NowSecure™

INCIDENT RESPONSE FOR ANDROID AND iOS

Table of Contents

Incident Response for Android and iOS

This book will prepare enterprises and practitioners for the inevitable increase in mobile compromise. We will use step-by-step tutorials, guiding the reader from setting up a mobile IR practice all the way through continuous monitoring of mobile devices. Chapters include:

Mobile Incident Response Overview

The goal of this chapter is to provide background on incident response, mobile security challenges and specifics around mobile incident response. The sections include:

A brief history of incident response

On November 2, 1998, Robert Tappan Morris released the first Internet worm (dubbed the Morris Worm) and became part of computer security history by:

1. Unleashing the first large scale denial-of-service (DoS) attack on the Internet
2. Impacting an estimated (and refuted) 10 percent of computers on the Internet at that time (6,000 impacted)
3. Being the first person prosecuted under the Computer Fraud and Abuse Act (CFAA)

The impact to the Internet plus the significant challenges in coordinating a response in such a distributed environment led to the creation of the Computer Emergency Response Team Coordination Center (CERT/CC) by DARPA in 1988.[1] The goal of the CERT organization was to provide the central hub for communicating and coordinating a response to security incidents.

The CERT/CC flourishes today and is part of the Software Engineering Institute at Carnegie Mellon University.

GOAL OF INCIDENT RESPONSE

The goal of incident response is to quickly contain and mitigate an incident. A more formal definition of incident response (IR) will be helpful as we continue on in our discussion:

> Incident response is an organized approach to addressing and managing the aftermath of a security breach or attack (also known as an incident). The goal is to handle the situation in a way that limits damage and reduces recovery time and costs. An incident response plan includes a policy that defines, in specific terms, what constitutes an incident and provides a step-by-step process that should be followed when an incident occurs. [2]

INCIDENT RESPONSE TODAY (2016)

Today, most large government and enterprise organizations have an incident response team. And government regulations increasingly mandate incident response capabilities.

For example, the Federal Financial Institution Examination Council (FFIEC) is a "formal interagency body empowered to prescribe uniform principles, standards, and report forms for the federal examination of financial institutions" [3] that developed a system dubbed InfoBase to help introduce and educate the financial services industry on items field examiners were inspecting during audits. The FFIEC's guidance on incident response includes a mandate to develop and integrate this discipline into the financial institution's business continuity planning process.

This is only one example of regulated industries required to have incident response plans or capabilities in place.

Other examples include:

- The healthcare industry is beholden to a HIPAA incident response regulation that "requires a covered entity to implement policies and procedures to address security incidents"
- The payments industry must comply with PCI DSS incident response requirements [PDF] (section 12.10) that require organizations to "implement an incident response plan," and "Be prepared to respond immediately to a system breach"
- The financial services industry has multiple regulatory acts and bodies overseeing it including the FDIC and Gramm-Leach-Bliley Act incident response standards that "require the service provider to take appropriate actions to address incidents of unauthorized access to the financial institution's customer information, including notification to the institution as soon as possible of any such incident, to enable the institution to expeditiously implement its response program"
- Federal agencies are regulated under the Federal Information Security Management Act of 2002 that includes FISMA incident response regulations [PDF] requiring agencies to develop, document, and implement an information security program (FISMA Act of 2002, Pub. L. No. 107-347, Title III, Dec. 17, 2002).

CERT ORGANIZATIONS

There is a large group of resources that supporting CERTs around the world. They can be loosely organized into three types based on their sponsors:

1. Government
2. Enterprise

3. Community

Below is a list of various organizations that often provide extensive information and at times support during an incident.

Government-sponsored CERTs / CSIRTs

These organizations generally coordinate incident response for entire nations.

1. US-CERT (United States Computer Emergency Readiness Team)
2. ENISA (European Union Agency for Network and Information Security)
3. CERT (CERT Division is located within the Software Engineering Institute, a federally funded research and development center at Carnegie Mellon University, the majority of their work contributes to government and national security efforts)
4. CNCERT/CC (National Computer Network Emergency Response Technical Team/Coordination Center of China)

Enterprise-sponsored CERTs / CSIRTs

These organizations, while focused on a particular enterprise, have a visible public presence and sometimes support the industry with educational information and tools:

1. Microsoft Security Response Center
2. Apple Product Security
3. Facebook Security
4. Google Application Security
5. Android Security

Community

These organizations provide various resources to the incident response community including threat reporting, training, certifications, conferences, and research.

1. FIRST (global Forum for Incident Response and Security Teams)
2. SANS Institute
3. Internet Storm Center (part of SANS)

The FIRST website maintains a contact list of incident response teams at participating member organizations. Please note, leveraging this list for marketing is strictly prohibited.

DIGITAL FORENSICS AND INCIDENT RESPONSE

Incident response is closely connected with digital forensics as nearly every incident requires the collection, storage, and analysis of digital evidence.

As you explore mobile incident response more thoroughly, below are some mobile forensic resources what may be helpful:

- Linux for Mobile Forensics - free training
- JTAG for Mobile Forensics - free training
- Santoku Linux - free Linux distro for mobile forensics

FOOTNOTES

1. https://en.wikipedia.org/wiki/Morris_worm ↩

2. What is incident response? Definition from WhatIs.com. Wed. Wed Nov 11 2015. http://searchsecurity.techtarget.com/definition/incident-response. ↩

3. http://ithandbook.ffiec.gov ↩

The case for mobile incident response

Use of mobile devices has sufficently penetrated enterprises to warrant including support for mobile in an incident response strategy. Mobile devices have access to sensitive data and impact the operations of an enterprise. And, like all technology, they have security flaws which expose the enterprise to risk.

While most enterprises have some form of incident response plan in place, very few have developed processes and tools to respond to a mobile incident. This is a clear gap that security teams must address.

In this section, we will demonstrate that mobile apps and devices:

- are the focus of government regulation and law enforcement
- possess significant security and privacy flaws
- are the target of cyber-criminal and nation state attacks
- receive very little security focus and investment from enterprises

REGULATION AND LAW ENFORCEMENT

In the previous section, we explored a few examples of regulations that cover incident response. However, smartphones are a relatively new technology and at best are addressed broadly in a few of these regulated industries. This means that the industry is largely self-regulated today and, unfortunately, mobile device and app security issues abound.

If these issues persist and become a larger problem, it will ultimately lead to regulation and law enforcement. This is something the mobile industry should strive to avoid by addressing these issues within the ecosystem. This can be achieved far more quickly and efficiently by the industry itself than by government regulation.

FTC v. Wyndham

In 2008, Wyndham Worldwide Corporation experienced several security incidents resulting in the loss of credit card data. The impact of this data breach was over $10.6 million in fraud loss. The Federal Trade Commission (FTC) filed a complaint in federal court against Wyndham under Section 5 of the FTC Act for unfair methods of competition.

Wyndham fought the FTC suit, but the district court judge ruled in favor of the FTC. Wyndham appealed the decision but in August 2015, the U.S. Court of Appeals for the Third Circuit affirmed the district court, upholding the FTC's data protection authority.

Judge Thomas Ambro stated:

> A company does not act equitably when it publishes a privacy policy to attract customers who are concerned about data privacy, fails to make good on that promise by investing inadequate resources in cybersecurity, exposes its unsuspecting customers to substantial financial injury, and retains the profits of their business. [1]

While this ruling does not directly involve mobile devices or apps, it is significant for several reasons:

1. Mobile apps are required to publish a privacy policy through multiple mechanisms including the terms of service for Apple's App Store and Google Play as well as various privacy laws including California's Online Privacy Protection Act and the European Union's Data Protection Directive (95/46/EC);
2. A compromise involving a mobile device or app would likely trigger the "investing inadequate resources in cybersecurity" unless the company performed security testing and had an incident response policy in place to minimize the impact of any breach.

The combination of the FTC v. Wyndham ruling plus a widespread lack of investment in mobile security does not bode well for enterprises benefiting from mobile devices and apps.

MOBILE DEVICES AND INCIDENT RESPONSE TRENDS

Over the past three decades, the process of incident response has matured. Bruce Schneier, a respected security technologist, wrote about evolution and the future of IR in his popular Schneier on Security [2] blog. He identified broad focuses in each decade since the 1990s starting with protection, then moving into detection and finally focused on response by the 2010s.

He reflected that in recent years, new IR products and services are being developed and implemented due to three important trends:

1. Devices and data now regularly reside outside the control of IT departments (as a result of mobile, cloud computing, etc.)
2. Attacks and threats are far more sophisticated and effective
3. Companies continue to under-invest in protection and detection, increasing the need for response

Each of these trends are very much a reality in the mobile ecosystem and worth exploring further.

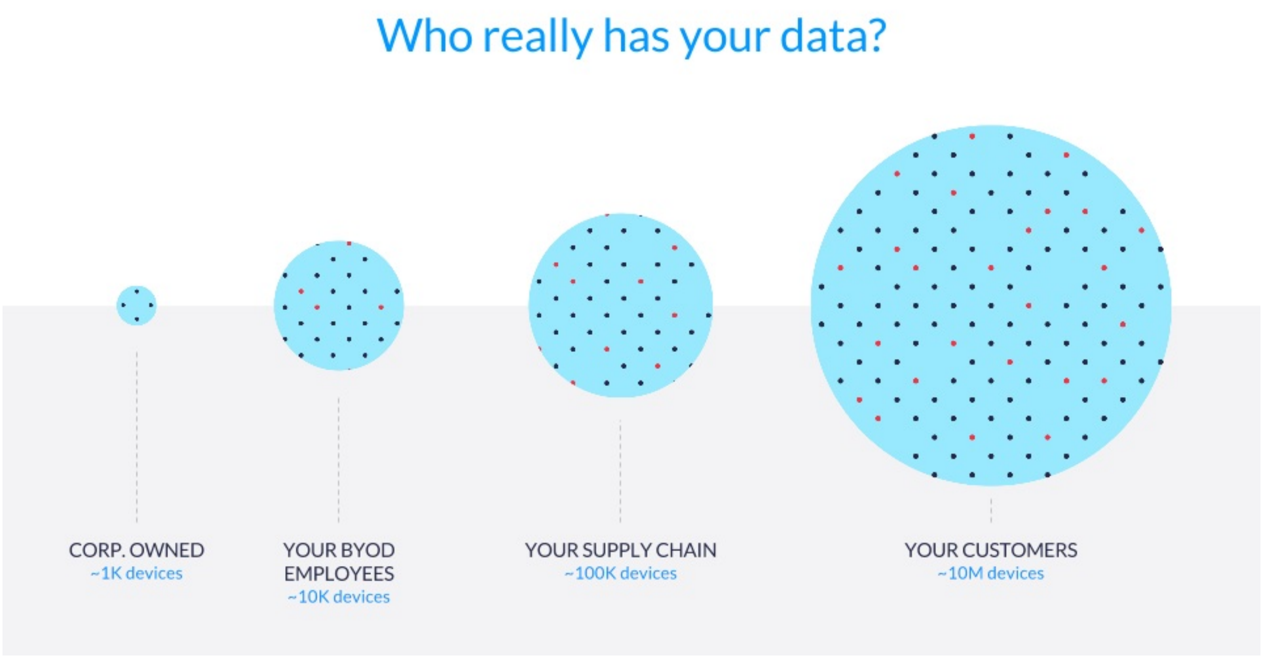
CONTROLING MOBILE DEVICES AND DATA

Mobile technology, by its very definition, lives outside the traditional definitions of IT boundries such as local networks and firewalls. This dramatically impacts the enterprise's ability to control both the devices and, perhaps more importantly, the data residing on them.

Device ownership and data storage

Increasing, the mobile devices which impact an enterprise's security are neither purchased nor managed by the IT department. Some quick examples of this include:

- Employees via the bring-your-own-device (BYOD) trend
- Contractors
- Vendors and supply chain
- Customers



While the recent trend in enterprise mobile security has been to deploy management software such as mobile device management (MDM), enterprise mobile management (EMM) and similar in BYOD scenarios, this is clearly a strategy that will not scale to an enterprise's vendors, supply chain and customers.

Network perimeter

Clearly, mobile devices defy traditional network policy enforcement since they do not exclusively reside on the corporate network. By design, mobile devices likely have a least two and generally more ways they can connect to networks.

The obvious network connection not directly controlled by the enterprise is the mobile phone network. Today 4G networks are prevalent and fast and many individuals choose to use their mobile operator's network over corporate-provided Wi-Fi.

The next most common accessible networks are provided via Wi-Fi. This generally includes a network provided by the enterprise (and thus a controlled ingress/egress point where a company can exert oversight or control) but also includes home networks and various networks available to the public. These public networks include Wi-Fi access points provided by retail stores (e.g., Starbucks), airports, city-wide public networks, home-based Wi-Fi via companies (e.g., Comcast via their XFINITY service), and, unfortunately, malicious networks controlled by attackers.

While cellular and Wi-Fi networks are the most common paths to network access, there are additional techniques including:

- Bluetooth networks
- Tethering over USB
- Near field communication (NFC)
- GPS (a form of radio-based network traffic)

There are many ways mobile devices can connect to networks and very few of them provide enterprises with control over the traffic. While some IT departments and mobile security solutions attempt to backhaul mobile traffic over a device or per-app VPN, this technique will ultimately fail due to a number of causes discussed in the next section.

Privacy implications

There is a growing backlash from device owners regarding mobile device management tools and techniques and their impact on privacy. In some instances, the legal teams at large enterprises are also pushing back against solutions which intercept or collect significant personal information as it becomes a liability.

The biggest challenges are best examined by exploring the effects on a single technique. For this example, let's examine how a device or per-app VPN impacts a mobile device:

1. Initial setup is cumbersome
2. Re-connecting to the VPN is frustrating and time consuming
3. A VPN slows down many network connections, especially on a higher latency mobile network
4. Additional battery drain is incurred
5. Personal privacy is significantly impacted
6. Enterprises have access to sensitive employee data (e.g. Internet searches, app traffic and geo-location), which can place significant liability on the enterprise
7. These solutions are not possible to implement outside of devices used by employees and perhaps contractors (though the later is unlikely)

The trend in mobile devices and applications is clearly moving toward an ecosystem where IT and security departments have very little control over devices.

EFFECTIVENESS OF ATTACKS AND THREATS

As mobile devices and apps proliferate, cyber-criminals and actors focused on espionage (with both intellectual property and national security objectives) are clearly following the trend. They recognize that a significant amount of the data they target is now present and at times more accessible to them on mobile devices.

While early mobile attacks were fairly trivial and frankly lazy, recent trends have revealed a more sophisticated and sustained effort to thwart mobile defenses. While we will explore this topic thoroughly in the Attacking Mobile Devices chapter, it is helpful to provide several examples here.

Mobile threats

Let's start off looking at mobile threats. This category includes vulnerabilities in mobile hardware, operating systems and apps that an attacker could exploit.

Educating enterprise decision makers about the risks of mobile threats and attacks is a critical first step to addressing the issues. However, mobile attacks are under-reported for several reasons:

1. Due to lack of visibility, many individuals and enterprises are not aware an attack has occurred
2. In server-side attacks, enterprises may be compelled to disclose the breach due to consumer protection laws, however, it is unlikely a targeted mobile attack would trigger such action

For these reasons, the most effective resources available today to educate individuals and enterprises about mobile risks are identifying the threats which could expose them to compromise.

ANDROID SECURITY THREATS

Android encompasses a deeply intertwined ecosystem including Google/Android, original equipment manufacturers (OEMs), wireless carriers and app developers. In this brief overview, we will provide examples of several publicly known Android security threats.

Andorid Stagefright vulnerability

In September 2015, Zimprium researcher Joshua Drake (@jduck) disclosed a vulnerability impacting nearly all Android devices since Android version 1.5 and could allow an attacker to remotely execute code on the device. The flaw was discovered in the Stagefright library, a librady of shared code used by Android devices to process media files. By employing fuzzing techniques, it was discovered that specifically crafted media files including images, audio and video files sent to the device would crash libstagefright and provide the attacker with the ability to compromise the device.

Joshua coordinated with the Android Security team and patches were submitted to the Android Open Source Project (AOSP) and shared in advance with Android partners. However, the patches were not fully effective in mitigating the flaws and additional research uncovered new attack vectors. In addition, due to the complexity in deploying changes to the fragmented Android ecosystem, most users remain vulnerable to this day.

Note: if you own or manage risk for Android devices, you can determine if the device is vulnerable to Stagefright or other know flaws by running the Vulnerability Test Suite (VTS) for Android app. The app, developed by the NowSecure Research Team, is fully open sourced and actively maintained.

Samsung keyboard vulnerability

In June 2015, NowSecure researcher Ryan Welton (@Fuzion24) presented at BlackHat London [4] and exposed a serious flaw in over 600 million Samsung devices. [5] Despite following a responsible and coordinated disclosure process over the course of nine months, the flaw was ultimately misunderstood by Samsung and at the time of the disclosure, no patch existed that users or enterprises could apply.

To make matters worse, the insecure application was signed with system privileges granting the app, and thus the attacker, significant access to the device and data through remote code execution. The app could not be disabled and regularly checked for updates over the network, the trigger necessary to launch the attack. Anyone with a position on the network between the endpoint and the SwiftKey update server could execute this attack.

APP EXAMPLE

The Google Play store had 1.6 million apps available for download as of July 2015. [10] Many of those apps contained security flaws, and so it's difficult to choose just one example.

In January 2016, Jake VanDyke spent a week reviewing video cameras and the mobile apps that can remotely control them. After examining some of the most popular cameras on Amazon, Jake noted that "every camera-and-app combination I tested included at least one security flaw," and that concerned him. The various apps exhibited numerous flaws including:

- Sensitive data transmitted without encryption including username, password and location
- Unencrypted communications allowing someone to adjust camera settings, format the SD card, access stored photos and videos, and initiate the recording of audio or video
- Sending the WPA2 key for the network to which the camera was connected
- Vulnerabilities to man-in-the-middle (MITM) attacks

Hopefully it's clear from this single example that security flaws in mobile apps create significant exposure and risk for not only the individuals using the apps but also the enterprises they interact with and work for.

IOS SECURITY THREATS

Apple is incredibly effective in how they position and market the security of iOS devices. And there are indeed many excellent security features built into the platform. However, Apple developers are just as susceptible to creating security flaws as any other developer.

A key difference between Apple's approach to security and Google's approach with Android is the availability of core Android source code (via the AOSP) for community inspection. This open approach has resulted in more flaws being identified and patched at the core of Android. However, when examining iOS code, security researchers have to employ different techniques, often requiring more time and effort.

Based on Apple's effective marketing, and the lack of open source code available for inspection, many people in the IT industry believe that Apple is more secure. However, the the real answer is far more nuanced but can be quantitatively analyzed by examining publicly available vulnerability data.

Apple iOS CVEs

The Common Vulnerablilty and Exposure (CVE) system maintained by MITRE was designed to provide visibility into flaws exposed in IT systems. You can filter the data by vendor and operating system which provides visibility into the quantity and category of CVEs for Apple iOS and Android:

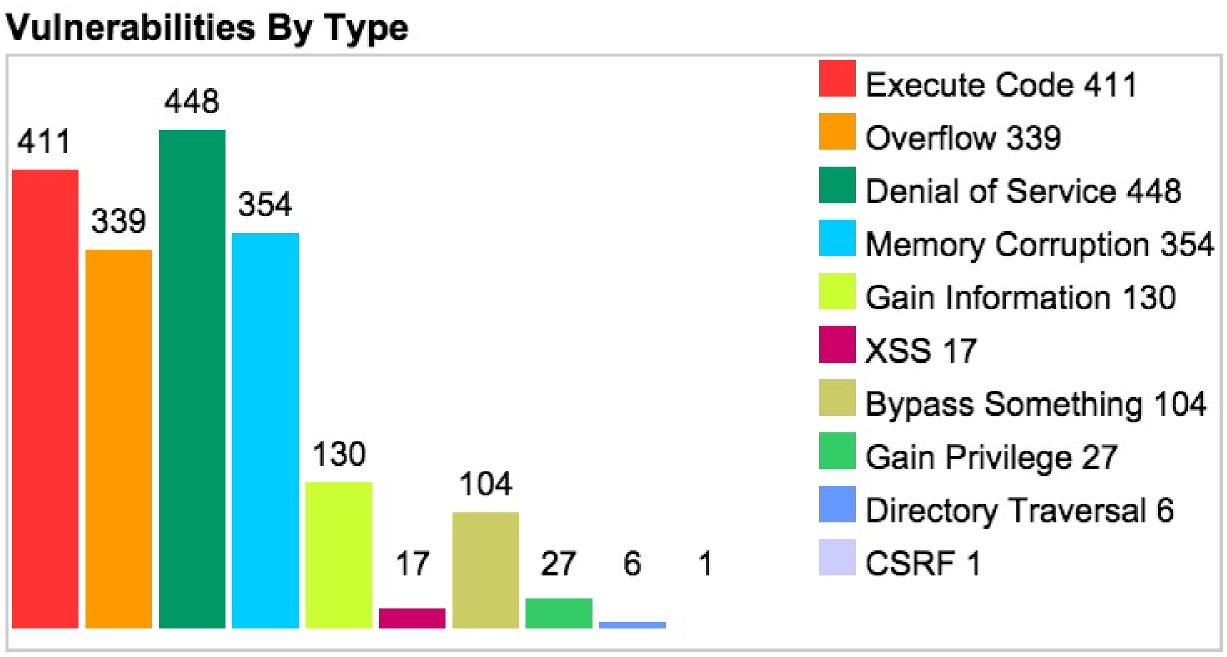


Figure 1: iOS CVEs from 2007-2016

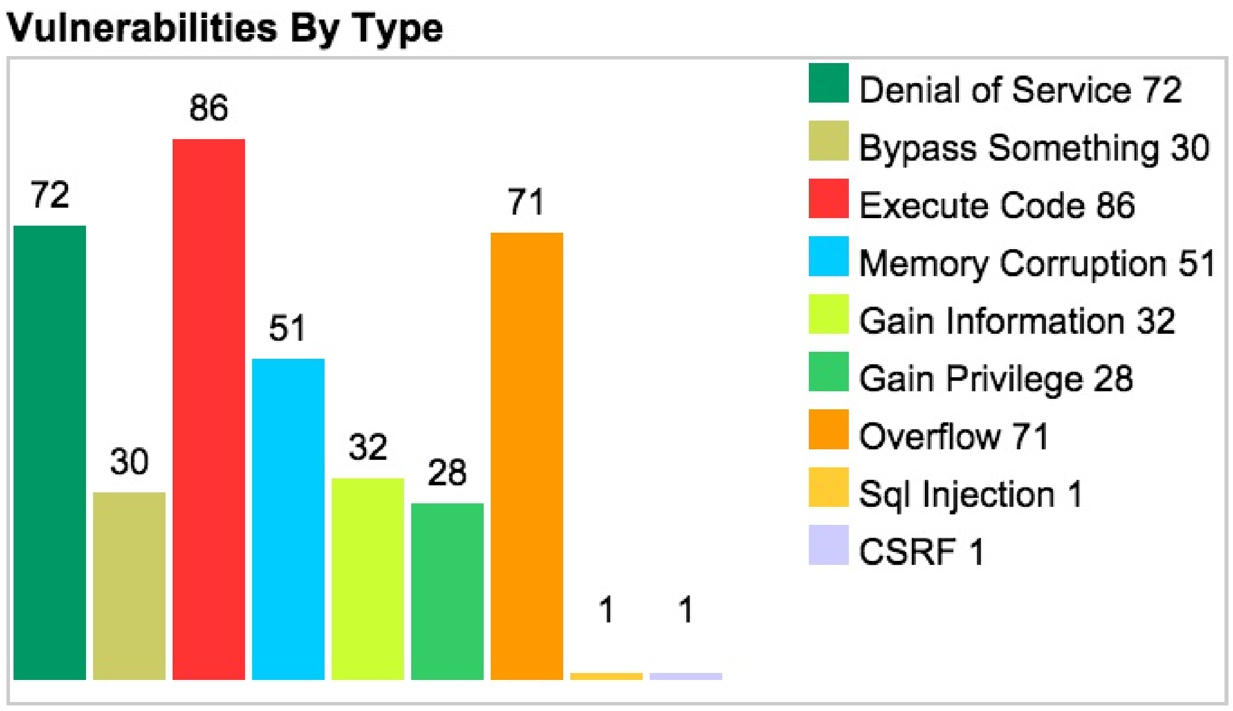


Figure 2: Android CVEs from 2009-2016

Clearly Apple developers fair no better than any other developer in writing secure code. And the closed nature of their system means far fewer eyes are available to inspect and find security flaws. If your strategy for mobile security is to simply rely on the Apple iOS platform, you need to explore additional layers of protection.

Apple's iOS 9 security updates

It is also revealing to examine Apple's own security page for the flaws they patch in each new iOS version. I applaud Apple and other technology companies for their transparency in posting these updates, but it seems to contrast with the perception of the security of iOS.

In particular, in the initial iOS 9 release, Apple patched over 70 security flaws, many of them quite serious. There is also a history of Apple releasing a follow-up update very quickly after a major release to patch serious security flaws found by researchers in the new update. For example, Apple has had over 10 PIN-bypass flaws over the years, a vivid example that many users quickly understand. The ability for an attacker to circumvent the lock screen on iOS devices broadly exposes the device and data to compromise and exfiltration.

iPhone 4 hardware flaw

While most identified flaws occur in software (both the operating system and mobile apps), at times flaws are identified in the hardware of a device. This issues are extremely dangerous as there are generally no mitigations against these types of flaws.

It has been quite some time since a flaw like this was found in an iPhone, however, it is worth pointing out an instance of this from the past. When Apple manufactured the iPhone 4, there was a flaw in the boot process that allowed an attacker to boot an unverified boot disk. This allowed any attacker with physical access to boot the device into a modified version of iOS that disabled the pass-code and allowed full access to the operating system and the data on the device.

This flaw was fixed in the next release of the iPhone, but anyone using that device was at risk and had very few, if any, mitigation actions they could take.

Considering the history of CVEs in iOS, the sheer amount of new code that goes into each release, and the ever increasing complexity of the device and operating system, iOS will continue to have security flaws which, if exploited, place the individual and enterprise at risk.

iOS app security

Similar to Android, a significant part of the iOS ecosystem is the mobile apps available from the App Store (1.5 million apps as of July 2015)[10]. Interestingly, iOS apps tend to be less secure than Android apps based on a high-level security scan of 100 popular apps. Such a small sample size prevents me from extrapolating this to the entire ecosystem, but it is certainly true for the popular apps tested.

One theory is that iOS app developers place more trust in the smart phone's operating system than Android developers do. However, while iOS provides many strong security features, it cannot prevent developers from making common mistakes like sending or storing sensitive data without encryption, failing to adequately perform certificate checking, or ignoring other common mobile app security best practices.

Mobile attacks

Unlike mobile threats, mobile attacks identify actual instances of mobile threats being exploited in the wild. While these examples are less frequent, they make the strongest case for enterprises to invest in mobile security and incident response. These attacks fall broadly into these categories:

- Known malware
- Targeted attacks
- Weaponizations of mobile threats

In the Case Studies chapter we will provide deeper examination of real world mobile attacks. However, it is useful to provide examples of several known mobile attacks.

MOBILE MALWARE

Mobile malware is the most common form of mobile attack, in particular because they are far easier to identify by examining mobile apps. While all mobile attacks are undesirable, mobile malware is the most benign of the categories listed above. We will provide an example of malware on both iOS and Android.

Apple iOS XCodeGhost attack

In late 2015, researchers from Palo Alto Networks uncovered modified versions of XCode, the development environment for iOS, available on multiple websites in China. XCode is a large download and developers in China would often choose cached versions of XCode hosted in China to decrease download times.

Attackers seized this opportunity and modified XCode so that when developers compiled and ultimately deployed thier applications to the App Store, malicious code was automatically included in the applications. The malicious code would exfiltrate private data from the iOS devices upon which the app was installed and run.

Apple identified the affected apps and immediately removed them from the App Store. They published a list of affected apps on their website and worked closely with those apps' developers to ensure new versions were pushed to the App Store quickly. Some of the apps are extremely popular and the full impact of the attack is likely not yet understood.

Android SimpleLocker malware

The Android ecosystem has been a larger target of mobile malware than any other smart phone to date. As such, choosing a single example malware to discuss is difficult. In our Mobile IR Case Studies section, we will explore the outbreak of a malicious app targeting customers of Aetna, an insurance provider, and how the incident was handled. For this overview, though, we will discuss a piece of malware that falls into the "ransomware" category.

In June 2014, a number of security companies (Sophos | Blue Coat) began reporting on and analyzing a newly discovered instance of malware dubbed SimpleLocker.

The initial attack vector appeared to be fake porn sites that prompted the user to download and install a video player.[9] After installation, the app launched and immediately took over the screen. It then presented a threatening message demanding payment to decrypt the user's files. The back button was disabled, and if the user hit the "home" button, the app rapidly re-launched. During this time, the app encrypted images (PNG, JPEG) and plain text files (TXT) within the device's media storage. It also leveraged the TOR network for its command-and-control (C2) traffic.

We will study this malware more closely in the Lab Exercises included in this book.

WEAPONIZING MOBILE THREATS

While security problems in mobile apps and operating systems, as well as malicious apps are grounds for concern, a more nefarious threat exists. It is generally accepted that it is nearly impossible to prevent a targeted attack against an individual or organization. As such, any evidence of a malicious actor enabling or performing targeted attacks is a area of great concern.

Zerodium's one million dollar iOS 9 bug bounty

Zerodium is a privately held, venture-backed company that positions itself as "the premium acquisition program for zero-day exploits and advanced cybersecurity research."[11] On September 15, 2015 they announced a $1 million bug bounty for iOS. The bounty would be paid out for any individual or team that "creates and *submits to ZERODIUM an exclusive, browser-based, and untethered jailbreak for the latest Apple iOS 9* operating system and devices"[12]. While this alone should cause significant concern given the size of the bug bounty and the specific criteria, what's truly disturbing is that on November 1, 2015 Zerodium acknowledged that one team had won the prize.[11].

Of course, the natural question is what does Zerodium then do with the zero-day? According to their FAQ, they:

> analyze and document the vuln, and provide that documentation along with protective measures and security recommendations to clients as part of their Security Research Feed[13]

Zerodium also answers the next logical question of who their clients are: "ZERODIUM customers are major corporations in defense, technology, and finance, in need of advanced zero-day protection, as well as government organizations in need of specific and tailored cybersecurity capabilities."

Of particular concern is the reference to government organizations and tailored cybersecurity capabilities, which likely means targeted attacks. While it would be naive to assume government agencies are not involved in such activities, the risk organizations of major corporations should take note of Zerodium's public acknowledgment of this capability.

Finally, it's quite telling to see the pay-out ranges for different types of exploits and systems. In particular, mobile devices are positioned in the highest pay-out category.

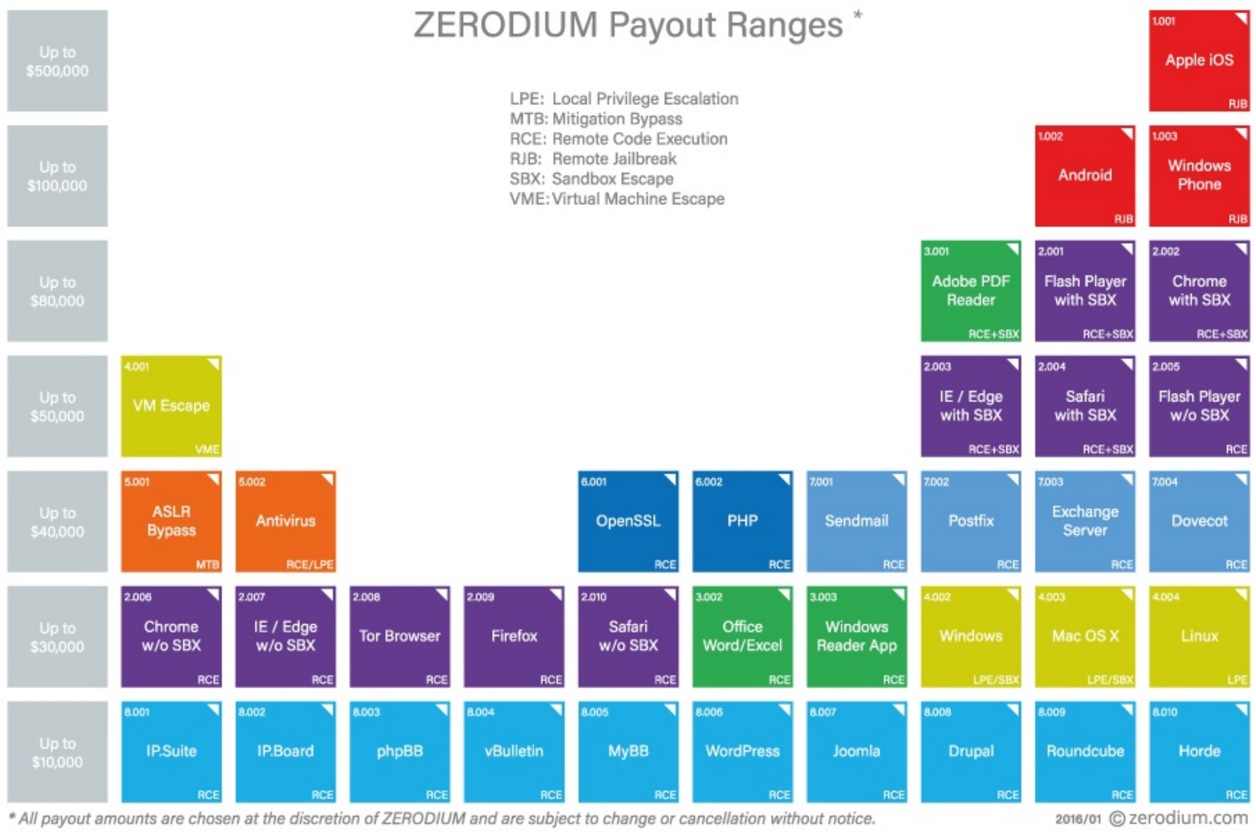


Figure 3: Zerodium Payout Ranges[14]

Hacking Team

While Zerodium is in the business of buying and then re-selling exploits, we do not have a clear view into their internal operations nor their customers. However, in July 2015 the compromise of Italian firm Hacking Team exposed not only the internal operations of a company weaponizing and selling exploits but also their customer list. That list included sovereign nations with documented human rights violations against reporters and activists. Until the time of the compromise, Hacking Team adamantly denied they sold their software to any countries with documented human rights violations. However, the compromise and exposure of 400GB of Hacking Team emails and files revealed that Hacking Team was in fact weaponizing mobile security flaws and selling them to governments around the world.

We explore the Hacking Team case study in more detail later, but a key takeaway for security professionals is that attackers see value in targeting mobile devices and are capable of compromising them. In addition, there are companies out there whose whose business model is based on selling "offensive intrusion and surveillance capabilities to governments, law enforcement agencies and corporations." [8]

UNDER-INVESTING IN SECURITY

In the final incident response trend pointed out by Bruce Schneier, companies are clearly under-investing in mobile security, which greatly increases the need for effective incident response.

In an IBM sponsored study in 2015, the Ponemon Institute found that:

> Among the more than 400 organizations studied — nearly 40 percent of which were Fortune 500 companies — almost 40 percent of them aren't scanning the code in their apps for security vulnerabilities, leaving the door wide open to the potential hacking of sensitive user, corporate and customer data. The average organization tests fewer than half of the mobile apps it builds, and a whopping 33 percent of companies never test their apps. [7]

Instead of taking a survey, NowSecure released their 2016 Mobile App Security Study, which performed aggregated analysis of more than 140 million mobile security data points collected each day by the NowSecure Intelligence engine and evaluated 400,000 Android apps available on the Google Play store. Key findings from the study include:

- 24.7 percent of mobile apps include at least one high risk security flaw
- The average device connects to 160 unique IP addresses every day
- 35 percent of communications sent by mobile devices are unencrypted
- Business apps are three times more likely to leak login credentials than the average app
- Games are one-and-a-half times more likely to include a high risk vulnerability than the average app

The case for mobile incident response

The objective of this section was to outline the need for mobile incident response. Mobile devices have deeply permeated all facets of the enterprise and exhibit unique characteristics that necessitate a strong incident response capability. These characteristics include:

- Government regulatory and law enforcement bodies are beginning to require and enforce mobile security
- Mobile devices and data are increasingly outside the control of IT
- Cyber-criminals and nation states are targeting mobile devices
- Mobile apps and devices harbor a large number of security and privacy flaws

- Enterprises continue to under-invest in mobile security

FOOTNOTES

1. http://www2.ca3.uscourts.gov/opinarch/143514p.pdf ↩

2. The Future of Incident Response - Schneier on Security. Web. Wed Oct 21 2015. https://www.schneier.com/blog/archives/2014/11/the_future_of_i.html. ↩

4. Black Hat London 2015 | Summit. Web. Wed Oct 21 2015. https://www.blackhat.com/ldn-15/summit.html#abusing-android-apps-and-gaining-remote-code-execution. ↩

5. NowSecure. Samsung Keyboard Security Risk Disclosed: Over 600M+ Devices Worldwide Impacted | NowSecure. Web. Wed Oct 21 2015. https://www.nowsecure.com/keyboard-vulnerability/. ↩

7. https://securityintelligence.com/mobile-insecurity/ ↩

8. https://en.wikipedia.org/wiki/Hacking_Team ↩

9. http://securitywatch.pcmag.com/mobile-security/324666-mobile-threat-monday-android-ransomware-encrypts-your-files-don-t-pay-up ↩

10. http://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/ ↩

11. https://www.zerodium.com/ ↩

12. https://www.zerodium.com/ios9.html ↩

13. https://www.zerodium.com/faq.html ↩

14. https://www.zerodium.com/program.html ↩

Incident response process

The SANS Institute has defined six key IR steps in their book Computer Security Incident Handling: Step-by-Step:

1. Preparation
2. Identification
3. Containment
4. Eradication
5. Recovery
6. Lessons learned

While we will cover this topic extensively in the chapter Framework for Mobile Incident Response and build upon frameworks from both SANS and NIST, we'd like to discuss this process here at a high level and relate it to mobile specifically.

PREPARATION

Preparing for a mobile incident involves a number of steps. First, in order to determine the risks that exist within an organization today, a mobile threat assessment should be performed. Part of this threat assessment involves identifying mobile assets on the corporate network. Mobile assets include devices, operating system versions running on those devices, and applications installed on those devices if available. This inventory should then be correlated with mobile security intelligence. That intelligence will include device vulnerabilities, operating system vulnerabilities, information about leaky and insecure apps, known malware, and other risks such as known malicious Wi-Fi networks. The last step of the mobile threat assessment is mitigating the risk identified as a result of your inventory and correlation with threat data. Once you've collected intelligence from the mobile devices on your network, analyze that data to identify security risks, eliminate low hangout fruit, address risk that is unacceptable to your organization, document remaining risks, and prepare IR playbooks for each scenario.

Building your mobile IR tool box is another criticial step in the preparation process. When an incident occurs, your team needs to be prepared with a list of tools that are specific to mobile. Chapter 2 - Tools for Mobile Incident Response - goes into detail about different categories of mobile IR tools, lists suggested tools, and provides instructions on how to setup a mobile IR workstation.

Finally, you cannot fully be prepared without defining and documenting a process, and developing playbooks for each type of mobile threat scenario. Once you've developed your documentation and prepared a workstation with pre-installed tools, preparation should not stop there. The very best way to be prepared for an incident is practice and repetition. In a later chapter, we will provide you with sample scenarios where you can walk through some practice labs.

IDENTIFICATION

Identification of an incident can occur in several ways. There are varying device indicators of compromise (IOC) depending on the type of mobile incident. In some cases, a system administrator may observe increased battery drain, unusual network traffic, or certificate errors. Many times, a user may report an incident when noticing something strange occurring on their device.

Another type of mobile incident may be indentified by mobile app reputation services (MARS). MARS scour app stores (official and third-party) and the Internet for mobile apps using a company's brand name. If an organization is alerted to unauthorized use of their brand in a mobile app or sees unknown apps connecting to their transactional servers, this might indicate an incident relating to mobile apps.

CONTAINMENT

Once you have identified and logged an incident, it must be contained. It is preferable to have physical access to the device, which can be a challenge with mobile. If access to the device is obtained, baseline information should be captured including type of device, operating system version, and a list of installed apps. If appropriate, network analysis should also be considered. The full forensic acquisition of the device should then be performed before containing the incident by isolating the device from the network.

ERADICATION

Artifacts collected from the mobile device, router, or network packet capture must then be analyzed in an effort to determine whether the threat can be removed. At this step, it is important to identify all impacted users or devices, remove the threat, and/or wipe corporate data if necessary.

RECOVERY

Steps should then be taken to bring any systems that were shut down during the incident back online. This includes re-provisioning mobile devices. The recovery process should also include ensuring attacker didn't move laterally within your organization and pro-actively monitoring accounts and systems connected to the impacted mobile device and impacted users. During this phase, the effectiveness of social engineering attacks is greatly increased, so ensure that your employees are properly educated and trained in this area.

LESSONS LEARNED

The final step in the IR process is just as important as those before it. The purpose of the lessons learned phase is to summarize what went wrong, what worked, and most importantly, what can be improved. A debrief with the team should take place to identify recommended policies and procedures changes and user education. The team should discuss the indicators of compromise and determine how to best inoculate against future attacks by focusing on anomaly detection as well as shared insights and cross-referencable data available publicly or from other organizations.

Differences between computer and mobile incident response

In general, an incident response framework should not need to change to address new technologies. However, the unique properties of mobile devices and operating systems do present certain challenges when responding to a mobile incident.

Mobile incident response differs from traditional network- or computer-based incident response for a number of reasons:

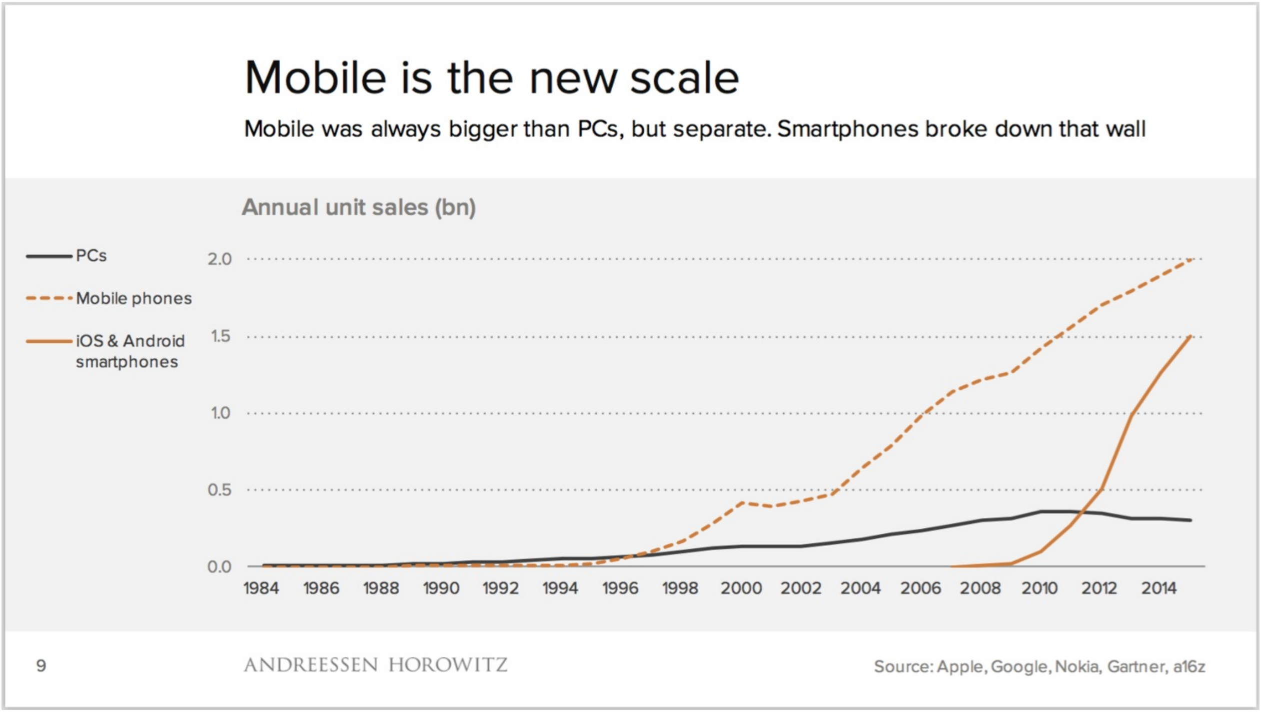
- Mobile devices are built for individual consumers
- Mobile's explosive growth
- Mobile devices are always powered on and connected
- A broad attack surface
- A lack of administrative access to mobile devices
- The implementation of app sandboxes
- App store models

MOBILE DEVICES ARE BUILT FOR INDIVIDUAL CONSUMERS

Manufacturers build mobile devices primarily for individual consumers, unlike computers and servers. Because managing a large number of devices doesn't usually concern the majority of individual consumers, most devices do not include important IT and security management functions. Without some of these basic management capabilities, a lot of the data incident responders might count on in a computer-based incident is not available to them after a mobile incident. This fact underscores the importance of conducting a mobile threat assessment so that you can baseline mobile devices that connect to corporate assets and identify anomalies.

EXPLOSIVE GROWTH

In 2014, analyst Benedict Evans gave a presentation entitled, "Mobile is eating the world." Below is a slide from that presentation showing that the smartphone industry now dwarfs the computer industry.



` Figure 1: Mobile, ecosystems and the death of PCs

Such explosive growth left the security industry behind trying to pick up the pieces. The industry didn't have time to mature, and this failure to keep pace was amplified by a completely new architecture and limitations created by operating systems’ sandboxes. In addition, a traditional approach to endpoint security (i.e., focused on malware) is less effective when applied to the mobile ecosystem. Mobile security failing to progress at the same speed as mobile usage grows results in a lack of visibility into mobile risk, how to manage it, and how to detect a mobile incident.

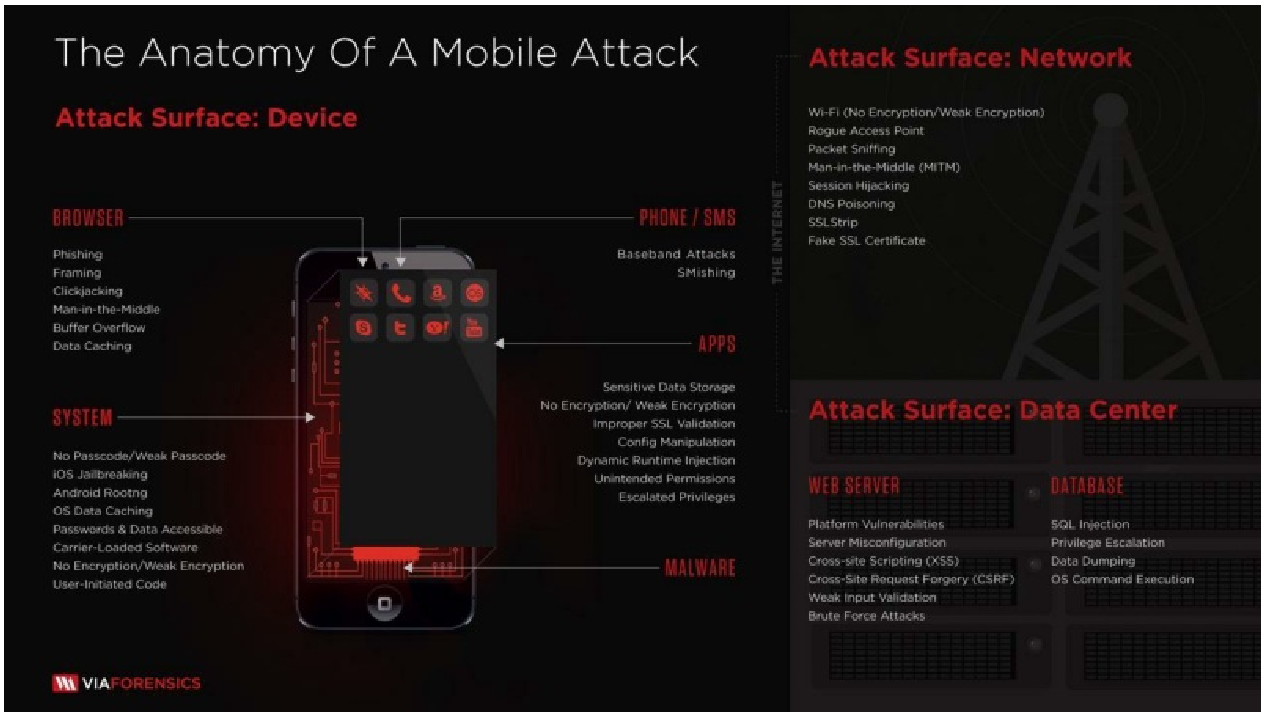
ALWAYS POWERED-ON AND CONNECTED

Unlike laptops and desktops, mobile devices are almost always connected to the Internet. According to a commissioned study conducted by Harris Interactive, 72 percent of mobile users are within five feet of their devices most of the time. Mobile devices also connect to cellular networks, Wi-Fi networks, and other technologies such as Bluetooth automatically. As documented in the 2016 NowSecure Mobile Security Report, the average mobile device connects to 160 IP addresses located all over the world each day, and 35 percent of the data transmitted via those connections is unencrypted. In addition, half of mobile devices in the U.S. connect to insecure Wi-Fi networks each day.

This adds up to a whole lot of mobile risk, and a whole lot of avenues that an incident responder may need to explore in responding to a mobile incident.

BROAD ATTACK SURFACE

The mobile ecosystem also presents a broad attack surface resulting from a wide user base, rapid development, and continuous connectivity. Mobile attacks can occur at the device layer, the network layer, the data center, or a combination of these.



Below we've provided a non-exhaustive list of areas of exposure within mobile and possible attacks sorted by device, network and data center. The list was originally published in NowSecure's collection of Secure Mobile Development Best Practices, and the information below is a condensed overview. We will take a closer look at the details later in the book.

- Device layer
    - Browser, mail, or other preloaded applications
        - Phishing
        - Framing - delivery of a Web/WAP site in an iFrame
        - Clickjacking
        - Drive-by downloading
        - Man-in-the-mobile (MITMo)
    - Phone/SMS
        - Baseband attacks

        - SMiShing
        - Radio frequency (RF) attacks (e.g., baseband, Bluetooth and other channels)
    - Third-party apps
        - Insecure storage of sensitive data
        - No encryption/weak encryption
        - Improper SSL validation
        - Configuration manipulation
        - Dynamic runtime injection
        - Unintended permissions
        - Escalated privileges
    - Operating system
        - No passcode
        - iOS jailbreaking
        - Android rooting
        - Passwords and data accessible
        - Carrier-loaded software
        - Zero-day exploits
- Network layer
    - Wi-Fi (weak encryption/no encryption)
    - Rogue access points
    - Packet sniffing
    - Man-in-the-middle (MITM)
    - SSL strip attacks
    - Session hijacking
    - DNS poisoning
    - Fake SSL certificates
- Data center layer
    - Web server
        - Platform (i.e., operating system, server software, or application modules) vulnerabilities
        - Server misconfiguration
        - Cross-site scripting (XSS)
        - Cross-site request forgery (CSRF)
        - Weak input validation
        - Brute-force attacks
    - Database
        - SQL injection

- OS command execution
- Privilege escalation

LACK OF ADMINISTRATIVE ACCESS

Neither individuals nor enterprises typically have administrative (or "root") access to their devices or the devices connecting to corporate assets. This makes it more difficult to monitor the security of a device, develop effective tools to repel attacks, and conduct thorough security testing of mobile apps.

Operating system developers, original equipment manufacturers (OEMs), and wireless carriers, however, can gain system or root access via their pre-installed apps. Unfortunately, attackers can exploit that administrative access via vulnerabilities in many pre-installed mobile apps. Developers, OEMs, and carriers are slow to patch such vulnerabilities (for myriad reasons), leaving individuals and enterprises exposed and powerless to do anything about it.

A lack of visibility into the lower-level operations of these devices and apps results in a lack of awareness of how an enormous amount of data is handled, and it's often handled insecurely. Worse yet, the individuals most affected by this lack lack of security, and the enterprises they interact with via mobile as employees or customers, are left in the dark. Such a wide variety of operating systems, devices and apps means mobile incident responders must understand a lot of different data from a lot of different sources in order to find and assemble the pieces of a mobile incident.

APPLICATION SANDBOX IMPLEMENTATIONS

An application sandbox creates a boundary of sorts around an app and limits the files and resources any particular app can access. Sandboxing provides some security benefit because it restricts the privileges of an app, which protects users from unintentional bugs and makes it more difficult for malware to compromise a device.

Unfortunately, installable security apps that might provide additional security and monitoring features are also limited by the sandbox and cannot truly protect the device due to restrictions. These restrictions hamper the effectiveness of traditional computer-based approaches to endpoint security such as the following:

- Anti-virus or anti-malware software
- Data loss prevention (DLP) software
- Host-based firewalls

APP STORE MODEL

Years ago, applications were installed using CD-ROMs and floppy disks. Today we take for granted the ease with which apps can be installed thanks to the app store model. The model has also significantly reduced the barrier-to-entry for app developers. In many cases, almost anyone can develop and distribute an app. This includes large companies with a mature development model that takes security and quality assurance (QA) into account (though plenty of them still don't). It also includes individuals (some amateur, some not) and smaller companies intent on developing the next "killer app." Unfortunately, it also includes malicious parties that can use an app store as an incredibly effective distribution channel.

It's never been easier to install an application on a computing device. In an aggregated analysis of data from users of the NowSecure Protect mobile security assessment app, as of November 2015 the average device has 300 apps installed on it. Users can't uninstall many of those apps, though they can disable some of them. The problem is many of those apps include security and privacy flaws. In an analysis of more than 400,000 apps available on the Google Play app store, NowSecure found that 49 percent of them include at least one high-risk security flaw.

MOBILE IS DIFFERENT

Each of the items explained in this section add up to significant differences between mobile and traditional IT security and incident response. The rest of this book will help prepare you and explain the tools you'll need to address these challenges in responding to a mobile incident.

Common Mobile Incidents

Here listed are a number of common scenarios you may encounter when responding to a mobile incident:

- Mobile malware discovered
- Device acting suspiciously
- Device lost or stolen
- Insider attack via mobile device
- Support for an internal investigation (e.g., e-discovery, legal hold)
- Data breach via mobile device

MOBILE MALWARE DISCOVERED

While mobile malware is overhyped by the security industry, it certainly exists. It is more prevalent in the Android ecosystem due to the ability to more easily distribute and install apps outside the Google Play store.

The two most common mobile malware incidents encountered are:

1. Malware installed on an individual's device
2. Malware impersonating your brand

The first situation generally impacts a single or small group of devices that are part of an organization (e.g., corporate-owned devices, employee-owned devices or BYOD, consultants).

In the second scenario, an unauthorized actor generally publishes a mobile app that targets your customers by impersonating your brand. Since this impacts your customers, the response steps vary significantly from an individual incident.

We discuss both of these scenarios in our mobile case studies later in the book and develop incident response playbooks for them as well.

DEVICE ACTING SUSPICIOUSLY

In many instances, an employee or contractor will inform the IT, security or incident response team that their device is acting suspiciously. This is a challenging scenarios due to:

- Limited visibility into the device
- A lack of historical data
- Any number of possible explanations or causes (and only some qualify as an incident)
- The sensitive nature of accessing an individual's mobile device
- Urgency due to potential impact and utility of a mobile device
- The incident reporter's accuracy

DEVICE LOST OR STOLEN

A lost or stolen mobile device is a scenario that is probably the most well understood. Both Android and iOS have built in capabilities for finding a lost or stolen device, locking it and performing a remote wipe.

However, with the information provided in this book, there's opportunity to better understand the potential impact of this type of event provided the security team has access to device properties such as operating system version and a list of installed apps. Armed with this information, a much better understanding of the data at risk can inform the response and resolution of the incident.

INSIDER ATTACK VIA MOBILE DEVICE

Detecting insider attacks is incredibly difficult, even more so on a mobile device because the device telemetry data available is limited and security tools are still evolving. The most likely event would involve an individual already under suspicion, and an incident responder being asked to perform an investigation on the mobile device.

SUPPORT FOR AN INTERNAL INVESTIGATION

Internal investigations, especially at larger organizations, are fairly common. Examples include:

- Data theft by a departing employee
- Employee violation of company policies
- Litigation freeze or e-discovery request

DATA BREACH VIA MOBILE DEVICE

As mobile devices play a greater role in the daily operations of large organizations, there is an increased risk of sensitive data (e.g. customer data, PII) being leaked out or breached in an attack. While many incident response teams have well practiced responses for incidents like this involving servers, very few have addressed the growing risk in mobile devices.

Tools for Mobile Incident Response

This chapter will provide an overview of the tools necessary to provide mobile incident response at your organization. We will focus on free and/or open source software but will mention commercial alternatives.

The sections include:

Categories of mobile incident response tools

It's helpful to characterize mobile incident response (IR) tools into four broad categories:

1. Continual Analysis
2. Device Acquisition
3. Analysis

In this section, we will describe these categories of tools and provide more detailed examples. In the subsequent sections, we will walk through the process of setting up an environment to use a variety of mobile IR tools, as well as provide a descriptive list of those that are open source, free, and commercial tools.

CONTINUAL ANALYSIS

The ideal scenario when responding to a mobile incident is to have had tools and procedures already in place to proactively log data that may be useful during an investigation. The purpose of this category of tools is to provide a baseline for device properties and behavior in four main areas that make up the mobile attack surface:

1. System (e.g., operating system, jailbreak status)
2. Configuration (e.g., passcode, encryption)
3. Apps (e.g., installed, updated, removed)
4. Network (e.g., security, DNS poisoning, SSL certificates)

DEVICE OR DATA ACQUISITION

This category describes the techniques used (and relevant tools) to properly handle the device, prevent any changes to the data, and properly acquire the data from the device.

Device Triage: Operating system, installed apps, running processes

Data acquisition tools are used to forensically recover the disk or memory contents from a device and store it as a copy in an external file. This file, or "image" of the data, can then be used during the analysis phase. The purpose of creating a forensic image of a device is to have a duplicate copy to run analysis tools against so that the original evidence remains pristine. The following are a list of various types of device or data acquisition tools.

Logical

A logical technique extracts allocated data and is typically achieved by accessing the file system. Allocated data simply means that the data is not deleted and is accessible on the file system.

Backup

Techniques are available on both iOS and Android devices to perform a backup of the operating system. On iOS, a backup is performed either via iTunes or iCloud, and stored on the user's computer or cloud account, respectively. On Android, backups can be performed through a 3rd party application, or using the android debug bridge (adb) command. Backups essentially include the same type of data that is recovered through a logical acquisition. Backups may be useful as an examiner if you don't have access to the physical device, or if you are looking to recover historical data that might be stored in an old backup but no longer exists on the device.

Physical

Physical techniques target the physical storage medium directly and these do not rely on the file system itself to access the data. There are advantages to this approach, the most significant being that physical techniques may provide access to deleted data. The file systems often only mark data as deleted or obsolete and do not actually erase the storage medium unless needed. As the physical forensic techniques provide direct access to the storage medium, it is possible to recover not only the allocated data but also the unallocated (deleted or obsolete) data. With encryption on the latest devices/operating systems, physical acquisitions are more difficult to perform than they used to be.

Proxying network traffic

Collecting a sample of network traffic from an access point that a mobile device is connected to will allow for additional network analysis to analyze protocols used by the various apps and services running on the device and determine if there are any unknown or insecure protocols in use.

Image verification

In digital forensics, examinations are performed on the original media only if absolutely necessary. In most cases, a forensic copy is made, and the examiner will analyze that image to avoid modifying the original media. In order to show that the working copy contains the same data as the original, it has become a best practice to create a unique signature for both the original and the copy by using a hash algorithm. If the values match, this technique shows that the forensic image is in fact a copy of the original. Most commercial tools will report either an MD5 or SHA1 hash value, two different algorithms, both of which are used for the same purpose. Linux commands can also be used to determine either of these values on an image or file using commands such as md5sum or sha256sum. Note that image verification will only be successful during the physical acquisition process.

ANALYSIS

Once the data has been acquired from the device using the techniques above, a number of analysis tools can be used to locate the target information.

Forensic analysis

Timeline analysis

Several tools are available that can be run against a disk image that will then list out each and every file within the file system, both allocated and unallocated. From this list, the tool creates a timeline of events that have occurred on the device. This process is typically run against a hard drive, but can also be used on a physical image file retrieved from a mobile device. The resulting timeline will show the file name; whether it was created, modified, or accessed; the date and time this event occurred; and other pieces of information that might be significant to an investigation. There are thousands, and sometimes hundreds of thousands, of files on these devices, so having the ability to sort by time is an important step in this process. Timelines are typically the most useful when an investigator has a specific time frame to narrow down the choices.

Searching

Once a physical image of a device is acquired, there are various tools that can be used to further analyze that image and search for specific keywords or other data. One of these methods involves the use of the Linux “strings” command. When run against a file (or even a full disk image), strings will extract printable characters that are at least four characters long. This command can be used against individual database files to potentially view and recover deleted data. Another method that allows the examiner to search for data is the use of a hex editor. By viewing a disk image within a hex editor, the examiner has the ability to jump to a specific area within the image. For example, if a particular e-mail address is of unique interest in a case, this address can be searched across the entire dmg. The examiner can then analyze the surrounding content.

File Carving

This process scans an entire disk for specified file signatures in order to recover files or file fragments from the disk. Using this method, both deleted and undeleted files can be recovered since the process focuses on the content of the files rather than its metadata. This is a popular technique used in forensic examinations because it's usually successful in recovering deleted photos, emails, text messages, and other important data. File carving techniques are built into some forensic analysis tools; however, there are open-source Linux tools available that will perform this action as well. These tools can be run via the command line against a physical image of a mobile device in order to recover valuable files.

Comparative

In any incident, it is ideal to use either current or historical data, and compare it to things like known malware, command-and-control (C2) servers, or app intelligence to identify security vulnerabilities. For example, in the event of an incident involving an imposter app existing on an app store, comparing the file signature to intelligence about known malware can help determine whether this is in fact a malicious or legitimate application. This comparative process can help an organization perform mobile app reputation monitoring and help identify any unauthorized use of their brand.

Malware analysis

A common incident involving mobile is that the device is "acting strange." In these cases, the user can't always identify what the problem is, just that something is happening on the device (e.g., increased battery drain or certificate errors). In this event, the device should be forensically imaged, and analyzed and, if possible, network analysis should be performed. If malware is suspected, the analyst would need to review the resulting artifacts to locate the potential malware. Once identified, the app should first be compared to intelligence about known malware, and then the app should undergo static and dynamic code analysis to determine whether the app is truly malicious.

Network analysis

Analysis of captured network traffic is important because it can reveal unique findings such as personal or corporate data being transmitted insecurely. It is important to understand what data is being transmitted, where it is going to in terms of the organization that owns the server, and what countries that data is being sent to. As an incident responder, if you have access to baseline or historical device data, this is ideal. You may then be able to identify whether the mobile device was connected to any insecure Wi-Fi access points, observe SSL re-signing, and more.

How to set up a mobile incident response workstation

Preparation is a key component to responding to any type of incident. Part of that step involves setting up an environment which contains all of the tools a mobile incident responder might need to properly respond to a situation. This chapter will provide you with the necessary tools to provide mobile incident response at your organization. We will focus on free and/or open source software. In addition, later chapters will provide labs you can perform on your workstation with detailed how-to walk-throughs.

In order to install and run the tools needed to respond to a mobile incident, it is easiest to simply setup a Linux virtual machine which already has everything pre-installed and ready for use. Santoku is a Linux distribution that contains a collection of tools related to mobile security, malware, and forensics and will be used in the various exercises and labs throughout this book.

On the mobile security side, app decompilation and disassembly tools are provided, along with scripts to automate decrypting binaries, deploying apps, and enumerating app details in order to detect common issues in mobile apps. Mobile device emulators are also provided along with utilities to simulate network services for dynamic analysis, which can be useful for both analysis of mobile malware and standard mobile apps. Finally, there are forensic tools available to help acquire and analyze data that resides on a mobile device. These include firmware flashing, forensic imaging, and other scripts that are specifically designed for mobile forensics. Later in this chapter, we provide a detailed list of free, open-source, and commercial tools that can be used.

WHAT YOU WILL NEED

- Santoku Linux .ISO file (Note: Santoku Linux versions 0.4 and later are 64-bit and require 64-bit hardware to run)
- VirtualBox or VMWare Player
- A host machine with a minimum dual-core processor, 2GB RAM, and 40 GB or more (recommended) free hard drive space

DOWNLOAD SANTOKU

Santoku is distributed as a .iso file. If you would like to install Santoku as the primary operating system on your host machine, you will need to create a bootable DVD or USB using the .iso file.

The recommended method of running Santoku is by installing it as a virtual machine (VM) inside either VirtualBox or VMWare player, which are applications that allow you to create and run VMs inside your native operating system. This tutorial highlights how to install VirtualBox, however, either should work for your needs. VirtualBox supports all major operating systems, including Microsoft Windows, Apple OS X, and Linux.

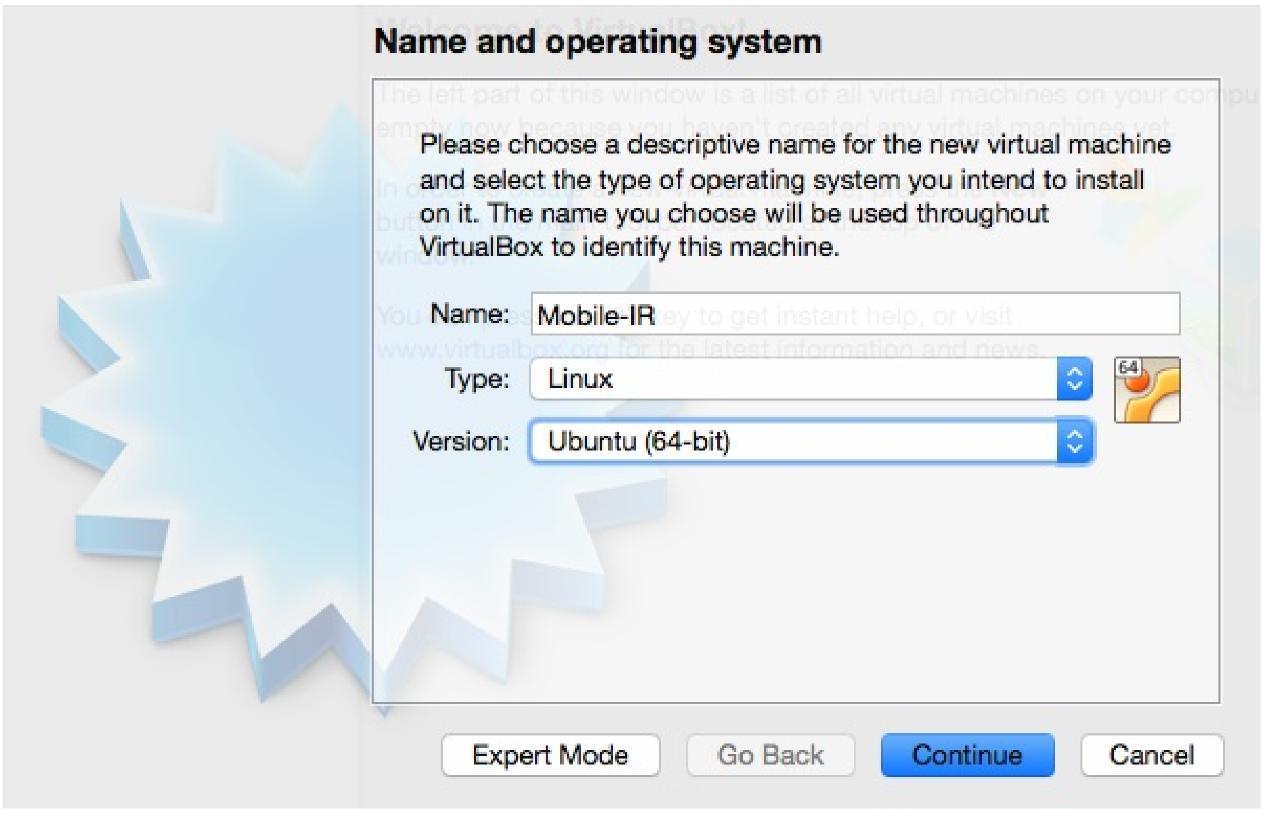
To acquire Santoku, you must first download the .iso file at https://www.santoku-linux.com/download.

SET UP YOUR VIRTUAL MACHINE

To run Santoku, you must install virtual machine software. For this section we will use the most recent version of VirtualBox, version 5.0.14. You can download it at: http://www.virtualbox.org/wiki/Downloads.

After downloading, install the virtual machine software on your forensic workstation, then take the following steps to start the virtual machine (VM):

- Locate your VirtualBox installation. Select “New” to create a new VM.
- Going through the wizard, create a name for your VM and select the Linux/Debian Operating System and Version. Note: For Santoku Linux 0.4 and newer, select “Ubuntu 64-bit”.



Name and operating system

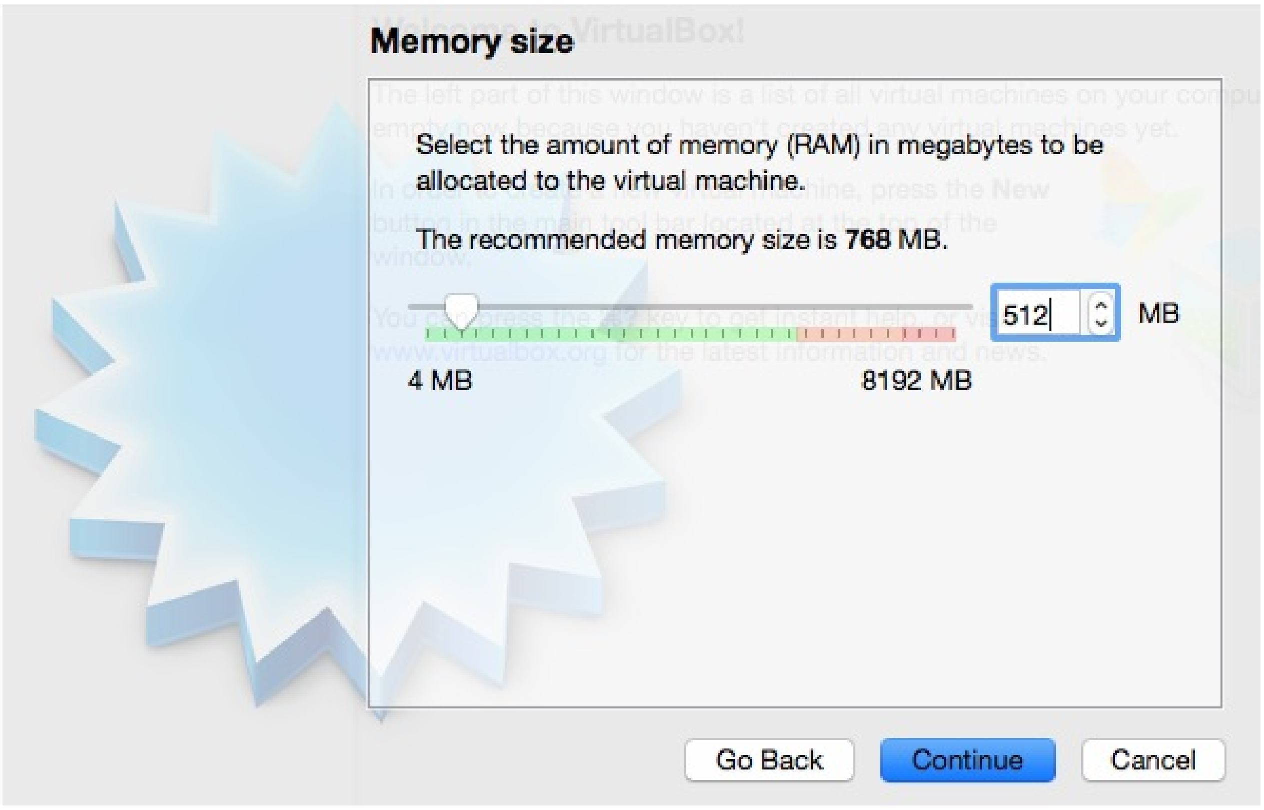
Please choose a descriptive name for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.

Name: Mobile-IR

Type: Linux

Version: Ubuntu (64-bit)

- Select an appropriate amount of memory for the VM. 512MB is standard, however, increasing the memory size will typically make your VM run faster (but make your host machine run slower). If you plan to use the Android Virtual Device Manager (AVD) and Android device emulator frequently, we recommend selecting at least 4GB of memory.



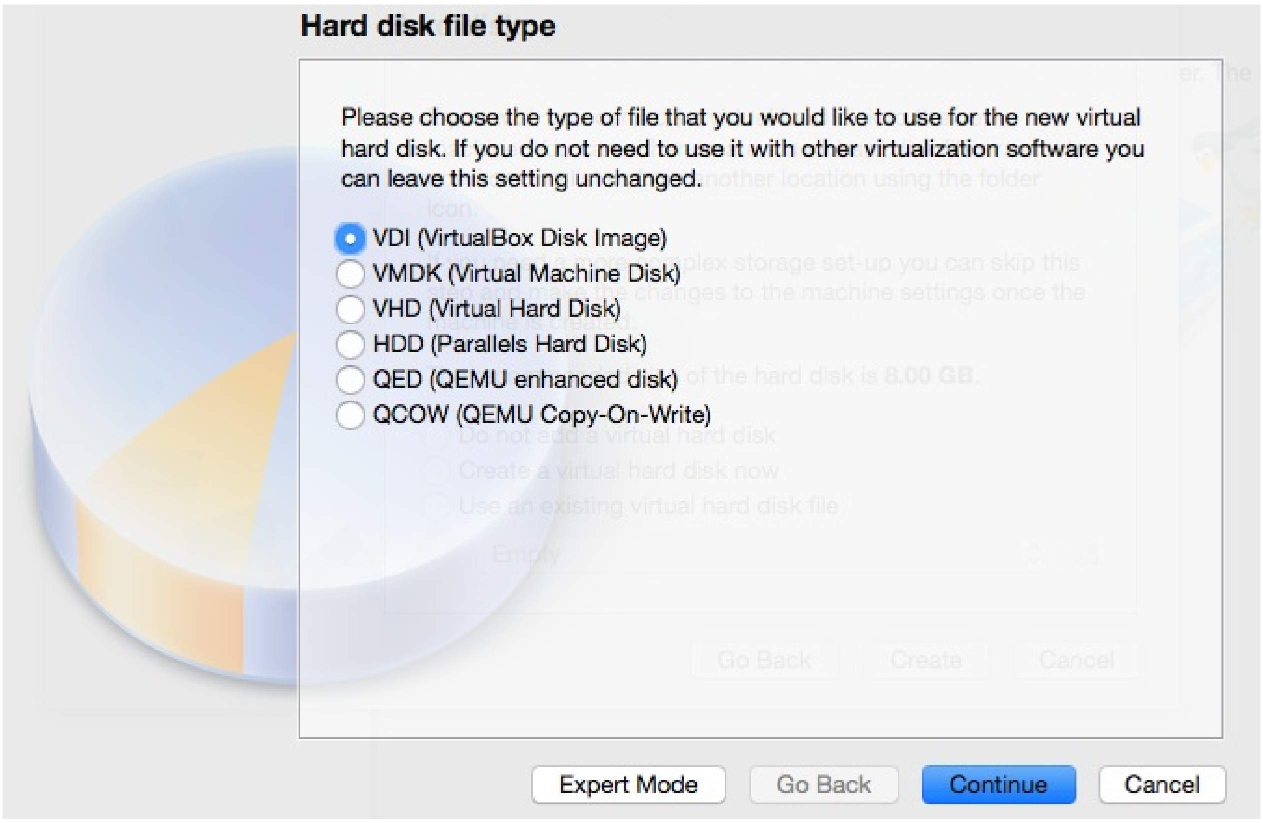
Memory size

Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.

The recommended memory size is **768** MB.

512 MB

- At the “Hard Disk” screen, select “Create a virtual hard disk now”. To create a new hard disk, select the “VDI (VirtualBox Disk Image)” option.

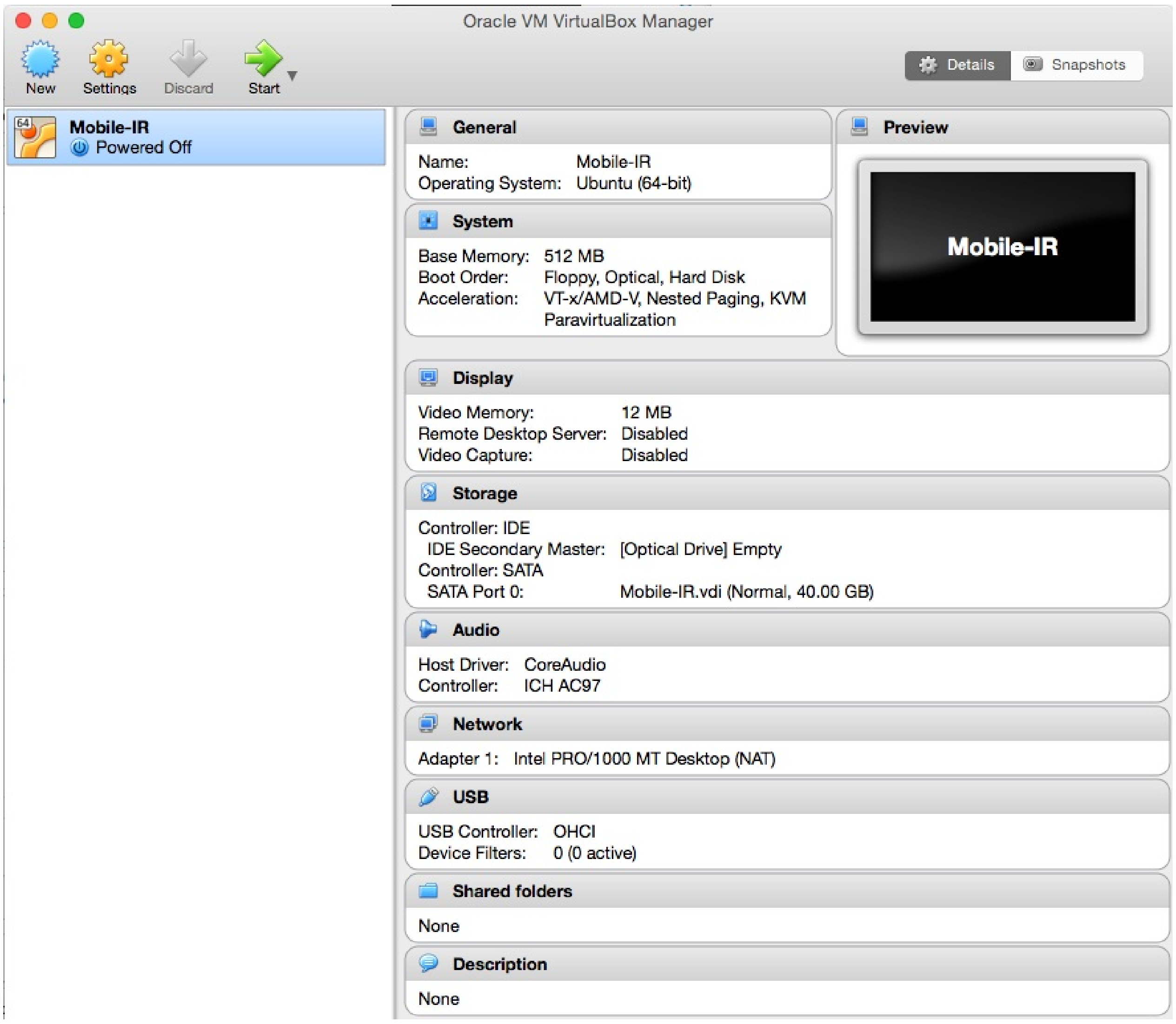


Hard disk file type

Please choose the type of file that you would like to use for the new virtual hard disk. If you do not need to use it with other virtualization software you can leave this setting unchanged.

- [✓ VDI (VirtualBox Disk Image)]
- VMDK (Virtual Machine Disk)
- VHD (Virtual Hard Disk)
- HDD (Parallels Hard Disk)
- QED (QEMU enhanced disk)
- QCOW (QEMU Copy-On-Write)

- On the next screen, select “Dynamically allocated” and click ‘Continue’.
- Choose the Virtual disk location where the virtual hard disk will be stored by clicking the folder icon on the right, then click the “Save” button. Adjust the ‘Size’ slider to allocate however much space you would like for your Santoku hard drive. Depending on your use, you may want to set this to a smaller or a larger value. The default in VirtualBox is 8 GB, we recommend increasing this to 40GB, as shown below. When finished, click “Create”. This will bring you back to the main VirtualBox menu.



General
Name: Mobile-IR
Operating System: Ubuntu (64-bit)

System
Base Memory: 512 MB
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization

Display
Video Memory: 12 MB
Remote Desktop Server: Disabled
Video Capture: Disabled

Storage
Controller: IDE
IDE Secondary Master: [Optical Drive] Empty
Controller: SATA
SATA Port 0: Mobile-IR.vdi (Normal, 40.00 GB)

Audio
Host Driver: CoreAudio
Controller: ICH AC97

Network
Adapter 1: Intel PRO/1000 MT Desktop (NAT)

USB
USB Controller: OHCI
Device Filters: 0 (0 active)

Shared folders
None

Description
None

- To get your Santoku-Linux to run on the VirtualBox, you need to have it attached it to your newly created Virtual Machine. This is similar to putting in a CD or DVD from which you will boot your machine when installing a new operating system. To do this, select the Santoku-Linux VM that you just created and click the “Settings” button at the top of the screen. Select the “Storage” option on the left of the Settings screen, and then click CD icon next to the “IDE Controller.”
- A warning will pop up asking you to choose a virtual DVD. Select “Choose disk” and navigate to your recently downloaded Santoku .iso file (in this case, it’s in the home user’s /Downloads/Santoku file). Click “Open”, then “OK”.

INSTALL SANTOKU

You can now click “Start” on the main VirtualBox screen to load the VM. Select to either boot from the live DVD or install. If you have created this in a Virtual Machine, choose “install – start the installer directly.”

Choose your language, time zone, and clock settings, then select “Erase disk and install Santoku” on the ‘Installation type’ screen. WARNING: If you choose this option and you are not installing Santoku in a Virtual Machine (i.e., you are not using VirtualBox and instead are installing it directly to your hard drive) this will ERASE YOUR HARD DRIVE. You have been warned. From there, add your username and password and click ‘Install.’

After the installation is complete, reboot when prompted and then login using the username and password you created during the install process.

INSTALL GUEST EDITIONS (VIRTUALBOX)

Once logged in, initiate the process to install VirtualBox Guest Additions by going to Devices -> Install Guest Additions. This is a one-time setup, and will allow for improved graphic performance, shared folders, and other features within the VM. You will see the Guest Additions icon appear on the Desktop. Right click it and select “Mount Volume”.

Next, open a Terminal window located under Applications -> Accessories -> Terminal (we have also created a shortcut to the Terminal window on the top status bar of the VM). Once open, navigate to the VBOXADDITIONS directory which was mounted in the previous step and execute the install script by running the following commands. *Do NOT type the “$.” This is intended to signify the beginning of a command prompt. Also, in this case it is VBOXADDITIONS_4.1.8_75467. The numbers following VBOXADDITIONS may vary.*

```
$ cd /media/VBOXADDITIONS_4.1.8_75467/
$ sudo sh VBoxLinuxAdditions.run
```

You may need to enter the administrator password, which you set up during install. Finally, the VM may need to be restarted for the changes to take effect.

Update your local package index with the latest changes made in repositories by typing the following: `sudo apt-get update`. After that completes, upgrade the packages by typing the following: `sudo apt-get upgrade`. These might ask if you want to continue (giving a Y/N option), type “Y” and hit enter.

GETTING STARTED WITH SANTOKU

Now that your VM is up and running, you'll want to connect a mobile device to it and start having some fun. To connect a mobile device to your VM, go to the VirtualBox menu, click Devices –> USB Devices –> . Make sure the checkmark is checked next to your device, and it will be passed through to your Santoku VM.

After reading about the tools in this chapter, feel free to head over to the Santoku HOWTO's section to read up on your favourite tools and learn about some new ones. In Chapters 7 and 8, we will provide additional exercises and Labs that will demonstrate how these tools can be used in practice.

List of mobile incident response tools

There are a number of open-source tools and distributions that can be used in investigating a mobile incident or during a forensic examination. The use of advanced Linux forensic analysis tools can help an examiner locate crucial evidence in a more efficient manner. Some of these tools are very powerful and provide the capability to quickly index, search, and extract certain types of files.

The following is a list of open source and other freely distributed tools that are available either within the Santoku Linux distribution or elsewhere, broken down by the categories discussed earlier in this chapter. While we don't cover tools that can be used to help establish an efficient IR process, there are a few open source options listed in Meirwah's Awesome Incident Response repo (under the "Incident Management" section).

The tools in the following section that have already been pre-installed within Santoku will be denoted by an "S", while others mentioned will need to be manually installed in the Santoku virtual machine (VM) that you've set up. Commercial tools will be briefly discussed at the end of this section.

CONTINUAL ANALYSIS TOOLS

Vulnerability Test Suite (VTS) for Android: Scans an Android device to detect known vulnerabilities. There are two types of vulnerability tests that can be performed:

- Detection only (does not attempt to exploit)
- Detects and attempts to exploit the vulnerability

iVerify-oss: Inspects an iOS device at boot-time to identify and collect information about any changes observed that may indicate the device has been modified by a jailbreak or other type of exploit.

X-Ray: X-Ray allows you to scan your Android device for security vulnerabilities that put your device at risk.

DEVICE AND DATA ACQUISITION TOOLS

This process involves not only acquiring the data from the device but also ensuring that the forensic image you've collected matches the file signature of the original (more details on this in the "Categories of Mobile IR Tools" section). Below is a list of tools that can be used to perform the device acquisition process, verify an image, and collect network traffic (when appropriate).

Device acquisition

- (S) AF Logical OSE: An open source tool that was released for use by non-law enforcement personnel or other individuals interested in Android forensics. It allows an examiner to extract logical data from an Android device through content providers. Here is a HOWTO guide guide for this tool.
- (S) iPhone Backup Analyzer 2: Allows user to browse content of iOS device made by an iTunes backup (or backup performed by another tool). More details on this tool can be found in it's repo; otherwise, you can find this tool in Santoku under the start menu -> Santoku -> Device Forensics -> iOS Backup Analyzer 2.
- (S) libimobiledevice: Cross-platform library that uses iOS specific protocols to recover data from the device's filesystem (no jailbreak required), perform a backup/restore, retrieve device information, and more. Here is a HOWTO guide for this tool. You will find the tool in Santoku under the start menu -> Santoku -> Device Forensics -> libimobiledevice.
- (S) dd: The “dd” command can be used on a device on which the examiner has root access (e.g., a jailbroken iPhone or iPad). This tool is generally used in forensics to acquire a full disk image of a hard drive, SD card, USB flash drive, or other device.
- FROST: A tool set that supports the forensic recovery of scrambled telephones.
- Lime: Allows for the acquisition of volatile memory from Android and other Linux-based devices.

Image verification

The following two checksum commands can be used to generate a digital fingerprint of a file, and in forensics, can be used to show that a physical image is an exact replicate of the data on a device at a given time. This verification proves that no files or content have been changed.

- (S) md5sum

- (S) sha256sum

Proxying network traffic

- Charles Proxy: Charles is a proxy tool that allows the user to view requests, responses and HTTP headers for HTTP and SSL/HTTPS traffic.
- (S) Burp Suite: A platform for security testing web applications, but it can also be used to assess the security of mobile app communications and web services.
- (S) ZAP: Open source penetration testing tool for finding vulnerabilities in web applications.

ANALYSIS TOOLS

Forensic analysis

- (S) Android Brute Force Encryption: This tool can help a forensic analyst crack the pin used to encrypt an Android device (this applies to Ice Cream Sandwich and Jelly Bean versions of the Android operating system). Here is a HOWTO guide for this tool.
- (S) scalpel: A file carving utility that is used to recover deleted files from a forensic image of a device (mobile or not). It recovers these files by searching a disk image for that file type's unique header and footer.
- (S) Sleuthkit: Sleuthkit is a collection of command line tools and a C library that allows you to analyze disk images and recover files from them.
- (S) strings: Running this command line tool against any file will provide printable characters that are at least 4 characters long from the file. Strings can really be useful when trying to locate information within a large file, such as a forensic image of a device (which can exceed 16GB depending on the size of the device). It allows for quick and efficient searching when used in combination with the "grep" command.
- (S) hexedit: No forensic investigation is complete without a hex editor. Hexedit is built into the Santoku VM and can be used to view or manipulate the binary data within a file.

Network analysis

The following tools can be used to analyze captured network traffic:

- (S) Wireshark: Wireshark is a network protocol analyzer and can be used for network troubleshooting and analysis. It can also be used to understand what type of data a mobile app is sending over the network unencrypted.
- (S) Ettercap is a suite of tools that are used to perform various types of man-in-the-middle attacks.
- (S) Nmap scans a server to discover hosts and services on a network and create a map using this information. Specific features include port scanning, version detection, operating system detection, and scripted interaction with the target.
- (S) Zenmap (as root) allows a user to easily run an "NMAP" scan using a graphical user interface (GUI) making it easy for beginners to use, and it includes advanced features for more experienced users.
- (S) Chaosreader can be run against captured network traffic and traces TCP and UDP sessions seeking application data.
- (S) dnschef is a DNS proxy tool for penetration testers and malware analysts. It can be used to create fake requests for a specified URL/domain and point it to a local machine as opposed to its intended server.
- (S) DSniff is a collection of tools for network and penetration testing.
- (S) mitmproxy allows for the interception and analysis of network traffic.
- (S) tcpdump is a command line packet analyzer.
- (S) wifite is an automated tool that is used to attack multiple WEP, WPA, and WPS encrypted networks.

Malware analysis

The following is a list of tools that can be used to reverse-engineer Android applications, decode resources and rebuild them after modification.

- (S) Androguard is a python tool used to reverse-engineer and perform malware analysis on Android mobile applications.
- (S) APKTool allows a user to reverse-engineer Android .apk files.
- (S) Smali/Baksmali is an assembler/disassembler for the dex format used by dalvik (Android's Java VM implementation).
- (S) AntiLVLcan be used to test an Android developer's protection methods against common types of attacks.
- (S) Bulb Security SPF is a smartphone penetration testing framework used to assess the security of a mobile device. The tool offers various types of attacks

such as remote, client-side, social engineering, and post exploitation attacks.

- (S) dex2jar converts .dex files to .class files (zipped up as .jar).
- (S) Drozer is a security testing framework for Android. It allows the user to search for security vulnerabilities in apps and devices by assuming the role of an app and interacting with the Dalvik VM, other apps' IPC endpoints, and the underlying operating system.
- (S) JD-GUI is a standalone graphical utility that displays Java sources from CLASS files.
- (S) Procyon is a suite of Java meta-programming tools.
- (S) radare2 is a multi-platform reversing framework that can be used to disassemble and assemble many different architectures, debug with local native and remote debuggers, and perform file system forensics and data carving, among other features.

COMMERCIAL TOOLS

While this section focuses on open-source software (OSS), commercial tools that can also assist in a mobile IR investigation are worth mentioning.

In addition to the list of OSS process/incident management tools that we linked to above, there are also commercial tools available such as Resilient's Incident Response Platform. This can be used to help automate the IR process by integrating directly with other prevention and detection systems that are already in place. It helps teams track incidents and offers dashboards and reporting features to provide status updates to various groups.

Continuous monitoring analysis

NowSecure's Protect mobile application, when installed in advance of a mobile incident, can help establish a device, operating system, and app baseline. When used within an organization, it provides an administrative dashboard that can allow access to aggregated security scores, network data, and vulnerability analytics. In the event of an incident, this baseline data can be used to pinpoint what went wrong and when.

zANTI: zANTI is a mobile penetration testing toolkit that lets security managers assess the risk level of a network.

Device acquisition / analysis

Most commercial forensics tools offer device acquisition capabilities and also offer built-in analysis tools. A sample of these tools are listed here:

- NowSecure Forensics (iOS / Android)
- Cellebrite
- XRY
- Lantern

Case Studies

Theory is great, but we often learn by example. This chapter will provide real-world case studies for common incident response scenarios you are likely to encounter. These scenarios include:

- Unauthorized app discovered
- Trojaned iOS app installs
- Compromise of company executive's phone
- Internal investigation
- Hacking Team analysis
- International travel

Unauthorized app discovered

Counterfeit apps developed by unauthorized parties and published to official or unofficial app stores (and elsewhere on the Internet) can damage a company’s brand name and put their customers at risk. As an incident responder, you may be called upon to investigate an incident involving an unauthorized app that might do any one or more of the following:

- Use a company’s brand name without permission
- Target the company’s customers
- Interact with the company’s backend services
- Is made available to the public via app stores or elsewhere on the Internet

AETNA++ CASE STUDY

In the fall of 2013, insurance provider Aetna’s Chief Information Security Officer (CISO) Jim Routh asked his IT staff for a list of all of the company’s mobile apps. Creating such a list can prove more difficult than it seems because mobile apps tend to proliferate. For example, an ambitious department within the company may hire a third party to develop and release a mobile app without communicating to the larger organization. Subsidiaries, affiliates and even unauthorized entities may publish an app that uses a company’s brand outright (and without explicit permission) or claims association with the company.

There are hundreds of third-party app stores in existence aside from the Apple App Store and Google Play. Jeff Lenton details the mobile app store ecosystem (as well as comments on this particular case of an unauthorized mobile app) in a 2015 presentation entitled "Understanding the Mobile Ecosystem". The ecosystem consists of official app stores, Internet company stores, manufacturer stores, carrier or operator stores, affiliate stores, and what he terms “feral apps” that are not available on app stores at all. Keeping track of all these stores, let alone the use of a company’s brand name on those stores, is a daunting task.

Routh chose to work with a third party (RiskIQ for whom Lenton worked) to help identify mobile apps that used his organization’s brand. The list included approximately 20 apps. Upon reviewing the list, one app in particular caught Routh's attention because he was not familiar with it. The app’s name appended two plus signs to the company’s name (i.e., “Aetna++”).

Initially, the Routh surmised that the app was developed for a marketing campaign. He began his investigation by identifying the developer of the app. The developer’s name was Nayem Junaid, and by reviewing the developer’s LinkedIn profile Routh determined Junaid was a ColdFusion developer based in Hydrabad, India. Routh set out to find Junaid in the company’s employee directory but couldn’t find any matches. Next, he explored whether the Junaid was a contractor and attempted to locate him with the help of a number of large contracting firms used by Aetna. The contracting firms did not have any record of the developer.

Routh then performed a quick Google search and found that 23 other large brands also had publicly available mobile apps that used their brand name along with the appended "++." At this point, Routh suspected the app was unauthorized.

Upon further analysis of the app, Routh discovered that the unauthorized app was a copy of Aetna’s official Android app with one difference -- the developer injected code into it. That code requested the GET_ACCOUNTS permission from Android, which allows for viewing what accounts are enabled on a mobile device (e.g., Google, Facebook, Twitter, etc.) through an account manager.

The first screen of the app matched the first screen of the official Android app, and a basic search function listed providers that the company supported. The rest of the app, however, did not function. The developer had republished the unauthorized app to third-party app stores and discussion forums. Users had downloaded the unofficial app more than 175,000 times.

In the unauthorized Aetna++ app, an ad library named com.edealya used the GET_ACCOUNTS permission. The company eDealya claims to help advertisers serve relevant ads to a consumer based on messages posted via a number of social media platforms. In the end, the unauthorized app harvested log-in credentials, and an article quotes Routh as saying the app was, “...a parasite capitalizing on Aetna’s brand equity to make money. This happens all the time.”

DETECTING AND REMEDIATING AN UNAUTHORIZED MOBILE APP

The same article explains that Routh recommends that in handling and preventing scenarios such as this one, companies need to acquire IT security intelligence from third-party sources, share information with their peers, and also work with state and federal entities.

As part of responding to the incident, Routh reached out to the other 23 affected companies to inform them of the unauthorized apps he found using their brand names. The third party Routh worked with, RiskIQ, also reached out to a number of advertisers to make them aware of their findings.

Obfuscating the code of mobile apps can also help make it more difficult to clone a branded app and insert malicious code into it.

Trojaned iOS app installs

This case study will examine the conditions leading up to, techniques used by and impacts of an attacker installing trojaned versions of iOS apps on a target's iPhone.

APPLE'S WALLED GARDEN

Apple tightly controls the iOS ecosystem and in particular, how apps are distributed via the App Store. While this frustrates some app developers, tinkerers, and security researchers; it has certainly reduced the impact of malicious apps by making it difficult to install them on target devices. An app needs to either be distributed via the App Store or signed with a special certificate issued by Apple to developers or businesses. The app then needs to be loaded onto the device via iTunes or a special link and requires user confirmation. Contrast this with Android where developers have many distribution channels at their diposal and device owners can configure the device to install untrusted apps.

If an attacker wants to install malicious apps on a target's iPhone, they either need to trick the user into accepting the developer- or business-signed application or gain physical access to the device to install the app directly.

CONDITIONS AND METHODS FOR COMPROMISE

In September 2014, Apple released the iPhone 6 and 6 Plus to the public. As with previous iPhone releases, the device was highly anticipated and sought after. In the U.S., that meant either waiting in long lines or waiting weeks on a backorder.

In countries other than the U.S., it is far more difficult to acquire a new iPhone model. Demand is extremely high, and people are willing to pay significant premiums to acquire a device. A new iPhone is also a status symbol. Often times successful business executives or government officials will receive a new model of a device. They may purchase the device, often using their connections, or receive it as a gift.

The iPhone 6 and 6 Plus frenzy presented a perfect opportunity to target an executive or government official.

In this case, the attacker knew of the target individual's desire for a new iPhone 6 and took advantage of the situation. The attacker purchased an iPhone 6 and, with the device in their physical possession, set the device up specifically for this individual (who also happened to be the unwitting target of the attack). The target individual's enthusiasm easily pushed aside any suspicions. The individual likely viewed this pre-configuration of the device as a courtesy so they could begin using the device as soon as they received it.

Since attackers had physical access to the device, they were able to install trojaned versions of popular apps on the device including:

- Chrome
- Facebook
- Skype
- WhatsApp
- Telegram

Trojaned applications were installed alongside with non-trojaned, but pirated, apps. Those apps included Pages and Numbers productivity apps, which also helped create the illusion that the device was "business-ready" to help in alleviating suspicion.

The apps targeted and collected sensitive data and communications from the device and sent the data back to the attacker via their command and control (C2) servers.

TROJANED IOS APPS

Creating a trojaned version of a popular app is a very effective technique for compromising a device. On the device, there would be very little indication that the device was trojaned. There would be negligible impact on battery and the functionality of the app would be consistent with the valid version.

To create a trojaned iOS app, an attacker does not need deep technical expertise. Instead, a developer with a few additional security tools could perform the following:

- Download the public app
- Decrypt the app
- Inject a dynamic library into the app bundle
- Modify the app's main executable to load the injected dynamic library at app launch
- Hook interesting parts of the app

- Configure the app to send intercepted data back to the attacker via a C2 server
- Re-sign the app with a developer or enterprise certificate

At that point, an attacker with physical access to the iPhone could install the app via either an iTunes sync or by clicking on a link to the binary (via the `itms://` uri) and then trusting the app when prompted by iOS.

DETECTING AND REMEDIATING A TROJANED IOS APP

As with any incident, you must first identify the incident. While it's unlikely that an individual (unless both technical and security savvy) will be able to identify a trojaned app (unless they're technology and security savvy), a number of signs might alert a security analyst to a trojaned app.

Since the app is modified from the App Store version, an analyst equipped with the proper tools and information will be able to detect and determine an illegitimate app. For example, if you review the outbound network connections from the device, you would notice traffic going to servers not associated with normal app traffic. There are various ways to inspect network traffic including:

- Continual analysis tools
- Netstat app
- Netstat command via a terminal app
- Proxying network traffic
- Inspecting network traffic on a network device (i.e., at the switch level)

Note that analysing *contents* of the data being sent (as opposed to the *destination* to which it's sent) can be problematic because it's possible, if not likely, that TLS is used to protect the confidentiality and integrity of the data. In such cases an analyst must either take additional steps to decrypt traffic (i.e., by adding an extra trusted root certificate to the device under scrutiny) or, if such device modifications are not desirable, choose alternative analysis techniques.

In addition, an analyst might inspect the app bundle and see that it is a developer- or enterprise-signed app, which means it's modified from the Apple App Store version. This can be seen by inspecting the app binary directly and examining the certificate used to sign the app and its executable content.

Finally, apps not installed via the Apple App Store will not auto-update, so the user may become suspicious when other people receive app updates and they do not. Or they may search the Apple App Store, install the original app, and wonder why two icons appear for the same app and requiring them to log in separately for each of the apps. Finally, with some continual analysis apps, the security analysts should be able to detect that a non-standard version on a popular app is installed on a device.

Once the trojan app has been identified, it is critical to quickly contain the threat. The most effective approach would be to isolate the device by enabling the device's airplane mode or placing the device into a Faraday cage (box/bag). You would then perform a device backup and secure a copy of the app binary for inspection. We will explore this in more detail throughout the Framework for Mobile Incident Response chapter.

Compromise of company executive's phone

Another mobile scenario an incident response analyst may encounter is the potential compromise of an executive's phone. Mobile devices have become very personal, and individuals are acutely aware of minor changes in their device's behavior. When an executive's mobile device is acting strangely, many times its owner will turn the device over to the IT or security team for investigation.

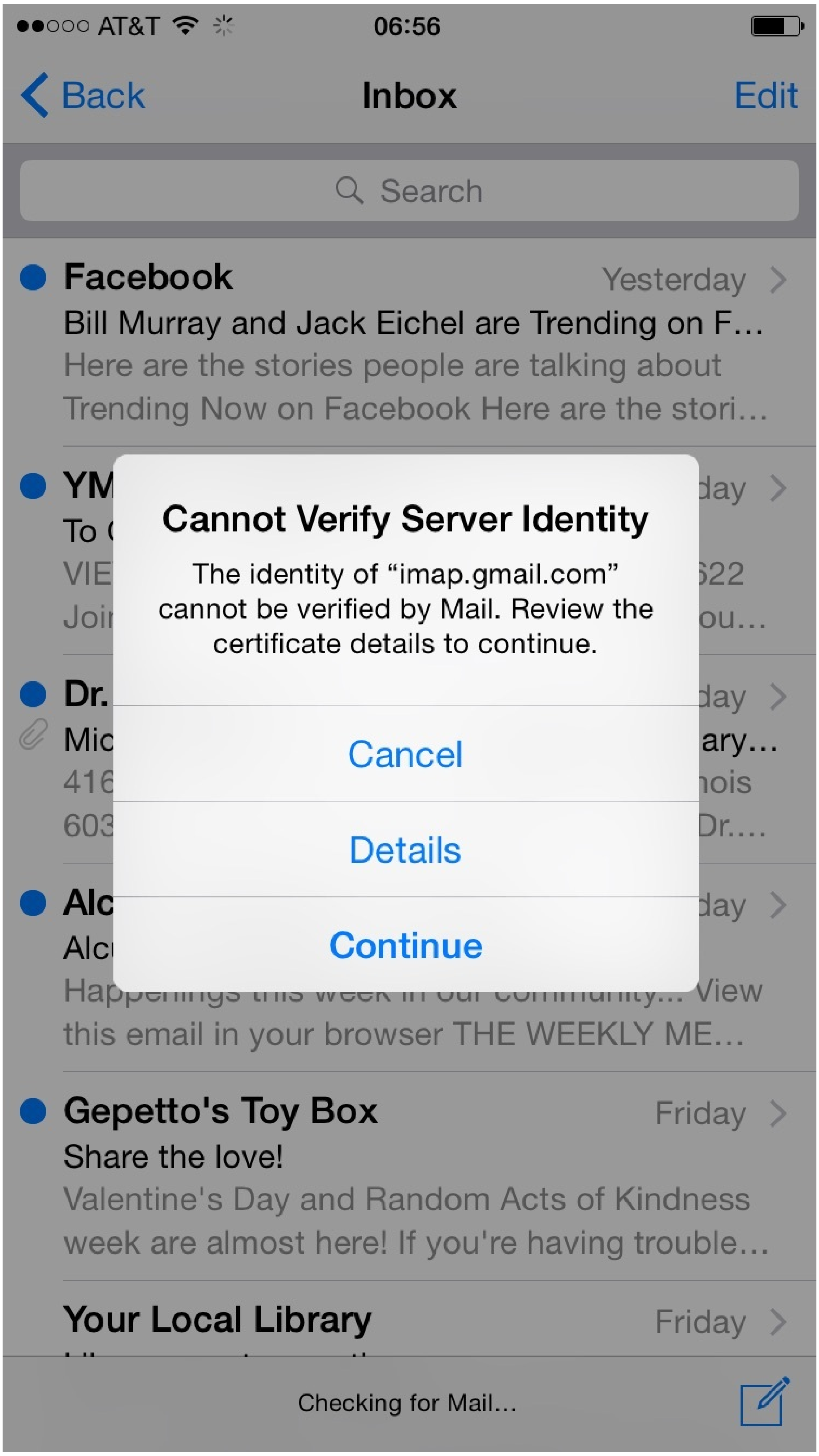
These are incredibly challenging cases because so many different factors can cause a device to act in a different manner. Very few organizations have the continual analysis tools in place to help understand how a device's state has changed over time.

Finally, another challenge is that company executives tend to be incredibly busy, glued to their mobile phones, and, often times, impatient. This is clearly an area where incident response teams want to ensure they have the proper tools and training in place to quickly identify and remediate incidents.

INVALID EMAIL CERTIFICATE CASE STUDY

The CEO and the co-founder of a technology company experienced problems syncing their email over a weekend in early 2016. When the iOS Mail app was launched, it would intermittently throw a certificate error. In this particular case, the CEO was technical and actually captured screenshots, which are very helpful in determining the specific errors that occurred:

> Cannot Identify Server Identity
>
> The identity of "imap.gmail.com" cannot be verified by Mail. Review the certificate details to continue.



●●○○○ AT&T 06:56

Back Inbox Edit

Search

Facebook Yesterday
Bill Murray and Jack Eichel are Trending on F…
Here are the stories people are talking about Trending Now on Facebook Here are the stori…

Cannot Verify Server Identity

The identity of "imap.gmail.com" cannot be verified by Mail. Review the certificate details to continue.

Cancel

Details

Continue

this email in your browser THE WEEKLY ME…

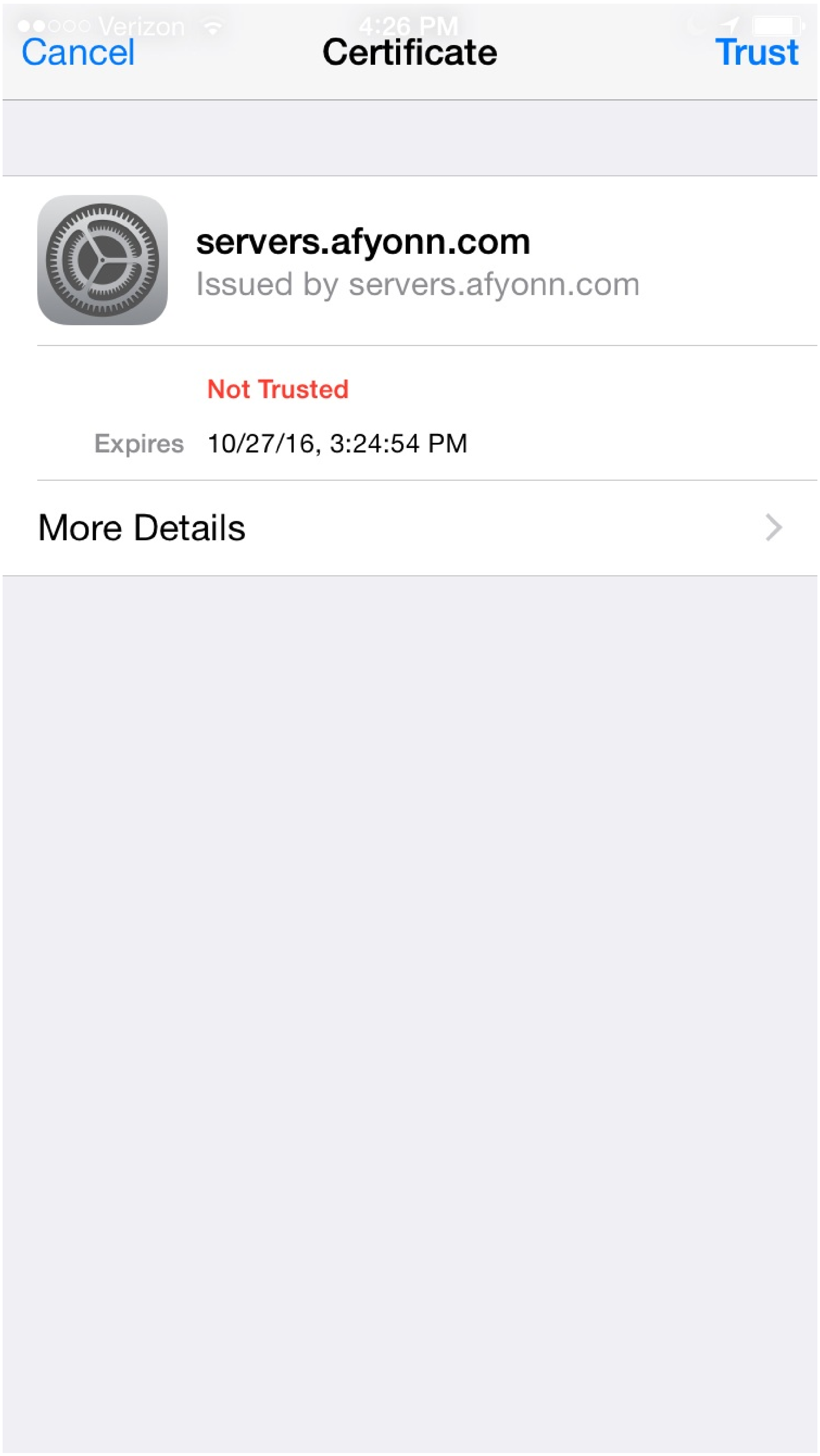
Gepetto's Toy Box Friday
Share the love!
Valentine's Day and Random Acts of Kindness week are almost here! If you're having trouble…

Your Local Library Friday

Checking for Mail…

The Gmail app for iOS would not sync email either, but it dropped the connection silently.

The certificate was self signed and had a common name of "servers.afyonn.com," and it was set to expire in Oct 2016:



servers.afyonn.com
Issued by servers.afyonn.com

Not Trusted

Expires 10/27/16, 3:24:54 PM

ADDITIONAL INVESTIGATION

Since this anomaly impacted two specific devices, further analysis was warranted. The CEO reported an incident to the company's security team, which performed additional data collection and analysis.

It was determined that a non-standard DNS server was active at the executive's home. This DNS was located outside of the country and was not part of the infrastructure of their Internet service provider (ISP).

The home router was inspected and several security issues were discovered including a default, publicly known password and some remote management interfaces being enabled.

The team was able to capture the TLS negotiation at a later time so that the full certificate chain was available for inspection.

INCONCLUSIVE FINDINGS

Clearly, an incident of concern occurred involving the CEO's devices. Like many incidents, it was difficult to determine the root cause. The steps proposed by the incident response team included:

- Take the home router offline
- Analyze the router for signs of compromise
- Attempt to capture the incident live and collect more data
- Forensically acquire and analyze the devices
- Wipe the devices and re-provision them
- Issue a new home router and secure it according to best practices

The investigation benefited from the individuals involved being security aware. However, this scenario is not common. Even with the technical capabilities of the impacted individuals, it still took several days to identify, analyze and remediate the incident.

Internal investigation

A number of incidents may occur within an organization that require a team to perform an internal investigation. The IR team should follow a similar process for these internal incidents as they would any other. In this section, we discuss ways in which a mobile device may be involved in an internal corporate investigation.

DEPARTING EMPLOYEE DATA THEFT

When an employee leaves an organization, it is important to have a process in place to properly disconnect the employee's access to corporate resources and ensure that no corporate data resides on that individual's mobile device. One of the challenges in mobile involves the bring your own device (BYOD) trend and employee-owned versus corporate-owned devices. In a BYOD scenario, the IR team may not have direct access to the device or artifacts needed to properly investigate an incident. With a corporate owned device, the specific incident details may vary, but the general steps would involve:

- Forensic acquisition of data on the device
- Collection of network traffic transmitted from device
- Analysis of artifacts to determine whether data theft had occurred

An example scenario may involve using advanced SQLite recovery techniques to retrieve deleted SMS records, call logs, or other forms of communication that may assist in the investigation. Documented communication with co-workers or individuals outside of the organization may prove useful. Performing a timeline analysis of the files that reside on the device can also be a helpful piece of evidence in any type of investigation. As discussed in Chapter 2, there are tools available that can provide a listing of all files that exist on the device as well as when they were created, modified, or accessed. By narrowing this list to the specific timeframe in question, an investigator can get a good understanding of the actions that took place on the device during that time. An example timeline analysis might show that a PDF file called "Company Confidential Data" was created at 10:15 a.m., and an email was sent at 10:16 a.m. Additional email analysis might show that "Company Confidential Data.PDF" was e-mailed to a user outside of the organization.

Hacking Team

One challenge incident response, and more broadly security, teams face is providing the business case for investing in the necessary people, tools, training and process. This section includes information about the compromise of the notorious company Hacking Team to achieve the following:

- Provide empirical evidence that nation-states and malicious actors target mobile devices
- Explain that there exists a market willing to pay for the ability to compromise mobile devices
- Demonstrate that vulnerabilities, technologies and companies exist that make mobile compromise possible.

Case studies such as this one will help you in making the business case for investing in mobile security and incident response.

HACKING TEAM BACKGROUND

The Milan, Italy-based company Hacking Team markets surveillance software, called Remote Control System (RCS), to government and law enforcement agencies. The system allows a user to launch exploits against targets (including desktop computers, laptops, and mobile devices), deliver payloads, and remotely control compromised systems and/or exfiltrate data from those systems. Among other features, RCS can monitor and record communications, decipher encrypted data, and remotely activate microphones and cameras. Many people have criticized Hacking Team for selling their software to repressive governments that violate basic human rights and use the software to spy on citizens, activists, and journalists. Predictably, Hacking Team vehemently denied these allegations, but as the raw data and analysis definitively proves, Hacking Team was very much engaged in selling exploits to anyone who had the means to purchase them.

COMPANY COMPROMISE

On July 5, 2015, an allegedly unauthorized individual took control of Hacking Team's Twitter account and posted a message stating, "Since we have nothing to hide, we're publishing all our e-mails, files, and source code" and including a link to more than 400 GB of internal Hacking Team data. The files included company emails, customer data, and source code among other things.

The data showed that Hacking Team also sold software to private companies and in one case for the purposes of surveilling Android and Blackberry mobile devices. In order to install the RCS software on a targeted device without the user's knowledge, a vulnerability on that device would need to be exploited. Emails reveal that Hacking Team recruited exploit developers and also purchased exploits that paved the way for installation of the RCS software.

In addition, an archive of the leaked emails provided by WikiLeaks includes an email stating that Hacking Team worked on developing an exploit for a vulnerability in a version of the SwiftKey keyboard shipped with several Samsung Galaxy models. The email states, "Is it really exploitable? Yes. Are WE working on it? Obviously," and includes a link to an Ars Technica article, "New exploit turns Samsung Galaxy phones into remote bugging devices," about the vulnerability discovered by NowSecure researcher Ryan Welton.

Source code allegedly recovered from the leaked data and posted on GitHub includes RCS agents for Android, Blackberry, iOS, and Windows Phone. These agent's capabilities included the ability to take pictures, copy events from calendars, and intercept communications such as, phone calls, SMS messages, and chat messages from apps like Skype and WhatsApp.

It was also revealed that Hacking Team possessed an Apple enterprise certificate. Reports claim that apps created by Hacking Team and signed with this certificate could be installed on even non-jailbroken iOS devices. This allowed Hacking Team to develop and distribute custom apps outside of the Apple App Store and without going through the typical Apple review process. Researchers found in one case that Hacking Team had created spyware hidden in the native Apple Newsstand app.

International travel investigation

If a company representative plans to travel abroad in potentially hostile areas, ideally they will use a dispensable tablet or phone specifically for the trip and leave the laptop or device they use for work at home. The traveler should use that designated device under the assumption that it and any data upon it is or will be compromised (i.e., they shouldn't input, send or receive any sensitive data on it).

This section will cover three aspects of how a device might be handled, used and connected to communications infrastructure as a result of international travel. If, for example, a company representative returns from abroad and reports a potential incident, you need to determine how the device has changed, what parties handled the device, how the device connected to mobile infrastructure and the Internet, and what data may have been exposed.

CUSTOMS

When a traveler enters or exits a country, customs authorities may inspect luggage including personal effects such as a mobile device or laptop. During customs procedures, authorities may intercept and examine a tablet or phone. Depending on the country, authorities may also perform a forensic acquisition on the device. A forensic acquisition consists of extracting data from the device and creating a copy of that data (called a forensic image). When authorities take possession of the device, they also have an opportunity to implant malicious code on the device for the purposes of surveillance.

CONNECTING TO MOBILE INFRASTRUCTURE AND NETWORKS ABROAD

Once an employee is overseas, their phone will connect to that country's cell phone network. Mobile carriers have considerable, low-level access to the devices that connect to their infrastructure. In China, the three main telecommunications providers - China Mobile, China Unicom and China Telecom - are state-owned enterprises.

Numbers from June 2015 show China Mobile with a 60.2 percent market share compared to China Unicom’s 24.5 percent and China Telecom’s 15.3 percent. If someone from your company connects to a cell tower in China, it’s likely the tower is state-owned.

A mobile carrier or malicious party can route a traveler’s device to specific networks upon the device’s connection to a base transceiver station (BTS), which is part of the technology a cell tower uses to facilitate mobile communication. The baseband processor, a chip found within a mobile device, handles the communications emanating from and received by the device. Security flaws within a device’s baseband processor can be exploited via a BTS as explained by Ralf-Philipp Weinmann in his paper "Baseband Attacks: Remote Exploitation of Memory Corruptions in Cellular Protocol Stacks." In addition, mobile security researcher Aaron Turner also stated, “Foreign carriers and malicious BTS operators can 'catch and release' your device while making updates or settings changes that will enable persistent monitoring,” in a presentation at RSA Conference 2013 entitled, "Mobile APT - How Rogue Base Stations Can Root Your Devices".

If you’re called to respond to an incident related to a company representative’s travel abroad, you’ll want to keep the possibility of a baseband attack in mind. If possible, you’ll also want to consider taking forensic images of the device prior to travel abroad and upon the device’s return to the home country.

CONNECTING TO WI-FI ABROAD

Business travelers connecting to Wi-Fi presents another vector by which malicious individuals may compromise a mobile device. One example of this is the Darkhotel campaign labeled an advanced persistent threat (APT) by Kaspersky Lab. In their report on the attack published in November 2014, Kaspersky Lab researchers stated that Darkhotel targeted corporate executives staying at hotels around the world via the hotel’s in-house Wi-Fi or business-center Internet access points.

The report states that the majority of infections occurred in Japan, Taiwan, and China. The malicious actors behind the attack seemed to have information about targeted executives’ names and the hotels at which they planned to stay. The attack used forged certificates to dupe executives into downloading what posed as one of several major software releases but was in fact a Trojan consisting of keyloggers and other malware. While the Darkhotel campaign does not target mobile specifically, it’s an example of the targeting of executives abroad via Wi-Fi.

Framework for Mobile Incident Response

Incident response is a process. And the great thing about a process is that a general framework can be adapted to new scenarios. This chapter will provide an overview of popular incident response processes as well as help develop and demonstrate a mobile specific plan with playbooks. The sections include:

- NIST Computer Security Incident Handling Guide
- SANS Incident Handling Steps
- Mobile Incident Response Plan
- Mobile Incident Response Playbooks

Attacking Mobile Devices

Mobile devices have a broad attack surface that incident response analysts much understand at a high level to preperly respond and defend against attacks. This chapter will explore important concepts impacting mobile security and a detailed review of how mobile devices are compromised:

- The SCAN Principle of Mobile Security
- Anatomy of a Mobile Attack
- Mobile Exploitation Process
- Real-world Mobile Attacks
- Who is Responsible

Mobile Persist and Exfil

Once an attacker has compromised a mobile device, there are a variety of way in which they can persist on the device and exfiltrate sensitive data. This chapter will explore the different techniques in the following sections:

- Mobile Persistence
- Mobile Exfiltration

Android Incident Response

This chapter will provide a step-by-step technical approach for responding to an Android incident. We will discuss Android security mechanisms, a tailored incident response process and a lab exercise you can complete. The sections include:

- Android Security Model
- Android Incident Response Process
- Android Data Collection
- Android Incident Response Analysis
- Android Incident Response Lab Exercise

iOS Incident Response

Processes for incident response and digital forensics have a lot in common and iOS is no exception to this. One important difference though is that for incident response process we assume owner of the allegedly compromised device to fully comply with investigator's request to provide access to the device, such as to disclose or disable passcode. This is a subtle but very important difference, especially on iOS, where passcode protection relies on proven cryptography and, in many cases, cannot be bypassed.

This chapter opens with a general introduction to iOS system and application security and then moves on to explain iOS-specific parts of mobile incident response.

- iOS Security Model
- iOS Incident Response Process
- iOS Data Collection
- iOS Incident Response Analysis
- iOS Incident Response Lab Exercise

Mobile Malware Analysis

Inevitably, you will encounter some form of mobile malware which must be analyzed to fully understand the impacts of the incident.

This chapter will cover:

- Malware Analysis Best Practices
- Android Malware Analysis
- iOS Malware Analysis

Remediate and Prevent Mobile Incidents

The ability to quickly remediate a mobile attack and then prevent future attacks is an essential goal of any incident response team. This chapter will cover the following topics:

- Mobile remediation techniques
- Mobile incident prevention

Glossary

SANTOKU LINUX

Santoku Linux is an Ubuntu-based Linux distribution focused on mobile forensics, malware analysis and app testing. It comes pre-installed many of the tools needed to support these capabilities. The distribution is open source and maintained by [NowSecure](https://www.nowsecure.com/)

2.2. Setup a Mobile Incident Response Workstation
2.3. List of Tools