more rich methods of detection on end-devices where the cyber threat sensors in the mobile ecosystem are not very mature. We discussed the need for higher-assurance integrity monitoring as well as the need for reporting on anomalous behavior from the full stack – device, service providers, and carriers. The discussion shifted towards who the consumers of this threat data are and what data they need and at what frequency. The stakeholders (control points) are:

1. Platform vendors (e.g. Apple, Samsung, etc)
2. Tools for developers (SDKs)
3. Developers (translate weaknesses into best practice)
4. Vetting (store front)
5. Provisioning/deployment
6. Operations
7. End-user
8. Program Owner

Each one of these stakeholders has a need for cyber threat data, most in a different format with different levels of timeliness. Moving from development to operations there is a need for more timely data. For developers there is a need to abstract it for future development. For end users there is a need to convey in human language such as email or text message.

The discussion shifted towards the cyber threat intelligence sharing standards being used in the non-mobile ecosystem that are components of CybOX: the Structured Threat Information Expression[10] (STIX), and Trusted Automated Exchange of Indicator Information[11] (TAXII) and how we could put a lens on mobile and capture mobile threat observables. This sharing infrastructure to support mobile does not exist today. There is an opportunity to identify mobile threat observables associated with mobile applications that can be ingested by the app vetting process. There is also the opportunity to share mobile threat observables, particularly those associated with a mobile app among established partners. The gaps include:

1. Defining cyber threat observables for mobile (need a better understanding of TTPs)
2. Determining how to generate a rich set of observables given lack of cyber threat sensors
3. Defining the translation of mobile threat observables/weaknesses into best practices (e.g. SRG- like activities) for developers

[10] http://stix.mitre.org
[11] http://taxii.mitre.org

4. Extending the community of cyber threat sharing to include data on mobile observables
9. Defining how the app vetting process can ingest the mobile threat observables as new vulnerabilities are identified

4 Summary

Several key themes emerged as a result of the four collaboration areas. First, the government mobile application security space is still fragmented. There are a wide variety of independent efforts scattered across the government, which creates challenges for commercial companies that are developing tools to satisfy government mobile application security needs.

Secondly, mobile application security encompasses more than just a single technology or tool. The government needs an approach/framework to characterize threats, attack patterns, weaknesses, vulnerabilities, protection mechanisms and vetting criteria. While there is growing consensus on how to document and share information about attack patterns[12], vulnerabilities[13], and weakness[14], the other aspects still need work. It is critical not only for Federal agencies to be able to share information and mobile apps across organizational boundaries, but also to provide commercial vendors with critical information to develop secure products and tools for Federal government use.

Thirdly, mobile operating systems and devices themselves provide sandboxing and other security capabilities that, if properly used, can provide protection against vulnerable or malicious applications. NIAP's Mobile Device Fundamentals Protection Profile specifies security properties that mobile devices should possess to mitigate well-known threats. Mobile device management systems can be deployed to monitor and enforce device configurations. Standards-based solutions[15] can be used to check and report the state of mobile devices against security configuration guides and lists of known vulnerabilities.

Fourthly, the mobile space is moving very fast. Having a robust, integrated Federal wide cyber intelligence program to inform common attack pattern repositories, protection mechanisms, and vetting criteria is critical to ensure agencies and commercial IT providers can adapt to new and emerging threats.

Fifthly and lastly, the academic community is a ripe, largely untapped resource to use to help devote research into some key challenges, and government and industry can work with academia to not only shape research, but to help shape curricula to produce engineers prepared to work in the mobile application security space.

---

[12] Common Attack Pattern Enumeration and Classification (CAPEC) capec.mitre.org
[13] Common Vulnerabilities and Exposures (CVE) cve.mitre.org
[14] Common Weakness Enumeration (CWE) cwe.mitre.org
[15] Open Vulnerability and Assessment Language (OVAL) oval.mitre.org

5 Recommendations

Extend CybOX[16] to Include Observables Associated with Mobile Threat Data

The Federal CIO Council Information Security and Identity Management Committee's (ISIMC) Mobile Technology Tiger Team should work with key federal agencies to define processes to develop and disseminate cyber intelligence to key mobile ecosystem participants. For example, this information can be given to the application vetting community to help identify threats, as well as help tool developers stay current with existing and emerging threats.

This information can also be provided electronically to enterprise mobile device managers (MDM), and mobile application managers (MAM). This machine-to-machine dissemination of observables can aid in automated discovery and protection of threats within the enterprise.

Improve the Mobile Application Information in CAPEC

The Federal CIO Council ISIMC Mobile Technology Tiger Team should champion the improvement of CAPEC by adding in more mobile application related information, using the draft attack patterns developed during the Collaboration sessions as a start. This gap should be filled through the already established partnerships in the DHS CAPEC effort between Government, industry and academia. Research potential exists in expanding specific areas of the CAPEC hierarchy.

Develop Federal Level Mobile Application SRG

The Federal CIO Council ISIMC Mobile Technology Tiger Team should champion and identify an owning government agency to create and maintain a Federal level mobile application SRG, based on DISA's SRG model. GSA or NIST are prime agencies to own this effort. This SRG should be created using standard languages, and created by mapping standards describing application weaknesses, and vulnerabilities such as DISA SRG, CWE, CVE, OWASP and should also map to the attack patterns (CAPECs) that would reveal a weaknesses presence as an exploitable vulnerability if simulated by a pen-test team, dynamic analysis tool, or a red-team assessment.

Once the initial Federal SRG is developed, there should be routine industry collaboration to encourage tools to adopt specific criteria sets, as well as to help improve the SRG. Government, industry, and academic feedback on new weaknesses, attack patterns or vulnerabilities should be used to continuously improve the Federal SRG.

[16] CybOX package is open source and can be found here: *https://github.com/CybOXProject/python-cybox*

Create a Central Clearing House for Mobile Apps

It can be cost prohibitive for many federal agencies to acquire and maintain application vetting tools and employ a full set of staff with the requisite subject matter expertise for all mobile platforms. The Federal CIO Council ISIMC Mobile Technology Tiger Team should work to identify a central government clearinghouse that can make the appropriate investments to specialize in application vetting. This clearinghouse can invest and refresh the appropriate vetting tools and staff, and can sell vetting services to other agencies. This approach allows agencies to focus on their mission and operations, and not have to make substantial investments in an application vetting capability. Agencies can make application-fielding decisions based on information developed by the vetting clearinghouse.

This government clearinghouse should also work with appropriate government agencies, such as NSA and NIST, to develop effectiveness measures to rate mobile application vetting tools. This will provide tool vendors with a minimum set of criteria to build towards, while providing the government with a level of assurance that the tools will detect applicable CWEs and CVEs.

Create an Integrated Advisory Capability Across FFRDC/Academia/Commercial

The Federal CIO Council ISIMC Mobile Technology Tiger Team should champion an Integrated Advisory Capability to broker discussions between academia, industry, and government through targeted events and special initiatives. Initially, the Integrated Advisory Capability should also focus on partnering with academic institutions to develop curricula at the graduate and undergraduate levels to ensure graduates are prepared for government mobility challenges. This curricula should also be expanded to include government training opportunities to ensure key government personnel have affordable access to high quality training in emerging mobile topics.

This Integrated Advisory Capability should also work with universities to target research around key government and commercial mobility challenges, which include, but are not limited to:

- Methods to evaluate effectiveness of mobile application vetting tools
- New attack patterns and countermeasures
- Innovative identity and access management methods
- Tools and methods that provide high assurance for vetting mobile applications without source code

Attachment A: Mobile Attack Pattern List

The following tables list the individual mobile attack patterns that were discussed during the mobile application threats and attack patterns collaboration session.

1) Adversary has physical access to a device.

Threat	Attack Type	Attack Patterns	Negative Technical Impact
An adversary with physical access to a device performs a Jailbreaking/Rooting process and downloads all the data off of it. Anything that is not properly encrypted will eventually be obtained by the adversary.	Privilege Escalation	iOS Jailbreak / Android Rooting	Information Disclosure
An adversary with physical access to a device attaches a debugger and modifies the execution flow of an application, potentially subverting authentication routines or data validation functionality.	Reverse Engineering	Expose Hidden Functionality via Debugger	Application Misuse
An adversary with physical access to a device reverse engineers a target application and retrieves hard-coded keys		CAPEC-190 Expose Hidden Content via Disassembler	Information Disclosure
An adversary with physical access to a device performs a backup of the device using the adb command. The adversary leverages this to lift potentially private application data off the android device by performing a backup. (Android only)	Functionality Misuse	System Backup Exploitation	Information Disclosure

An adversary with physical access to a device views the history file in an attempt to find passwords that were entered into standard text fields and hence recorded.		Saved History Exploitation	Information Disclosure
An adversary with physical access to a device views the screenshots in the multi-tasker switcher and is able to see sensitive information. (iOS only)	Excavation	CAPEC-498 Probing Application Screenshots	Information Disclosure
An adversary with physical access to a device obtains access to an application by using brute force techniques to determine a user's authentication credentials.	Authentication Theft	CAPEC-49 Password Brute Forcing	Information Disclosure Application Misuse

2) Adversary tricks user into executing a malicious application.

Threat	Attack Type	Attack Patterns	Negative Technical Impact
An adversary, through a previously installed malicious application, presents a link/button that claims to direct the user to a trusted site. (e.g., a Facebook like button) However, the malicious application instead shows a fake screen that looks identical to what the user expects. The fake screen requests sensitive information (e.g., the user's password) in the identical way that the trusted site would. The user enters the information not knowing that they are giving it to the adversary.	Spoofing	Fake User Interface	Information Disclosure

An adversary, through a previously installed malicious application, intercepts an implicit intent sent to launch a trusted activity and instead launches a counterfeit activity in its place. The malicious activity is then used to mimic the trusted activity's user interface and convince the user to enter sensitive data as if they were interacting with the trusted activity. (Android only)		CAPEC-501 Action Spoof (Activity Hijack)	Information Disclosure Modify App Data Application Misuse
An adversary, through a previously installed malicious application, registers for a URL scheme intended for a target application that has not been installed. Thereafter, messages intended for the target application are handled by the malicious application. Upon receiving a message, the malicious application displays a screen that mimics the target application, thereby convincing the user to enter sensitive information.		CAPEC-505 Activity Hijack (Scheme Squatting)	Phishing
An adversary, through a previously installed malicious application, displays an interface that sits on top of the target application interface. The user is convinced to tap in an attacker chosen location. The malicious application allows the tap to pass through to a trusted application running in the background. (Android only)		Action Spoof (TapJacking)	Information Disclosure Modify App Data Application Misuse
An adversary, through a previously installed malicious application, monitors the task list maintained by the operating system and waits for a specific legitimate task to become active. Once the task is detected, the malicious application launches a new task in the foreground that mimics the user interface of the legitimate task. (Android only)		CAPEC-504 Action Spoof (Task Impersonation)	Phishing

An adversary uses a malicious app that loads a trusted page via WebView to inject JavaScript and perform actions as the (logged in) user against a trusted web app.	Remote Code Inclusion	CAPEC-500 WebView Injection	Information Disclosure Execute Arbitrary Code
An adversary uses a malicious app that loads a trusted page via WebView to sniff events and achieve information disclosure.			Information Disclosure
An adversary uses a malicious app that loads a trusted page via WebView to hijack events and issue a fake response.			Information Disclosure Denial of Service Data Injection
An adversary, through a previously installed malicious application, evades malicious app detection by dynamically downloading application code. By installing the malicious code after the user has installed the application, the adversary is able to avoid detection during the vetting process. (Android only)		Dynamically Downloaded Code	Execute Arbitrary Code
An adversary convinces the user to give a malicious application privilege to potentially sensitive resources. For example, microphone/camera recording, reporting back geo-location information, stealing contact list information, intercepting SMS text messages (potentially including authentication codes).	Incorrect Permissions	Obtain Extra Privilege	Information Disclosure

3) Adversary interacts with a target application through a malicious application.

Threat	Attack Type	Attack Patterns	Negative Technical Impact

An adversary, through a previously installed malicious application, issues an intent directed toward a trusted application's component that has been unintentionally exported and made public. If the component blindly trusts the intent's action, then the target application performs the functionality at the attack's request, helping the attacker achieve the desired negative technical impact. (Android only)	Spoofing	CAPEC-502 Intent Spoof	Information Disclosure Denial of Service
An adversary, through a previously installed malicious application, intercepts an implicit intent sent from a trusted application. If the intent is not protected by a permission that the malicious application lacks, then the malicious application has access to the data contained within the intent. The interception of the implicit intent can lead to information disclosure, denial of service, data injection, activity hijacking, and false responses. (Android only)	Interception	CAPEC-499 Intent Intercept	Information Disclosure Denial of Service Modify App Data Application Misuse
An adversary, through a previously installed malicious application, obtains access to internal files that were given improper Linux file permissions (world readable and/or writable) by the host application. (Android only)	Exploitation of Authorization	CAPEC-1 Accessing Functionality Not Properly Constrained by ACLs	Information Disclosure

4) Adversary interacts directly with a target application.

Threat	Attack Type	Attack Pattern	Negative Technical Impact

An adversary sends a malicious request to an application installed on the device. The application improperly handles the request and exposes sensitive information in its response.	Injection	CAPEC-63 Simple Script Injection (Cross-Site Scripting)	Information Disclosure Execute Arbitrary Code
An adversary sends a malicious request to an application installed on the device. The application uses malicious data in the request to improperly build SQL statements.		CAPEC-66 SQL Injection	Information Disclosure Modify App Data
An adversary leverages permissions that are granted (but not needed) by an existing application. Applications should only be allowed to request the permissions that they need.	Incorrect Permissions	Abuse Extra Privilege	Information Disclosure
An adversary looks to exploit an application that registers and incorrectly handles a custom URL scheme.	Vulnerable Functionality	Initiate Without Confirmation	Application Misuse
An adversary tricks the user into loading malicious content into an application leveraging functionality exposed through WebView. The malicious content takes advantage of the exposed functionality to access system resources or sensitive application data.	Exploitation of Authorization	CAPEC-503 WebView Exposure	Information Disclosure
An adversary obtains access to an application by using stolen authentication credentials of a valid user.	Authentication Reuse	Reusing Logon Credentials	Information Disclosure Modify App Data Application Misuse

5) Adversary communicates with a target application's user.

Threat	Attack Type	Attack Pattern	Negative Technical Impact

An adversary masquerades as a legitimate entity with which the user might do business with in order to prompt the user to reveal some sensitive information (very frequently authentication credentials) that can later be used by the adversary.	Impersonation	CAPEC-98 Phishing	Information Disclosure

6) Adversary has access to a target application's supply chain.

Threat	Attack Type	Attack Pattern	Negative Technical Impact
An adversary embeds malicious code into a target application at some point during the supply chain.	Supply Chain	CAPEC-442 Malware Infection into Product Software	Execute Arbitrary Code
An adversary is also a developer of a target application and hides a backdoor or some other malicious code.	Insider Threat	CAPEC-443 Malicious Logic Inserted Into Product Software by Authorized Developer	Execute Arbitrary Code

7) Adversary manipulates network traffic.

Threat	Attack Type	Attack Pattern	Negative Technical Impact

An adversary targets the communication between two components (typically a client and a server) by placing himself in the communication channel between the two components. Whenever one component attempts to communicate with the other, the data first goes to the attacker, who has the opportunity to observe or alter it, and it is then passed on to the other component as if it was never intercepted.	Impersonation	CAPEC-94 Man-in-the-Middle	Information Disclosure Modify App Data
An Adversary creates a public wireless network that appears legitimate (e.g. airport, Starbucks) and sniffs network traffic in-transit.	Interception	CAPEC-158 Sniffing Network Traffic	Information Disclosure
An adversary consumes the resources of a target by rapidly engaging in a large number of interactions with the target. This type of attack generally exposes a weakness in rate limiting or flow control in management of interactions. Since each request consumes some of the target's resources, if a sufficiently large number of requests must be processed at the same time then the target's resources can be exhausted.	CAPEC-125 Flooding	<numerous>	Denial of Service