

Anatomy of Mobile Attacks Whitepaper

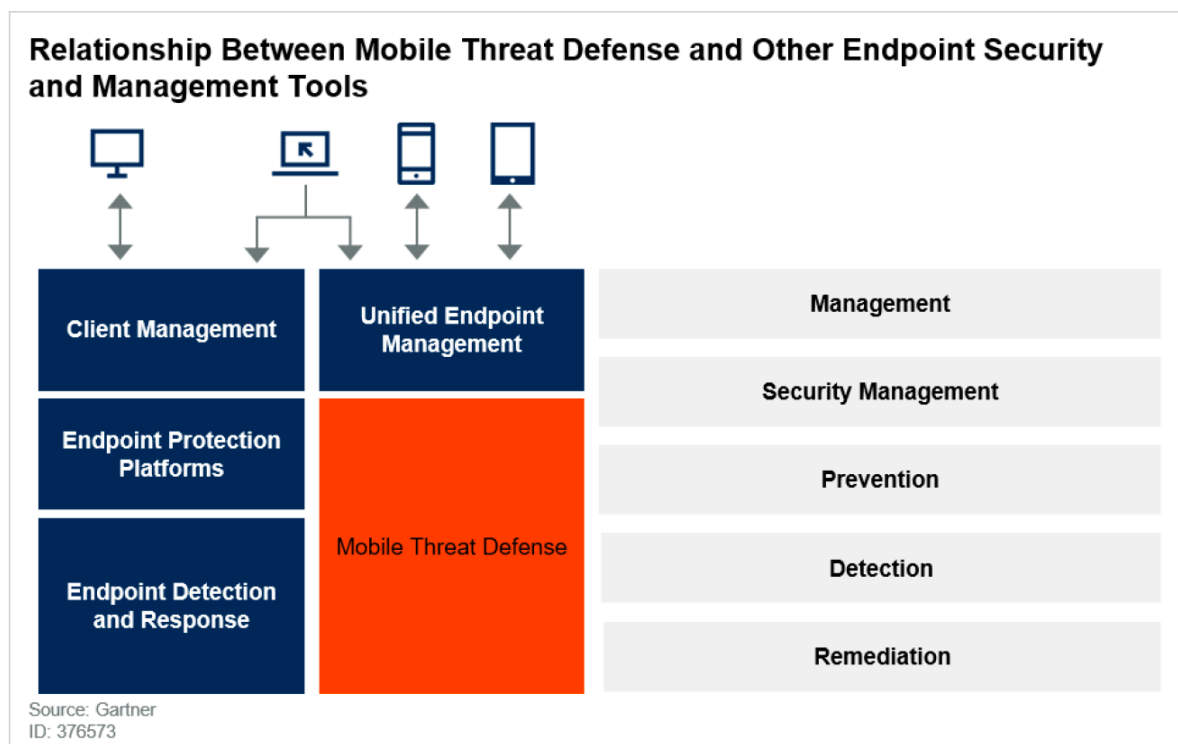
June, 2020

MISSION

In order to protect Federal & State, Local & Education Government agencies from mobile cybersecurity threats and attacks, we must first understand the enormous scope of mobility and that it spans every corner of the country and globe to conduct business; end-users will leverage both custom and native applications contained on iOS and Android-based mobile platforms for both phones and tablets while leveraging Cellular and Wi-Fi connections (*including both controlled and open Access Points*) for outbound connections to transfer data from those devices back to closed and public servers in data center or the cloud.

And although these mobile platforms are often protected by a set of baseline solutions, such as device encryption and Unified Endpoint Management suites (A.K.A. MDMs), they are NOT protected with sufficient detection or remediation of malicious attempts to compromise them from any of the (4) pillars of attack surface: Data, Network, Application & Phishing, nor does it provide the IT Mobility or Security Operations support staff with the valuable intrinsic 'Threat Intel' necessary to remediate for Endpoint Detection & Response and Incident Response.

Figure 1. Relationship Between Mobile Threat Defense and Other Endpoint Security and Management Tools



OBJECTIVE


There is universal acceptance that government endpoints are high-value targets (HVTs) of Nation State actors. The highest priority is the protection of the data contained on device, both at-Rest and in-Transit, as it should be. And although as mentioned some basic defenses exist on the mobile platforms, a Nation State Actor or a professionally focused advisory, whether directly or 'indirectly' working for a Nation State would have numerous utilities at their disposal from indirect phishing or direct spear-phishing attempts through application / web / SMS-MMS / network-based attack vectors that can circumvent Out-of-the-Box capabilities (ex. Encryption & Device Controls). Thus, a layered or Defense-in-Depth (DiD) approach is required to ensure that those devices <and> the network resources used to communicate with have zero-day protection via an on-device, Machine Learning (ML) tool.

BACKGROUND

Sophisticated attacks of this nature could include a myriad of available options that are openly available on the darkweb and in any one of a half-dozen categories, such as Cryptomining/ Click-Bait/ Ransomware/ Phishing.


Remote Access Trojan (RAT)/ Spyware are included with such well documented nefarious kits and tools, Dark Caracal, Desert Scorpion, FrozenCell, Karma, Pegasus, RCS, SilverHawk and ViperRAT!

Lastly, new in this arsenal bag are 'unpatchable' jailbreaks for iOS such as checkra1n, that has been weaponized with checkm8.

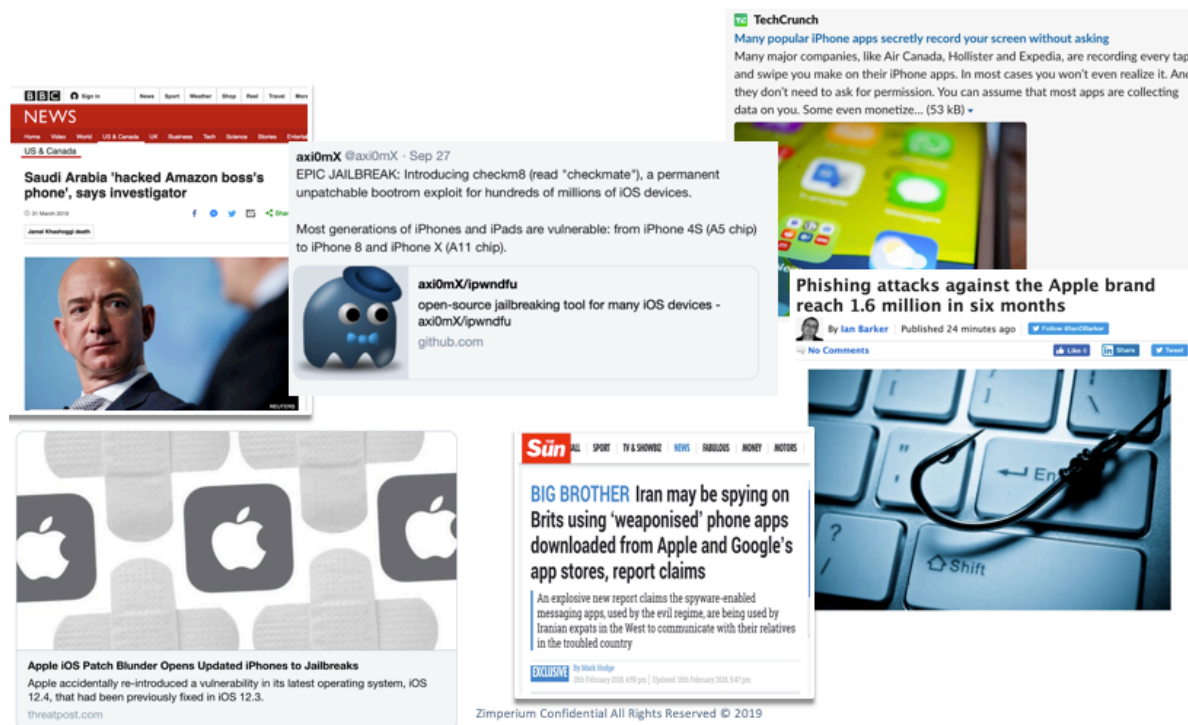
 **Zimperium Mobile Security Blog**

The Unpatchable Checkra1n Exploit

Today, the "unpatchable" jailbreak known as Checkra1n (Device Compatibility) was officially released and generally available. Checkra1n is unprecedented in potential impact with millions of devices at risk as a result of the extensive device and iOS targets.



It's clear adversaries have a staggering amount of options within this community beyond custom development, thus making assignment of protection of end-users against another nation-state security & Intel agency a challenge at best without a well-crafted and directly designed toolset, when considering also that it is well-known that countries like China/DPRK/Russia will often hire contractors or groups like China's APT41 Or N. Korea's Advanced Persistent Threat (APT#) groups ex. APT38 to do their dirty work, all the while having a growing solicitation from industry vendors in the Cyber Spycraft arena ex. Expert Team, FinFisher, HackingTeam, IPS, NSO Group, Verint and Wolf Intelligence to name a few, which can again provide quick access to the equivalent, enterprise-grade set of tools provided the opportunity of these potential intelligence or politically motivated attacks.














In summary, most attacks are simply modifications or customization of tools that have been developed in the past and then often tailored to specific attack goals <and> can be grouped together for additional complexity. Thus, the most important thing is to ensure agencies reduce their over-reliance on static defenses and end-user training in an attempt to safeguard against such advanced, persistent attacks! Even if newly developed code is used for an attack, signatures or fingerprints of any such attack still exhibit 'behavioral' activities when executed that expose them as nefarious actions and thus can be detected and remediated by an advanced ML MTD like Zimperium.

CURRENT STATE

Today, there are several key attack vectors and targeted vulnerabilities or 'threats' for mobile platforms and although statistically there are differences between them, in the end the use of any of them are targeting the same result, data exfiltration or an attempt to own and control the device as an unknowing weapon or 'beach-head' against an ultimate target. Nation State Actor(s) are going to look for soft targets for an attack surface. We know that in the Top 50 financial fraud was associated with mobile and the FinServ market each year is perennially considered both a top target and a top cyber security practitioner. There is no logical argument against these nation states not using similar vulnerabilities to exploit FedGov mobile endpoints <and> those 'new' exploits that will be derived from the advent of checkra1n with access to the O/S and the communications will only further promise the development of new variants of malware beyond checkm8!

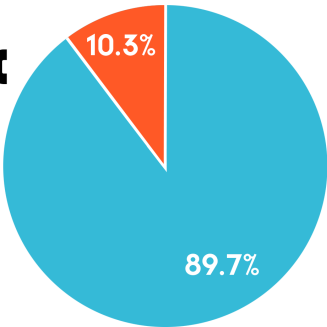
Below is a current state of enterprise mobile security from Zimperium's trove of data from '45 million endpoints worldwide' providing analysis on 'Threats' & 'Attacks', where 'Threats' are conditions that increase the likelihood of a device being attacked or enable attacks to be made more efficiently.

| VECTOR | KEY FINDINGS |
|---|---|
|  | 27% of enterprise mobile endpoints were exposed to device threats. |
|  | Mobile OS vendors created patches for 440 security vulnerabilities. |
|  | The majority of malicious profiles (68%) were considered “high-risk”, meaning they had elevated access that could lead to data exfiltration or full compromise. |
|  | 32% of enterprise mobile endpoints encountered risky networks, and 7% of enterprise mobile endpoints were exposed to network attacks. |
|  | ARP man-in-the-middle (MITMs) were attackers’ favorite weapon with 48% of all network attacks and 45% of total attacks on enterprise endpoints. |
|  | The number of network attacks detected in The Republic of Korea is slightly less than the total detected in the next four countries combined. |
|  | Zimperium’s machine learning-based engine, z9, detected thousands of malicious apps that were not in ‘VirusTotal’ or any other repository. |
|  | Malicious apps were 45% of Android attacks versus less than 1% of those detected on iOS. 98% of all detected malicious apps were on Android. |
|  | 5% of enterprise mobile endpoints had sideloaded apps from sources outside the authorized and vetted Apple App Store or Google Play Store. 36% of the Android devices has sideloaded apps versus only 2% of iOS ones. |
|  | 70% of iOS apps had advertising capabilities and iOS Bluetooth beacon usage exploded to 69% of apps (from 38% at the beginning of 2019). |
|  | 24% of iOS apps passed sensitive information over the web unencrypted. |

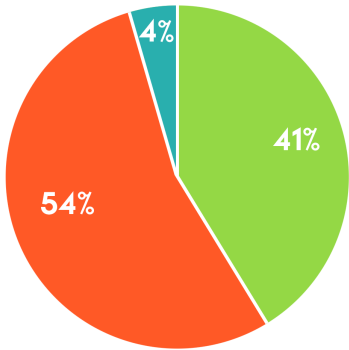
TOTAL THREATS & ATTACKS, PER O/S TYPE

THREATS & ATTACKS

THREATS ATTACKS

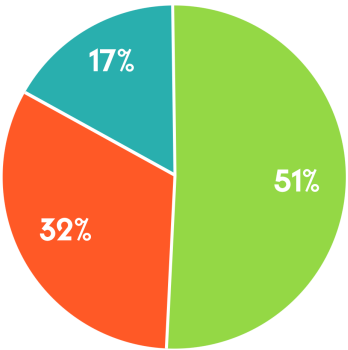





iOS



DEVICE NETWORK APPS

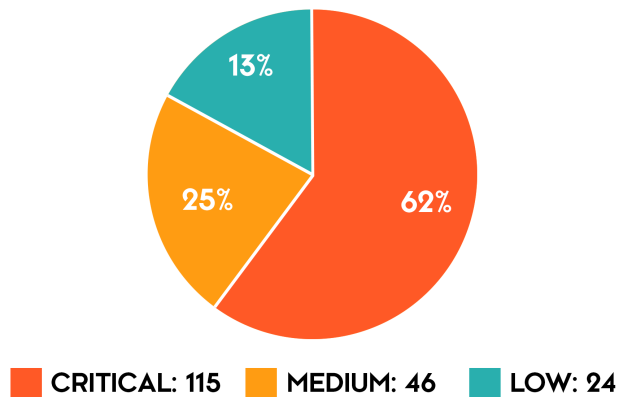
Android



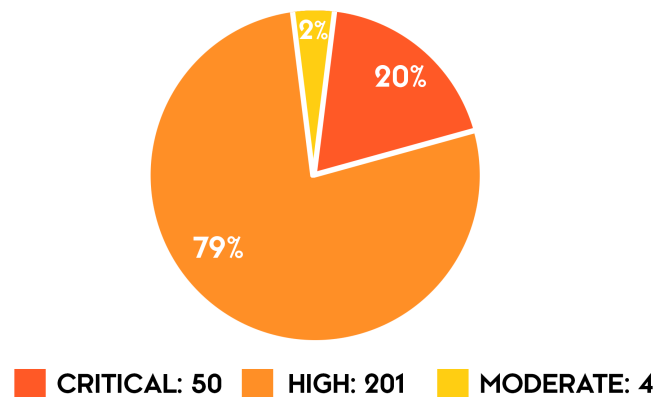
| ATTACKS | | iOS | Android |
|---|---------|-------|---------|
|  | DEVICE | 0.2% | 4% |
|  | NETWORK | 99.7% | 52% |
|  | APPS | 0.1% | 45% |

APPLICATION-BASED COMMON VULNERABILITIES & EXPOSURES (CVEs) & ATTACK VECTORS

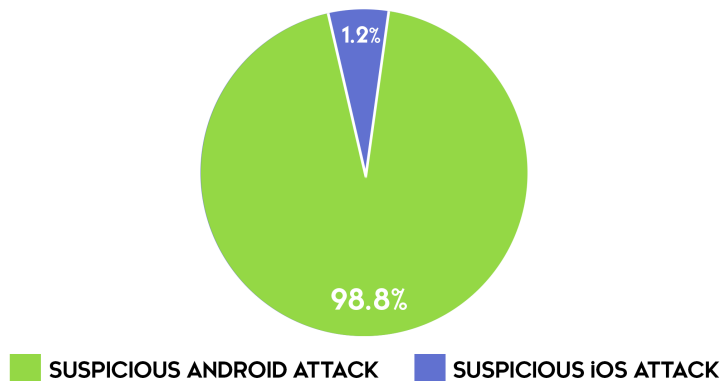
iOS CVEs (1H2019)



Android CVEs (1H2019)

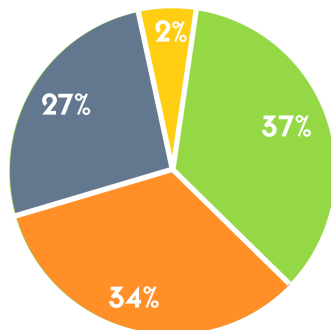


APP ATTACKS



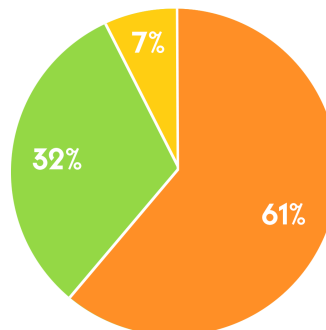
DEVICE & NETWORK-BASED ATTACK VECTORS

DEVICE ATTACKS



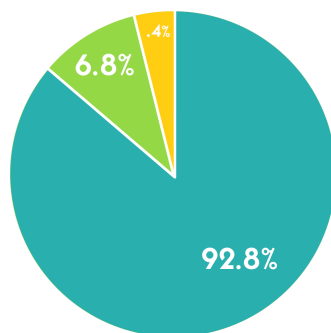
ABNORMAL PROCESSES SYSTEM TAMPERING
DEVICE ROOTED OTHER ELEVATION OF PRIVILEGES (EOP)

NETWORK THREATS



UNSECURED WIFI NETWORK CAPTIVE PORTAL OTHER NETWORK THREATS

NETWORK ATTACKS



MAN IN THE MIDDLE (MITM) SSL / TLS DOWNGRADE ROGUE ACCESS POINT

We only need look at news items related to NSO, Gamma, LU, the Russian Military Intel Service's (GRU) mobile surveillance of NATO, and countless others to know they are actively using these but to reiterate, the number one objective of nation state hackers is to either steal or disrupt. In either case, data will be exfiltrated or tampered causing considerable financial and political repercussions. Zimperium's solutions are designed to detect attacks at their very earliest stage. In fact, we have the effect of lowering the risk posture of devices such that many attacks never occur that would otherwise have succeeded in a complete kill chain or the MITRE ATT&CK for Mobile (AfM).

Anatomy of a Mobile Attack

Cyber Kill Chain

The Cyber Kill Chain (CKC) and MITRE's ATT&CK for Mobile (AfM) are tightly coupled 'case models' used to provide context around both the attacks performed, and ultimately what Zimperium's capabilities are to detect and defend against attacks that occur at every stage of the models, regardless of when they are started, or stopped.

Each case covers different (7) stages of the CKC and the use of (13) different AfM technique categories used in the wild against FedGov mobile endpoints. Although, some only cover individual stages, others show how an attack can be chained together to cover the entire model grouping, and some show how the attacks can start later on in the CKC without being chained together with earlier attacks. Regardless of if those stages happen in isolation, or if they're chained together, Zimperium's solutions can detect them at each stage!



Staged Attack Model

Below is a description of each of the stages and goals for different attack techniques that effect each stage.

Stage I: Discovery & Reconnaissance

Attackers will start off by doing recon on the victim and their devices to uncover details regarding the user and its identity, the device and/or the related network. Ultimately attackers aim to understand what they can leverage in order to reach their intended goals. Reconnaissance can cover anything from information gathering techniques to scanning networks.

Examples of reconnaissance are things like gathering OS versions to understand vulnerabilities, URL's of websites the user visits, IP and mac address of a devices, Ports open on the devices to identify vulnerable services that are running, etc as shown below:

- **Device:**
 - Vulnerable OS's (and how it's different from the latest OS)
 - Configurations (Profiles settings, etc)
- **Network:**
 - Current network connection
 - Nearby network's
- **Apps:**
 - Impact of app security risk to device risk posture
 - Risk of data exfiltration

Stage 2: Weaponization

In this stage, attackers can aim to establish control of the communication the device produces on the attached network(s). This control will then be leveraged to exfiltrate data from the communication and/or manipulate it to set up for further stages of the attack for weaponization.

The goal for this stage can be established on various levels of the OSI model using various techniques. Most common ones include but are not limited to:

- Rogue Wi-Fi Access Point
- ARP MitM
- SSL/TLS MitM

Scenarios that demonstrate this:

- Network MitM Compromise
- Cyber Kill Chain (i.e. Android)
- Rogue Access Point
- Malicious Profile (i.e. iOS)

Stage 3: Delivery

During the Delivery stages attackers aim to gain some kind of presence on the device they can leverage for their malicious intent. Delivery can be realized in many ways and all potential data input streams to a device should be considered. Commonly the delivery is also combined with some form of social engineering to persuade the user to allow some form of access, like through an Application download. This is however not a requirement as there are exploitation techniques that don't require any user interaction whatsoever.

- Wi-Fi Network
- GSM Network
- USB
- NFC
- Bluetooth
- Message (ex. SMS/MMS/E-Mail)
- Redirect to a site with malicious payload or exploit (ex. Stagefright or Pegasus)
- Malicious image (ex. iMessage)

- Other?

Once the attackers have delivered their payload and/or established their presence on the device they can progress to the next/other stages as they see fit.

Testcases that demonstrate this:

- Malware Sideloaded (i.e. iOS)
- Malware Sideloaded (i.e. Android)
- Cyber Kill Chain (i.e. Android)
- Cyber Kill Chain (i.e. iOS)

Stage 4&5: Exploitation & Installation

During the Delivery stages attackers aim to gain some kind of presence on the device they can leverage for their malicious intent. Delivery can be realized in many ways and all potential data input streams to a device should be considered. Commonly the delivery is also combined with some form of social engineering to persuade the user to allow some form of access, like through an Application download. This is however not a requirement as there are exploitation techniques that don't require any user interaction whatsoever.

- Wi-Fi Network
- GSM Network
- USB
- NFC
- Bluetooth
- Message (ex. SMS/MMS/E-Mail)
- Redirect to a site with malicious payload or exploit (ex. Stagefright or Pegasus)
- Malicious image (ex. iMessage)
- Other?

Once the attackers have delivered their payload and/or established their presence on the device they can progress to the next/other stages as they see fit.

Cases that demonstrate this:

- Malware Sideloaded (i.e. iOS)
- Malware Sideloaded (i.e. Android)
- Cyber Kill Chain (i.e. Android)
- Cyber Kill Chain (i.e. iOS)

Stage 6: Command & Control

At the Command and Control stage the goal for the attacker will be to leverage any presence in order to trigger behaviors on the device desired by the attacker. This can involve using remote exploiting frameworks or command and control server. E.g. Metasploit

In principle at this stage the attacker sends command to the device that allows him to control the behavior of the device. These behaviors can be used to support data exfiltration or set up for additional stages during an attack.

For demonstration purposes commonly remote shells are used to show that command and control capabilities have been established but for real life attacks it would be very likely that attackers use fully automated command and control processes.

An example of this is a bot-net where attackers have a persistence on multiple devices and send commands to them in order to leverage the Command & Control (C&C) capabilities to start a DDoS attack.

Cases that demonstrate this:

- Cyber Kill Chain (i.e. Android)
- Cyber Kill Chain (i.e. iOS)
- Stagefright
- Malicious Charger (ex. Checkm8)

Stage 7: Persistence & Data Exfiltration

Once a device has been compromised, the attacker will be able to achieve the ultimate goal of a) creating a persistence on the device that the user cannot easily get rid of and b) exfiltrate data that is of interest to the attacker. Data might include things like sensitive files but also user identity tokens e.g. certifications, usernames, passwords, etc...

Creating the persistence can for example be established by adding an application to the firmware of the device that is hidden from view and automatically launches with the device.

Cases that demonstrate this:

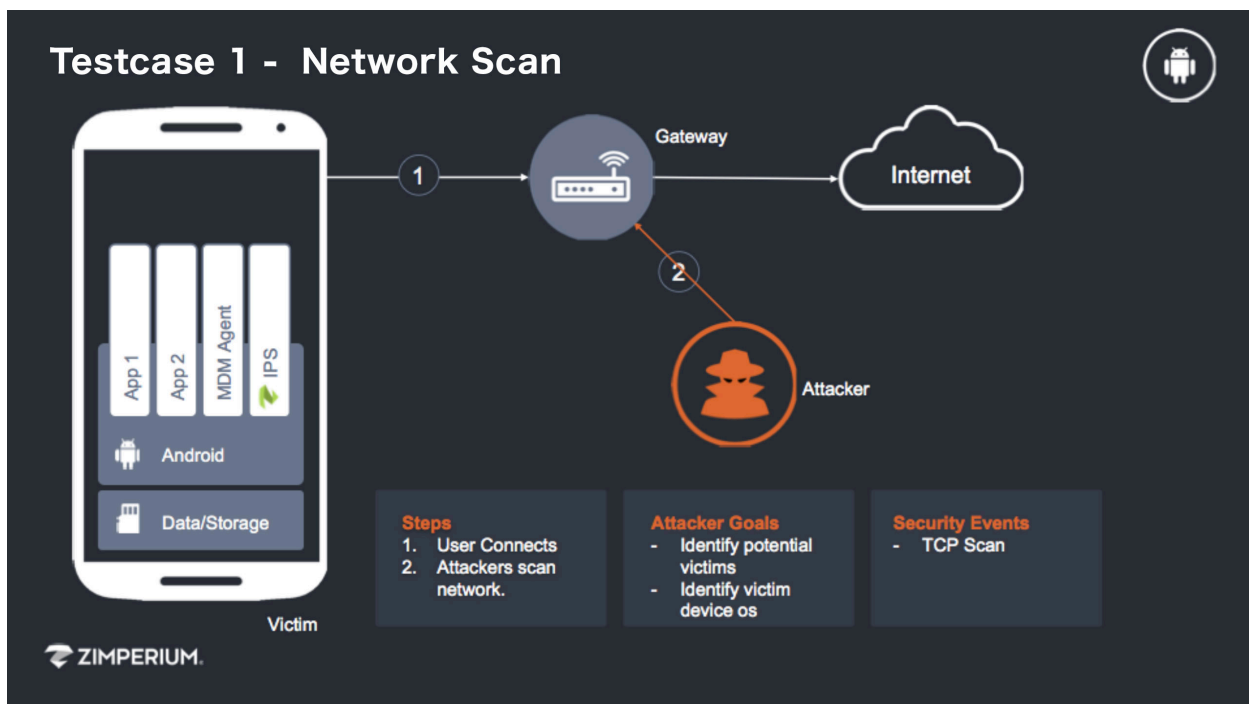
- Cyber Kill Chain (i.e. Android)
- Cyber Kill Chain (i.e. iOS)
- Stagefright
- Malicious Charger (ex. Checkm8)

Testcases for Cyber Kill Chain & ATT&CK for Mobile

In order to achieve successful implementation of data exfiltration, eavesdropping or manipulation, an adversary must first perform staged attacks through the (6) stages of the CKC, and any one of numerous testcases from the (10) MITRE ATT&CK for Mobile (AfM) technique categories and over (80) methods. The following is an anatomy of an attack for mobile, based on several well-known usecases from them:

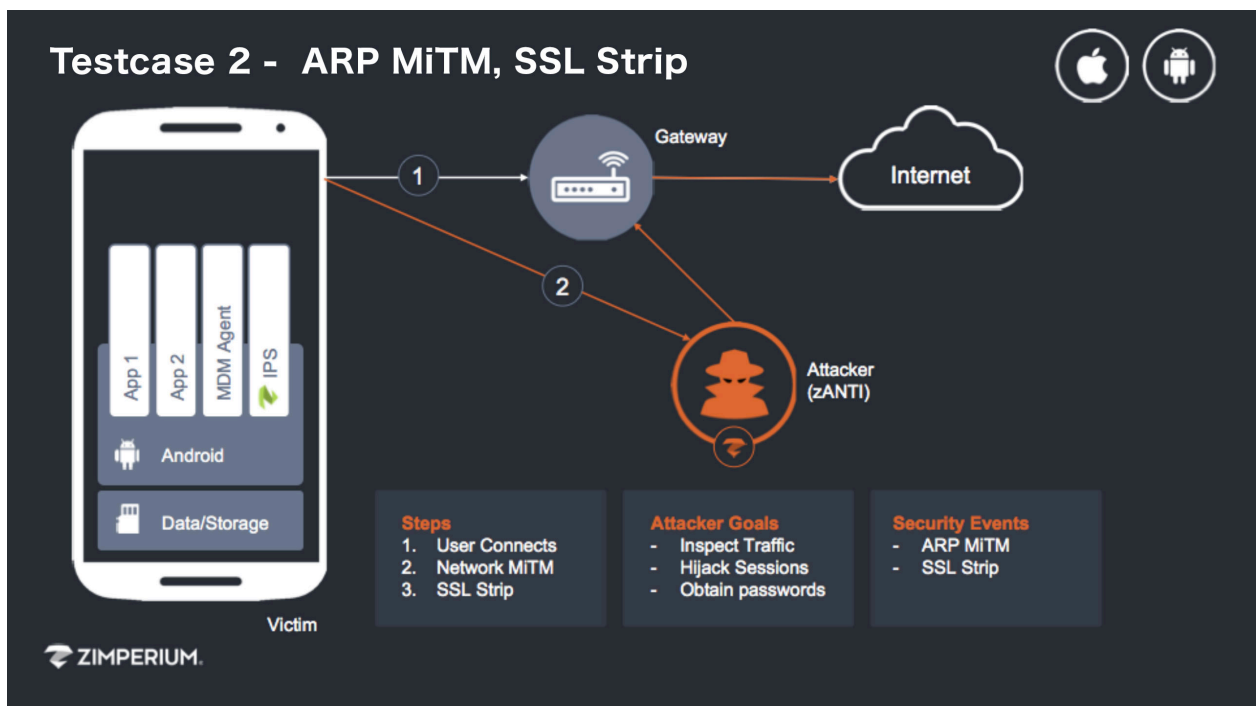
| Testcase Name | Network Scan |
|------------------------|--|
| Platform | Android (all versions), iOS 9.x and below |
| Description | Scanning on the network allows attackers to discover potential targets and is generally a precursor to an actual attack. This test cases will be aimed at discovery of android devices on the network using a variety of scan techniques |
| Expected threat events | <ul style="list-style-type: none">• TCP Scan• IP Scan• ARP Scan |

| | |
|---------------------|---|
| | <ul style="list-style-type: none"> • UDP Scan |
| Applicable stages | Stages = 1 |
| Pre-Conditions | Wi-Fi Network is connected to the internet No Host-Isolation is present in the Wi-Fi Network No VPN has been established by the device Attacker is connected to the Wi-Fi network. |
| Post-Conditions | Attacker has identified the victim in the network. |
| Execution Steps | Victim connects to the Wi-Fi Attacker starts Network Recon Scan Attacker obtains information as a result of the scan. |
| Possible Variations | ... |



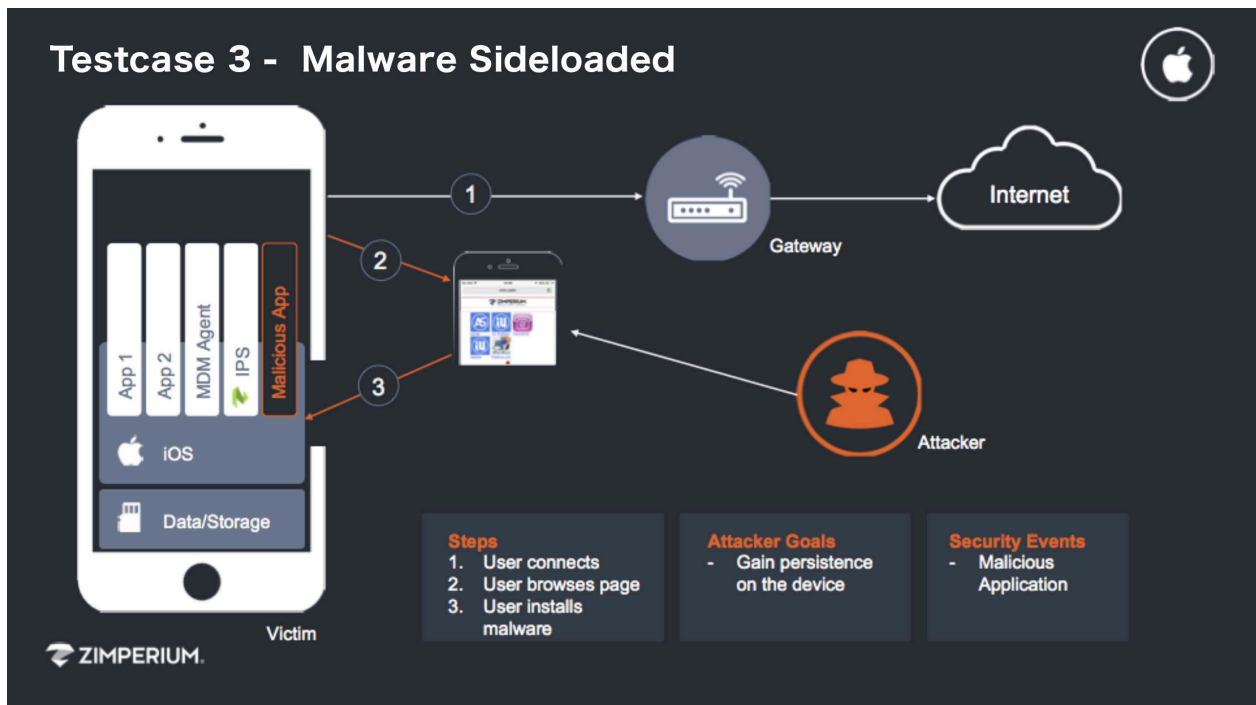
| Testcase Name | ARP MiTM, SSL Strip |
|------------------------|---|
| Platform | iOS, Android |
| Description | Attackers can gain control of the network traffic produced by a device running an ARP MitM and subsequently changing the content of the traffic During the attack images displayed on web pages will be replaced to demonstrate control of the traffic |
| Expected threat events | ARP MitM SSL Strip SSL MitM |
| Applicable stages | Stages = 2, 3 |
| Pre-Conditions | Wi-Fi Network is connected to the internet No Host-Isolation is present in the Wi-Fi Network |

| | |
|---------------------|--|
| | No VPN has been established by the device Attacker is connected to the Wi-Fi network |
| Post-Conditions | Victim is connected to the attacker instead of the gateway of the network. Attacker has full control over traffic flow between victim and network gateway demonstrated by: <ul style="list-style-type: none"> - Manipulation of http traffic (content injection) - Inspection of http and https traffic (traffic in cleartext) On the victim's device, the impact of traffic control can be observed by browsing various pages which are clearly manipulated by the attacker e.g. images are replaced |
| Execution Steps | Victim connects to the Wi-Fi Attacker starts Network Attack <ul style="list-style-type: none"> - ARP MitM - SSL MitM - SSL Strip Victim browses a page on the device |
| Possible Variations | Configure the MTD agent with mitigation actions to route traffic to a sinkhole or disconnect from Wi-Fi |



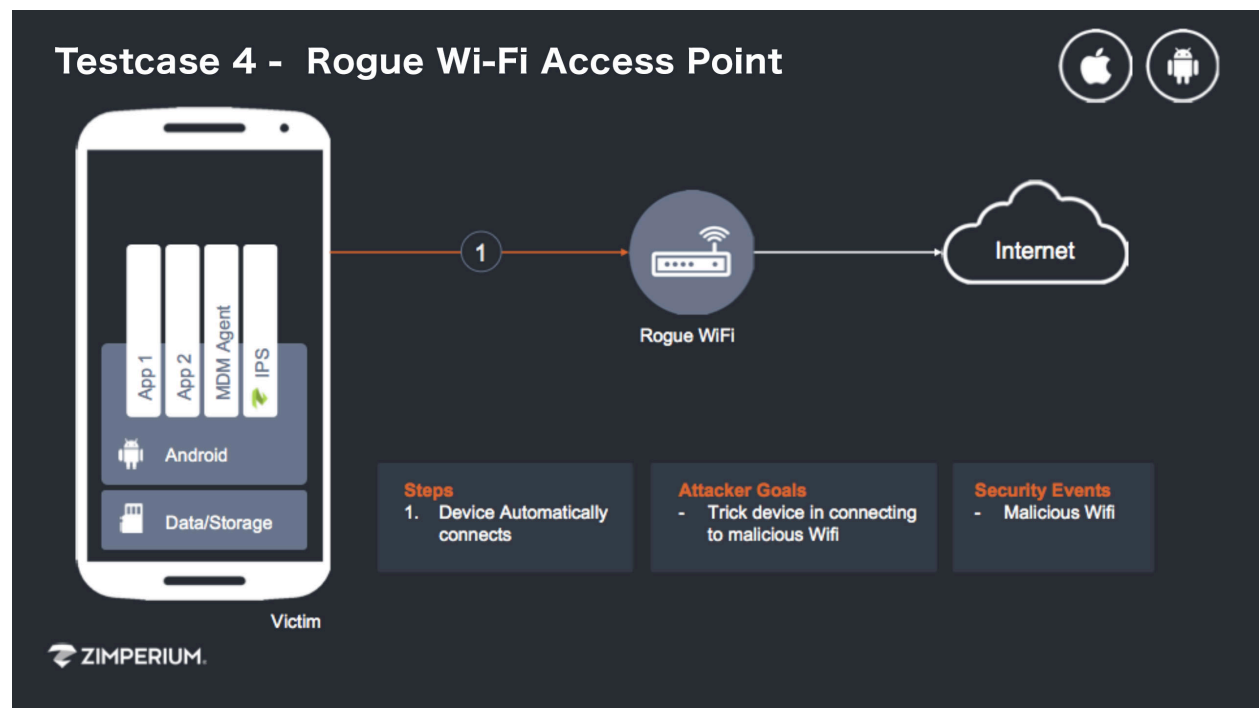
| Testcase Name | Malware Sideloaded |
|------------------------|--|
| Platform | iOS (MDM enrolled) |
| Description | Download and installation of malware from a custom crafted page. |
| Expected threat events | <ul style="list-style-type: none"> • Suspicious App • Sideloaded App |
| Applicable stages | Stages = 3, 4 |

| | |
|---------------------|--|
| Pre-Conditions | <p>Attacker is connected to the Wi-Fi network</p> <p>Attacker is actively routing traffic to his own host via either Rogue AP or ARP MitM</p> <p>No Host-Isolation is present in the Wi-Fi Network</p> <p>No VPN has been established by the device</p> <p>Wi-Fi Network is connected to the internet</p> <p>A certificate trusted by the device is installed on the webserver</p> |
| Post-Conditions | <p>Malicious application is installed on the device</p> <p>Side loaded application detected on the device</p> |
| Execution Steps | <p>Attacker Launches Attack:</p> <ul style="list-style-type: none"> Start DNS poisoning Launch webserver hosting malicious payload <p>Victim connects to the Wi-Fi</p> <p>Victim browses a page allowing the installation of various apps is shown</p> <p>The victim taps a link and runs the app installation.</p> |
| Possible Variations | Sideloaded a malicious iOS application |



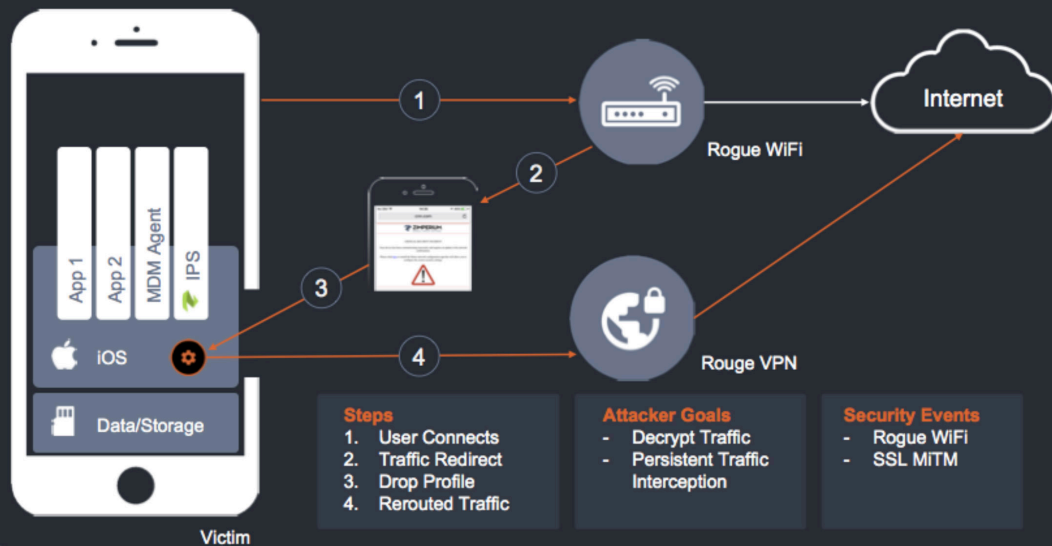
| Testcase Name | Rogue Wi-Fi Access Point |
|------------------------|--|
| Platform | iOS, Android |
| Description | Attackers can manipulate and control traffic produced by a device once the devices is tricked into connecting to a malicious Wi-Fi with an SSID previously known by the device. After the connection has been established the traffic should be routed to a malicious captive portal |
| Expected threat events | Suspicious App |
| Applicable stages | Stages = 3, 4 |

| | |
|---------------------|---|
| Pre-Conditions | <p>Victim device has previously connected to open Wi-Fi's</p> <p>Victim device Wi-Fi is switched off</p> <p>Attacker has rogue access point running and is actively responding to Wi-Fi probes</p> |
| Post-Conditions | <p>Victim device is connected to the rogue access point.</p> <p>Attacker has full control over traffic flow.</p> <p>Control can be demonstrated by routing traffic for specific websites to the attacker's host (Kali) where a notification page is hosted.</p> |
| Execution Steps | <p>Attacker launches attack by:</p> <ul style="list-style-type: none"> Starting a webserver on his host serving a notification page Configures DNS route for particular URL to the attacker's host in the pineapple access point Enables a captive portal that notifies anyone connecting to the Rogue AP Starts actively responding to Wi-Fi Probes Start actively capturing SSIDs that are being requested <p>Victim switches on the Wi-Fi on the device</p> <p>Device connects to the rogue Access Point</p> <p>Victim gets presented with the captive portal notification and accepts</p> <p>Victim browse particular URL and is presented with the notification page.</p> |
| Possible Variations | <p>SSL MitM Proxy to decrypt traffic with a Pineapple that actively de-auth a device from a connected network</p> |



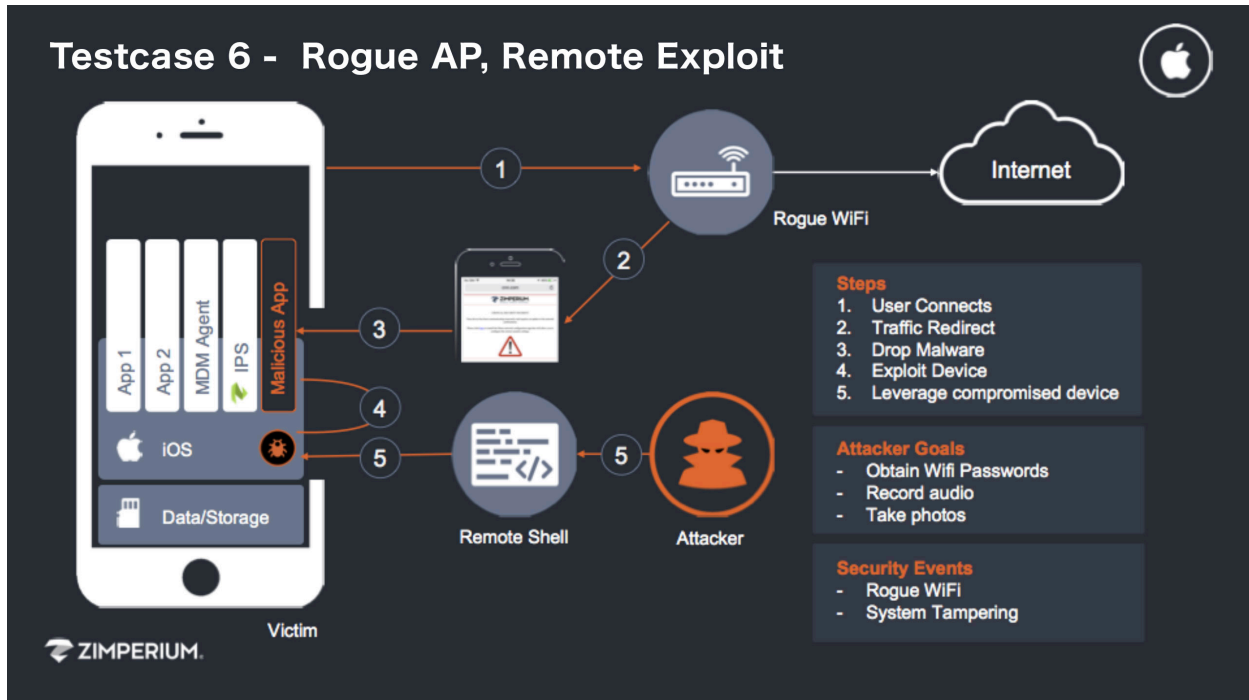
| Testcase Name | Rogue AP, Malicious Profile, SSL MitM |
|------------------------|---|
| Platform | iOS |
| Description | When the user is connected to the Wi-Fi and browses a page, they will be redirected to a page that tricks them into installing a malicious profile that will connect the device to a VPN service. Within the service the traffic will be decrypted in order to obtain sensitive information |
| Expected threat events | <ul style="list-style-type: none"> • Rogue Access point • SSL MitM • Suspicious Profile |
| Applicable stages | Stages = 3, 4 |
| Pre-Conditions | <p>Attacker is actively routing traffic to his own host using either a Rogue AP or ARP MitM</p> <p>No Host-Isolation is present in the Wi-Fi Network</p> <p>No VPN has been established by the device</p> <p>Attacker is connected to the Wi-Fi network</p> <p>Wi-Fi Network is connected to the internet</p> <p>A certificate trusted by the device is installed on the webserver</p> |
| Post-Conditions | <p>Malicious profile is installed on the device</p> <p>VPN is established from the device to an attacker's VPN GW</p> <p>The traffic is fully compromised, and attacker has full control</p> |
| Execution Steps | <p>Attacker Launches attack:</p> <ul style="list-style-type: none"> • Start DNS poisoning • Launch webserver hosting malicious payload <p>Victim connects to the Wi-Fi</p> <p>Victim browses a page and a warning requesting a profile to be installed</p> <p>The victim confirms the installation steps in order to proceed</p> <p>Attacker intercept the traffic to exfiltrate data, inject content</p> |
| Possible Variations | |

Testcase 5 - Rogue AP, Malicious Profile, SSL MiTM



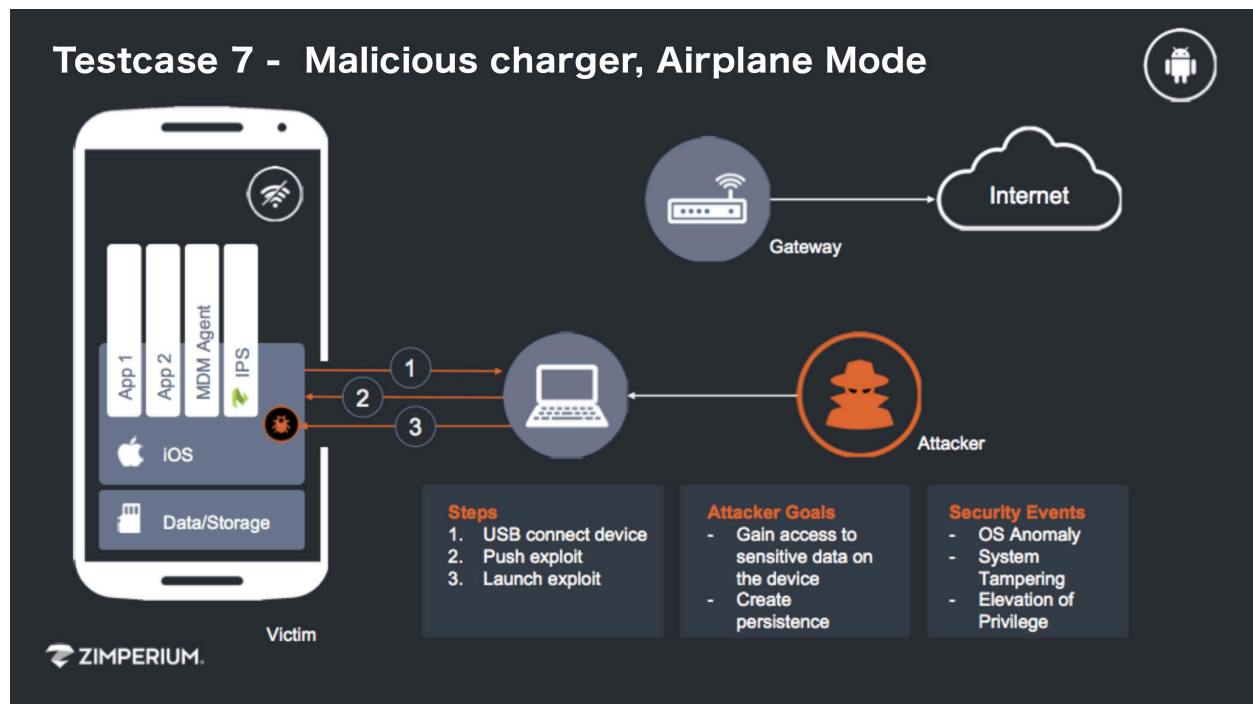
| Testcase Name | Rogue AP, Remote Exploit |
|------------------------|--|
| Platform | iOS |
| Description | A user connects to a Wi-Fi that is hosted by a malicious Wi-Fi access point. When the user browses a page they will be redirected to page displaying a page that will trick the user into installing a malicious application that is undetectable. Once the application runs it will exploit the device and provide elevated access to the attacker that will be used in order to exfiltrate data. |
| Expected threat events | <ul style="list-style-type: none"> • Rogue Access Point • System Tampering / Jailbreak |
| Applicable stages | Stages = 2, 3, 4, 5, 6 |
| Differentiation | zLabs delivered exploit Exploit detection based on behaviors |
| Pre-Conditions | <p>Attacker is actively routing traffic to his own host using either a Rogue AP or ARP MitM</p> <p>No Host-Isolation is present in the Wi-Fi Network</p> <p>No VPN has been established by the device</p> <p>Attacker is connected to the Wi-Fi network.</p> <p>Wi-Fi Network is connected to the internet.</p> <p>A certificate trusted by the device is installed on the Apache server</p> |
| Post-Conditions | <p>Malicious application is installed on the device</p> <p>An exploit is running on the device</p> <p>The device is fully compromised, and attacker has full control</p> |
| Execution Steps | <p>Attacker launches attack:</p> <ul style="list-style-type: none"> • Start DNS poisoning |

| | |
|---------------------|---|
| | <ul style="list-style-type: none"> Launch webserver hosting malicious payload <p>Victim connects to the Wi-Fi</p> <p>Victim browses a page and a warning requesting an App to be installed and launched is displayed</p> <p>The victim taps the link and runs the app after installation is completed</p> <p>Attacker connect to the device exploited device to exfiltrate data.</p> |
| Possible Variations | ... |



| Testcase Name | Malicious charger, Airplane Mode |
|------------------------|--|
| Platform | Android & iOS |
| Description | <p>Malicious chargers are a strong potential way to exploit devices that are connected to them, including now iOS based A12 chip devices susceptible to Checkm8</p> <p>Users of mobile devices can get exposed to these on ex. airports or coffee shops while travelling or when using low cost chargers manufactured in specific geographic locations</p> <p>Attackers general aim to obtain information and access to the connected devices</p> <p>After connecting to the malicious charger, the users notice nothing or that they reboot, but an exploit is pushed to the device after which it is executed. Following this the attacker can exfiltrate data and create persistence in the firmware.</p> |
| Expected threat events | <ul style="list-style-type: none"> • Process Anomaly • System Tampering • EOP • Persistent O/S Modification |
| Applicable stages | Stages = 2, 3, 4 |

| | |
|---------------------|---|
| Pre-Conditions | Victim has accepted the message asking to trust the fingerprint of the connected USB station/device. Developer options & USB debugging enabled 3rd party app stores enabled The device is in airplane mode with Wi-Fi disabled |
| Post-Conditions | The device is fully compromised, and attacker has full control |
| Execution Steps | Victim connect their device to the USB Attacker pushes an exploit to the device Attacker runs the exploit(s) and gains elevated privileges |
| Possible Variations | ... |



MOBILE SECURITY PROBLEM SOLUTION

Zimperium's Mobile IPS (zIPS) client runs on-device for both android and iOS based mobile platforms providing defense via a Machine Learning (ML) agent (e.g. z9). Not only can it provide protection against network-borne attacks, but also protects against persistent attacks, such as those incorporated within Advanced Persistent Threat (APT) techniques, where continued payload execution of malware or malicious code and reconnaissance scanning attempts to either gain device O/S, file or application access are thwarted via the z9 ML agent detects and prevents such attack attempts in real time and protects against Zero-Day threats without the requirement to connect to cloud / server analysis engines and can do so 'without' internet connectivity and works on its own, on-device. All this to ensure that a device protected by Zimperium's zIPS won't become a part of the DDoS attack vector as a drone or botnet.

Additionally, Zimperium's solutions can provide the threat intelligence necessary to provide protection from numerous threat vectors on a device, such as malicious SMiShing (phishing) links via SMS / iMessage through their web-filtering protection and can detect and prevent jailbreak/rooting attempts while assisting to ensure policy enforcement for device integrity can be met. Lastly, Zimperium's solution can ensure airborne Wi-Fi or Bluetooth Man-in-the-Middle (MitM) attacks can be detected and prevented in order to prevent those early stages of an aggressive APT from a Nation State or other sophisticated actors.

Also, while the number of applications allowed onto the devices may be strictly controlled (e.g. *users cannot add or delete applications*) via Apple / Samsung device management solutions or in combination with the EMM suite tools such as, Mobile Application Management (MAM) & StoreFront (MAS); however, to ensure an agency fully understands the risk posture associated with any given app, Zimperium can help exhaustively analyze every behavior and network communication an app exhibits. Every privacy and security issue should be identified through a Dynamic Application Security Testing (DAST) tool to ensure the intent of the app meets security & privacy guidelines and can justify the exposure inherent within it. Zimperium has two different solutions which can help ensure protection of applications and their resident data based on both security and privacy frameworks.

The first, via the aforementioned zIPS client is capable of detecting Self-Modifying Apps, or any malicious "Download-and-Execute" payloads without relying on a signature database, again truly providing zero-day defenses for enhanced security to prevent an allowed application or its data from being compromised or exfiltrated or be used to either disable the device or compromise the data on the device.

Secondly, Zimperium's Advanced App Analysis (z3A) solution helps exhaustively analyze every behavior of a 'public (ex. Google/LinkedIn) or custom' app (e.g. (ex. Pega System's 'Mission App') via its native IPA or APK file from Apple or Android respectively. So apps, prior to deployment via the agency's Mobile Application Management / Storefront or the Apple AppStore would have DAST performed, scanning a host of different privacy and security risks that may be exposed by the app's code. Thus, when an app is determined that it presents a security risk to an irresponsible degree, Zimperium's solutions can then help blacklist that app to ensure it does not deploy to devices with sensitive data or access thereto.

Equally critical to an agency's deployment strategy should be to take into consideration the use of embedding the z9 agent directly into any agency's custom application via the Zimperium zDefend SDK. This embedded detection technology into a custom app provides visibility into runtime manipulation of the app, its memory, its files, or its network communication, thus detecting and defending it in real-time to ensure full protection from data exfiltration or code injection attempts from external sources, whether from network or device-borne attacks. And of course, any activity can be reported to a central security operation center (SOC) for heightened awareness and to trigger potential follow-up and integrated with a SIEM solution, providing a wealth of enhanced mobile threat metrics data for agency security analysts to perform Endpoint Detection & Response & Incident Response.

Lastly, Zimperium's solutions as can also be integrated with the widest range of on-prem and Cloud-Hosted Enterprise Mobility Management (EMM's), such as VMware VSI (A.K.A. *AirWatch*) in the market for advanced device controls and policy enforcement/compliance, as well as be hosted in a variety of methods beyond a commercial-hosted cloud service, which includes some of the most unique and industry-only options, such as on-prem within a customer's data center or private cloud and is the 'first and only' **FedRAMP Authorized SaaS MTD solution** in the market for open FedGov Cloud consumption!

SUMMATION

Zimperium realizes that outages of different data & service systems could cost the government millions of dollars per incident. This is unfortunately quite possible if hackers successfully reverse engineer those custom applications and begin corrupt transactions or use the authenticated connections to create a Denial of Service (DoS) attack against the backend systems themselves. An even greater cost in this threat modeling exercise is if these corrupt transactions are not caught immediately but rather stored alongside legitimate ones upon discover would compromise any faith in the entire database's integrity. And in fact, access to the system could allow data destruction or even manipulation, not just incorrect additions. These scenarios would render all collection up to the date of discovery useless and require starting from scratch. In cases like this a total impact cost would not be possible, considering the near insurmountable task in the required timeframe regardless of the dollars involved.

Considering the sophistication of Nation State Actor(s), their hired guns or hacktivist and their ability to take advantage of numerous tools at their disposal and unknowing hosts of vulnerable systems to launch from, relying on a single, static defense strategy or through training of end-user's is dubious at best. The only true way to ensure protection of both end device's themselves and the resident data on-device / in-Transit or the upstream potential targets within the hosting center and cloud is to envelop the mobile endpoint with an advanced, on-device Mobile Threat Defense from Zimperium that provides protection against data, network, application and phishing-borne attacks.