Global Information Assurance Certification Paper

BYOD Security Implementation for Small Organizations

GIAC GCIH Gold Certification

Author: Raphael Simmons, dejacpp@gmail.com
Advisor: Sally Vandeven
Accepted: December 2017

Abstract

The exponential improvement of the mobile industry has caused a shift in the way organizations work across all industry sectors. Bring your own device (BYOD) is a current industry trend that allows employees to use their personal devices such as laptops, tablets, mobile phones and other devices, to connect to the internal network. The number of external devices that can now connect to a company that implements a BYOD policy has allowed for a proliferation of security risks. The National Institute of Standards and Technology lists these high-level threats and vulnerabilities of mobile devices: lack of physical security controls, use of untrusted mobile devices, use of untrusted networks, use of untrusted applications, interaction with other systems, use of untrusted content, and use of location services. A well implemented Mobile Device Management (MDM) tool combined with network access controls can be used to mitigate the risks associated with a BYOD policy.

1. Introduction

Bring your own device (BYOD) is the current industry trend that allows employees to use their private equipment such as laptops, tablets, mobile phones and other electronic devices, to connect to the internal network of the company. Allowing personnel to use their personally owned equipment goes against the traditional standard of only using company supplied electronic equipment for use on the company's internal network. With company-supplied computers and mobile phones, employees could not make changes to the configuration of the device and had to adhere to the company's 'Acceptable Use' policy and other practices which governed the use of these devices. The purpose of the policies and practices by an organization is to protect the confidentiality of the company's data, assure the integrity of that data, guarantee that the data will be available to the organization's personnel when needed, and to authenticate authorized user access to the organization's enterprise.

With company-owned equipment, keeping the software up to date, updating the operating system, and applying security patches and maintenance was the sole responsibility of the organization. Ensuring that these devices were configured to use robust encryption, complex password requirements, and the automatic wiping or deleting of a mobile phone that is lost or stolen is the job of the network administrator. Leaving the protection of personal devices up to the abilities, resources, or whims of the users, rather than the dedicated efforts of security professionals can cause a data breach if the user fails to secure their devices. With company-owned equipment, it made protecting the organization and its users the main priority. The rule: company -owned, company-maintained, business use only (Miller, 2012).

Laptops, the first mobile devices, were limited in their capabilities compared to a desktop. Applications like Microsoft Office and other software tools could be installed on the laptop computer while allowing an organization's employee to work offline while traveling. The first mobile phones were merely communication devices with pre-installed games for mild entertainment. Significant advancements in technology over the past 15 –

20 years has combined the capabilities of a laptop computer and a communication device into a single unit. The joined components weigh less than a pound with a screen size of a 5x7 index card (smartphone) or, as large as an 8x10 sheet of paper without the traditional phone capabilities of a mobile phone (tablet). Equipped with a touchscreen, camera, voice recorder, video chat, text messaging, they can store music, pictures, videos, and games and can be used to watch live entertainment. They can run productivity applications like Microsoft Office and Adobe-- all clear evidence that times have changed.

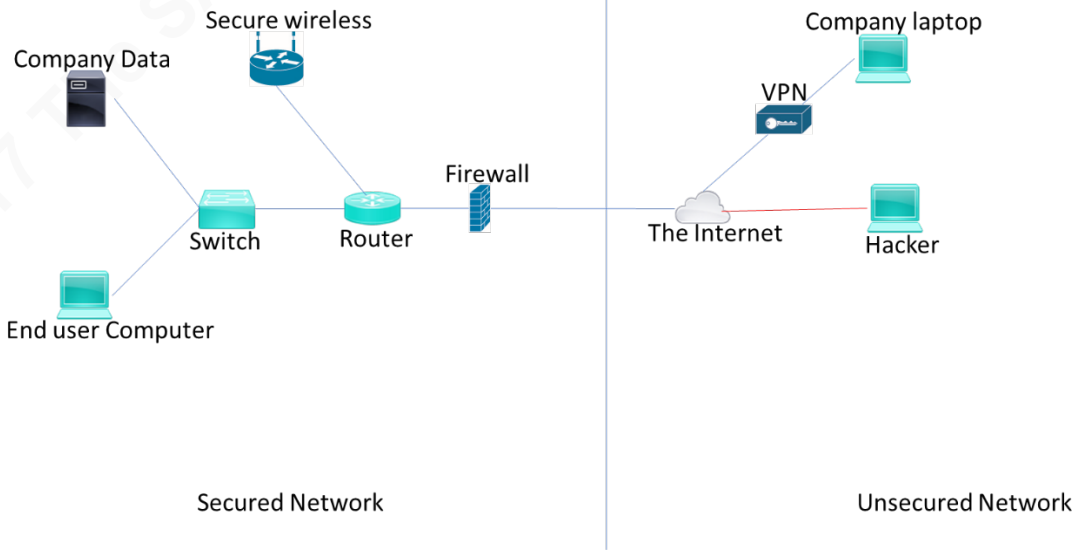
The exponential advancement of the mobile industry is causing a shift in the way organizations work across all industry sectors. BYOD utilization in business allows employees to work from anywhere they have a network connection. They can update company data, prepare documents, and participate in webinars away from the office. A 2014 survey, reported that industry leaders like Intel have approximately 70% of their 80,000 employees using their own devices for company work (Smith, 2017; Tse, 2016).

The security risks associated with mobile technology have introduced additional attack avenues that were not present in the typical wired network environment from twenty years ago. Listed below in Table 1 are some of the regular attacks on wired and wireless environments. Figure 1 shows an example of a simple wired network with an internal secured wireless device for internal use only. In this scenario, ‘internal' refers to business. All equipment on the secure side of the diagram is controlled by the organization. The firewall helps prevent hackers from gaining access to devices on the inside of the network. Not indicated in the diagram are the added security features an organization will implement at different areas of the network, such as at the router and switch, in addition to the devices themselves. On the unsecured side of Figure 1 is a company laptop configured with security features that the user cannot change which allows secure communication to the organization's network.

Raphael Simmons, dejacpp@gmail.com

Table - 1 Network Attacks			
Threats	Description	Wired	Wireless
Data Leakage	Unauthorized transmission of data	Yes	Yes
Sniffing	Tapping or eavesdropping	Yes	Yes
Spam	Unsolicited email messages	Yes	Yes
Spoofing	Spoofing user email	Yes	Yes
Phishing	Fake emails that appear to be legitimate	Yes	Yes
Pharming	Redirection traffic to a nefarious website	Yes	Yes
Vishing	Leaving voice mail purporting to be a legitimate company	Yes	Yes
Denial of Service(DoS)	Disrupting the availability of network resources	Yes	Yes
Distributed DoS	Many external systems involved in a DoS attack	Yes	Yes
Bluesnarfing	Stealing information via Bluetooth	No	Yes
SPIM	Unsolicited text messages	No	Yes
Jamming	Jamming a radio signal	No	Yes
Flooding	Text message flood	No	Yes
Exhausting	Running applications in the background to drain the battery	No	Yes
Blocking	Shutdown smartphone features	No	Yes

Figure 1. Typical Wired Network



Raphael Simmons, dejacpp@gmail.com

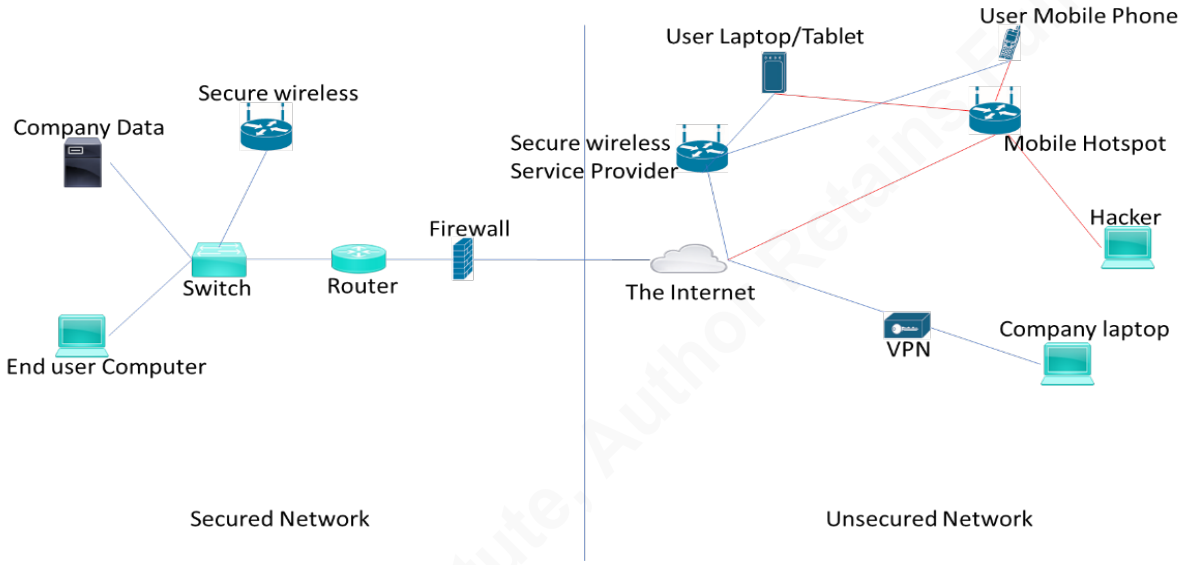
In the diagram above, an internal user initiates a request to a search engine. The call traverses the organization's switch and router. These devices are responsible for sending the user request to the search engine. The firewall allows the communication between the internal user and the search engine based on the configuration rules of the firewall. If an attacker initiates the communication to an internal device, the communication should be dropped since it did not initiate from an internal device.

In the next scenario, the user is allowed access the company's network using their equipment possibly on an unsecured access point. The BYOD is introduced into the network in Figure 2, below. The unsecured side has changed drastically. A hacker on the same wireless network may be able to access company information on an improperly configured device via the path shown in red. To add to the problem, since the user's individual devices are not controlled by the organization, there is no mechanism in place for the company to manage the security configuration of the equipment. On the secure side with company equipment, the principle of least-privilege should be applied which means users are given minimal needed access to accomplish their assigned duties.

Compare this to the user's personal equipment where the default configuration is set to privileged access. Equipment owners can configure the security features as they wish, download any software as they please, and are not obligated to tell the company if the device is lost or stolen. These devices provide open pathways for attacks on their equipment and can lead to intrusions on company equipment. In this situation, if the user's device is not properly secured, vulnerabilities could be introduced into the network and cause a serious problem to an organization's security posture. Preventing the exfiltration of company information is vital to the organization's ability to operate. A breach could diminish their reputation and threaten their ability to remain in business.

Raphael Simmons, dejacpp@gmail.com

Figure 2. Typical Wired Network with Mobile Devices



The first step in protecting the network is to identify what devices are connected to the organization's infrastructure whether it is through a wired or wireless connection (CIS, 2016). The management of the BYOD conundrum is exacerbated by the fact that a single user could have one, two, three or more mobile devices. These devices may always be on and connected. They may also contain sensitive company information, personal information, videos, images, presentations, emails, calendars, etc. The National Institute of Standards and Technology (NIST) lists the high-level threats and vulnerabilities of mobile devices: lack of physical security controls, use of untrusted mobile devices (BYOD), use of untrusted networks (mobile hotspots in hotels, restaurants or home networks), use of untrusted applications (free applications in the app stores), interaction with other systems, use of untrusted content, and the use of location services (NIST, 2013). With all devices identified a company can not isolate and remove equipment from the network that does not meet the company's security requirements.

Raphael Simmons, dejacpp@gmail.com

2. Apple iOS and MacOS

Apple equipment is well suited for BYOD use. The software and hardware are under strict control on Apple devices, a benefit that makes them the most secure smart devices on the market. Apple does not allow Java or Adobe Flash to run on their iOS. Both applications have a history of vulnerabilities, thus excluding them reduces the threat of malware infections (Rai, 2017; Tan, 2016). The important features of Apple's security concept are source vetting and application sandboxing (Rai, 2017). The drawback is that users cannot easily add applications that will change the device's configuration and the availability of applications are limited to Apple's App Store.

2.1. Security features of the Apple devices

Apple iOS is based on Apple's user interface for Mac OS X developed in 2001(Miller, 2009). In 2007 when the first iPhone went to market, Apple stated that it would not allow third-party applications to execute on their devices (Miller 2009). The only option was to use a web application accessed by the iPhone's built-in web browser Safari. The initial release of iOS was not as secure as it is today. Applications were executed as the root user and could access all the device's resources.

Apple's closed-source paradigm starts at the hardware level. The effective integration between the instructions embedded in the firmware and the boot kernel validates the state of the device. The process of booting an iOS device starts with the secure boot chain that protects low-level software from rootkit attacks (BCS, 2013; Tse, 2016). At the start of the boot chain, the iOS application processor reads and executes the code from the Boot ROM (Read Only Memory). This code in the Boot ROM will start the trust relationship and contains Apple's Root CA public key which is used to authenticate the iBoot bootloader before allowing it to load. When the bootloader process is complete, it will confirm and run the iOS operating system kernel (BCS, 2013; Miller, 2009; Tse, 2016).

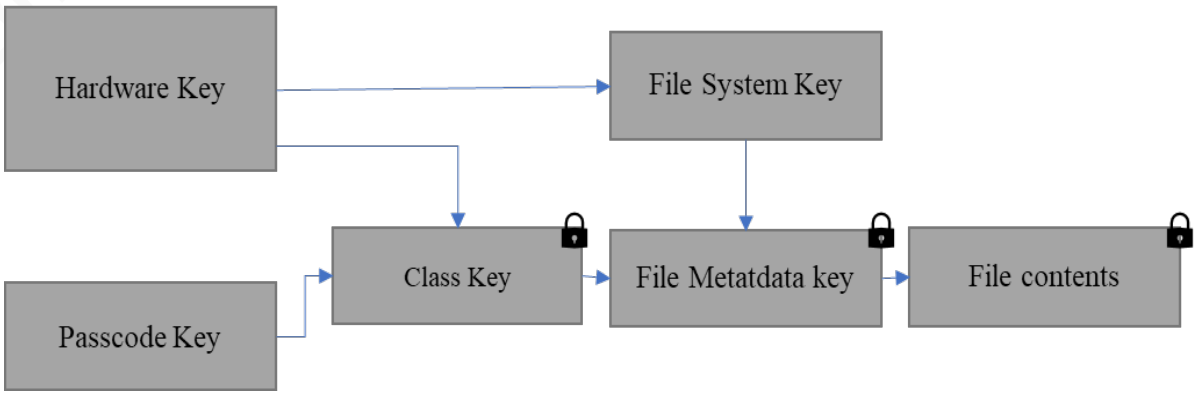
Raphael Simmons, dejacpp@gmail.com

If even one step in the boot process fails, start-up is halted, and the device will enter recovery mode. If the boot ROM cannot load, the device enters Device Firmware Upgrade (DFU) mode. In both cases, to maintain the integrity of the phone, the iOS device must be connected to iTunes via a USB cable and restored to factory default settings (Theil, 2016).

The startup process protects the iOS device and ensures that only Apple-signed code can be installed on the device. A process called System Software Authorization (SSA) is used to prevent iOS devices from being downgraded to an older version of the operating system, which can expose it to previously discovered security issues (Apple, 2017; Theil, 2016; Tse, 2016). Every Apple device has a unique ID (UID) that is embedded in the phone when it is manufactured. The UID is combined with other cryptographic keys to communicate within the device to determine if it is authorized for an upgrade and what components of the OS will be upgraded.

All data on an iOS device is protected with encryption. Apple uses its Data Protection (DP) technology to encrypt all information on the device (Apple, 2017; Tse, 2016). Figure 3 depicts the encryption process (Apple, 2017). When a file is created, the

Figure 3 iOS Encryption Overview



Raphael Simmons, dejacpp@gmail.com

DP generates a new 256-bit key (per-file key) and sends it to the Advanced Encryption Standard (AES) engine. The AES engine uses this key along with other encryption keys to encrypt the file as it is written to memory.

Software developers must register to have their software advertised on Apple's App Store. Application developers can download the SDK for free, but to publish an application on Apple's App Store registration is required. The registration fee is $99.00 a year. Apple's Software Development Kit (SDK) incorporated all the tools necessary to develop applications for the Apple device. Every application released on the App Store is tested and signed by Apple to ensure the software adheres to their software development standard.

2.2. Vulnerabilities

Given enough time and hardware access, hackers will eventually find a way to circumvent the security features of a device. With the release of each new and improved version of iOS, the jailbreaking community releases procedures to root the device to transfer complete control of the equipment to the owner minus the restrictions imposed by Apple. Table 2 lists the most popular software for jailbreaking iOS equipment along with the iOS version the software is capable of rooting. Jailbreaking an iOS device and rendering useless the security measures made by Apple's use of encryption, exposes the device to malware, remote access, prevents software updates and voids the warranty on the device. The iOS application Sandbox which limits software access to preferences, network resources, and data, will be vulnerable to untrusted software downloaded to the device which is a high price to pay for jailbreaking the equipment (Apple, 2007; Miller, 2009).

Raphael Simmons, dejacpp@gmail.com

Table 2 Jailbreaking Software by iOS & Device		
	iPhone, iPad, iPod	iOS
GreenPois0n	Yes	3.2.2 – 4.1
Redsn0w	Yes	6.0 – 6.12, 7.0 – 7.06
Pangu	Yes	7.1 - 9
TaiG	Yes	8.0 – 8.4

iOS devices that have not been jailbroken are susceptible to attacks. Figure 4 was extracted from Mitre Common Vulnerability and Exposures (CVE) database and displays how different attack vectors have been exploited and have increased since the first iPhone hit the market (CVE, 2017). For security reasons, Apple will not disclose, discuss, or confirm a security vulnerability until they have completed an investigation and patches are released. Apple released iOS 11 on 19 September 2017, which corrected several security issues identified in 2016 and 2017(Apple, 2017).

Figure 4 CVE Table

Vulnerability Trends Over Time

Year	# of Vulnerabilities	DoS	Code Execution	Overflow	Memory Corruption	Sql Injection	XSS	Directory Traversal	Http Response Splitting	Bypass something	Gain Information	Gain Privileges	CSRF	File Inclusion	# of exploits
2007	1		1	1											
2008	9	3	2		1						2				
2009	27	10	6	2	4		2			3	7				1
2010	32	14	14	9	6					5	3	2			3
2011	37	13	10	5	6		3			2	11	1			
2012	112	74	69	64	60		7			13	9	1			
2013	96	58	50	42	47		4			17	9	1			
2014	122	50	51	35	33		1	1		20	25	4			
2015	387	232	211	183	191			5		44	63	13	1		1
2016	161	107	78	85	75		3			8	39	11			
2017	291	177	169	152	140		12			30	47	5			
Total	1275	738	661	578	563		32	6		142	215	38	1		5
% Of All		57.9	51.8	45.3	44.2	0.0	2.5	0.5	0.0	11.1	16.9	3.0	0.1	0.0	

Raphael Simmons, dejacpp@gmail.com

2.3. How to protect iOS Devices

Though Apple uses significant security measures that protect their products from malicious intent, it is not perfect, and there are multiple avenues for harm. For Apple BYOD users, the best way to protect his or her device is to (Miller, 2009; Miller 2012; Rai, 2017):

- Keep it updated and patched.
- Install iOS updates when they become available. These updates fix security issues and can restore corrupt files.
- Treat the device like a credit card or checkbook.
- Enable device lock and keep the screen clean.
- Configure the device to "Ask to Join Networks," to prevent the device from inadvertently and automatically joining a rogue network.
- Use caution when downloading applications from the App Store.
- Encrypt the device and do not store passwords on the device.
- Use antivirus software and keep the software updated.
- Never set a web page to remember a password.
- On international travel, place the device in airplane mode with all communication radios disabled.

Though Apple does a reliable job of vetting software, there have been instances of malicious programs slipping through the curation process. Users should uninstall unneeded applications. If a user is planning to travel abroad, it may be wise to consider leaving the device home to prevent accidental international charges, device compromise or losing the equipment. Another option would be to purchase a low-end device with the minimum features needed for international travel. The Center for Internet Security benchmark guide for the Apple iOS can be used to lock down the device (CIS, 2016).

Raphael Simmons, dejacpp@gmail.com

3. Android OS

The Android OS was released by Google in 2007 and is currently the most common mobile operating system (Doherty, 2016; Net Application, n.d.). The Android OS is an open source operating system which allows the interested individual the freedom to download and customize the operating system. An individual or organization implementing an Android device, e.g., mobile phone, tablet, watch or other devices, must meet the minimum standard of Google's Android Compatibility Definition (Google, 2017).

3.1. Security features of Android devices

Phone manufacturers, e.g., HTC, Samsung, Sony, and LG have their CPUs customized by ARM's 64-bit SoC. The Android Compatibility document, Section 9.9, which covers the data storage encryption, and states (Google, 2017):

"If device implementations support a secure lock screen as described in Section 9.11.1, they:

- [C-1-1] MUST support data storage encryption of the application's private data (/data partition), as well as the application shared storage partition (/sdcard partition) if it is a permanent, non-removable part of the device. If device implementations support a secure lock screen as described in section 9.11.1 and support data storage encryption with Advanced Encryption Standard (AES) crypto performance above 50MiB/sec, they:
- [C-2-1] MUST enable the data storage encryption by default at the time the user has completed the out-of-box setup experience. If device implementations are already launched on an earlier Android version with encryption disabled by default, such a device cannot meet the requirement through a system software update and thus MAY be exempted.

- SHOULD meet the above data storage encryption requirement via implementing File Based Encryption (FBE)."

Samsung released the KNOX Platform in 2013 that includes device security, application security, and mobile device management for selected device models to address the security concerns on their Android devices (Samsung, 2017). When the device is manufactured, security is embedded in the CPU licensed to use KNOX. The hardware security components include:

- Device-Unique Hardware Key(DUHK) – a unique symmetric key used by the cryptographic module. Encrypted data is bound to the device and cannot be decrypted on another device.
- Samsung Secure Boot Key (SSBK) – a unique asymmetric key pair used to sign code for the boot components.
- Rollback Prevent Fuses (RP Fuses) – one-time programmable fuse used to identify approved bootloaders (kernels), prevents old kernels from loading.
- KNOX Warranty Fuse – one-time programmable fuse. Protects the device from rooting. If this fuse is set for any reason, the enterprise data on the device will no longer be accessible.
- ARM TrustZone Secure World – hardware isolated environment where cryptographic and monitor operation occur. Normal operation cannot access this environment.
- BootLoader ROM – Read Only Memory (ROM) that protects the primary bootloader code that starts the boot process.
- Device Root Key (DRK) – a unique asymmetric key pair that proves the device was created by Samsung

Raphael Simmons, dejacpp@gmail.com

At startup, KNOX uses a Secure Boot process that establishes a signature recognition chain to verify the integrity of all components in the boot process. Like iOS, if there is a signature failure at any step during the startup process the device will not boot. After the Secure Boot, the Trusted Boot will store and compare the cryptographic hash of the next component in the boot sequence.

This stage of the startup process prevents the loading of a previous or outdated kernel. When the kernel has loaded, the Real-Time Kernel Protection (RTKP) starts. This process protects the kernel from modification while it is running. Once the boot process is complete, the Samsung device enters the Android Framework state with two separate work areas, the personal and the KNOX workspace environments. An MDM tool can now manage the two segments.

Software development for the Android device can be accomplished using Android Studio, which includes an emulator that simulates the typical features of an Android device. Application developers must pay a one-time $25.00 registration fee to have their application advertised on the Google Play. Google Play does not have a software vetting process that is as stringent as Apple's. Developers are required to sign a "Google Play Developer Distribution Agreement" (Google Play, 2017). Developers must use Google's Verify, Bounce, and Play Protect. These tools aid in protecting the device from malicious code execution.

3.2. Vulnerabilities

The same vulnerabilities that exist for iOS devices exist for Android devices. The open-source paradigm has benefits but has a foreboding security problem with the open-source freedom; developers with unethical aims can modify an open-source application to attack the device. For Android devices, Google Play is not the only place an end user can download software. There are many third-party application sites with a

Raphael Simmons, dejacpp@gmail.com

plethora of ‘free' software tools. Many of these tools are poorly written applications that may contain ‘free' viruses, adware or ransomware.

McAfee’s 2017 Threat Report, “What Lies ahead for 2017”, identifies three significant categories of invasion to the mobile market (McAfee, 2017):

(1) Ransomware infiltrating the smartphone and other connected devices.
(2) Dead software that has been removed from app stores but still resides on end user’s devices – the invisible threat.
(3) IoT vulnerabilities.

McAfee identified over 4,000 applications withdrawn from Google Play. However, end users were never notified and thus were not able to protect themselves from potentially malicious software. In another report from Kaspersky Labs, mobile malware attacks rose 300% from 2015 to 2016 (Kaspersky, 2016; Kaspersky, 2017). The reports validate the need for company and end-user responsibility to secure Android devices.

3.3. How to protect an Android Device

The same precautions identified in section 2.3 for iOS devices also apply to the Android class of devices with the added caveat – beware of third-party application stores and free software. It is a difficult, time consuming and an expensive task for Google to examine and ensure that all the software available in the Google Play store is safe, some third-party stores may not even supply a testing process. For example, a tool advertised on Google Play that could help Instagram users gain followers contained a Trojan. (Miller, 2009). The malicious tool took users to a fake website that was indistinguishable from the legitimate Instagram site and captured the unsuspecting user’s credentials. The Center for Internet Security provides a valuable benchmark guide for locking down Android devices (CIS, 2016).

Raphael Simmons, dejacpp@gmail.com

4. Mobile Device Management

Mobile device management is a management technology used to control, monitor, and inventory smart devices that are connecting to a network infrastructure. MDM tools can be organized into three models: a device-centric model which uses the capabilities and features of the device's platform to manage the device; a data-centric model to focus on protecting and securing data and content; and a hybrid model which provides both device and data management.

4.1. MDM Risk Model

The first step in implementing an MDM security plan is to determine the data that must be protected:

- Is the data corporate, intellectual, customer, government or financial data?
- Why does the data have to be protected—is it government or industry mandated or just the best industry practice?
- What are the value of the data and the cost of a data breach?
- What is the data being protected from —internal or external threats, data-thieving, device control, or system access?
- What constraints will prevent the organization from protecting the data —is broad access required?

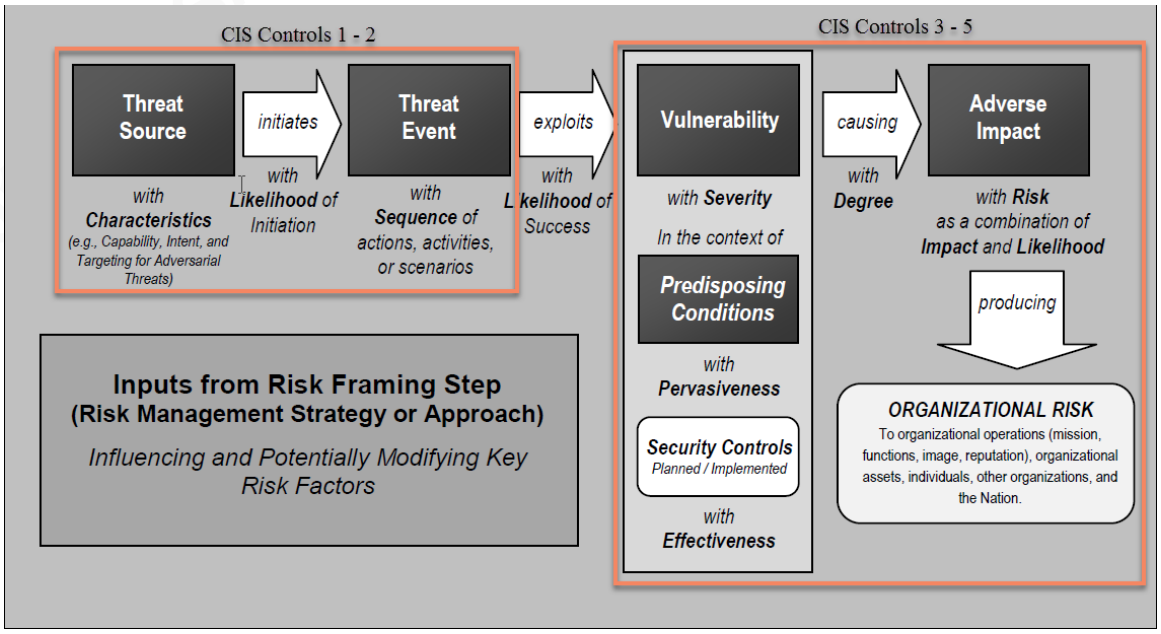
The Center for Internet Security (CIS) lists twenty critical security controls organizations should use to protect their networks (CIS, 2016). The first five controls are vital in securing a network. Studies conducted by CIS indicate that implementing the first five controls will be enough to defend against the most common cyber-attacks and to assist in developing a BYOD security plan (CIS, 2016). The first five controls are: (1) inventory of authorized and unauthorized devices, (2) inventory of authorized and unauthorized software, (3) secure configuration for hardware and software on mobile devices, laptops, workstations, and servers, (4) Raphael Simmons, dejacpp@gmail.com

continuous vulnerability assessment and remediation, and (5) controlled use of administrative privileges.

Combining the CIS controls with NIST 800-30 Rev 1, Guide for Conducting Risk Assessments, a simple and effective risk model, can be derived for a BYOD implementation (NIST, 2012). NIST 800-30 lists four processes for risk management:

- Framing – describes the environment.
- Assess – identify threats to the organization in terms of assets, operations, and individuals.
- Respond – develop, evaluate, and determine risk tolerance and implement appropriate risk response.
- Monitor – continuous evaluation of the risk management process.

Below is the generic risk model from the NIST guide that indicates where the critical controls apply. The goal is the ensure the security triad of confidentiality, integrity, and availability of the data for the business infrastructure and the end user.



Raphael Simmons, dejacpp@gmail.com

Conducting an inventory of the hardware and software is a primary goal for network infrastructure and BYOD amplifies the need for these requirements. Simply put, if a business cannot identify or does not know what is on their network, the company will be incapable of protecting their assets.

4.2. Policy Acceptance and Enforcement

Much of the research on BYOD focuses on the following items that must be mutually agreed upon by the employer and employee (Downer, 2016; Flores, 2016; Tan 2016): keep personal information private and separated from corporate data; mobile device enrollment must be simple; enrollment should be done over the air; and the continual monitoring of device compliance.

In conjunction with developing an MDM security plan, is the development of an acceptable use policy (AUP) which informs users how they are expected to use their devices and software regarding company work. There should be procedures the IT staff will use with the MDM tool to enforce and confirm AUP compliance (Downer 2016; Miller 2012; Flores, 2016). Because the mobile device belongs to the employee, the risk factor associated with allowing a user to connect his or her personal device to the corporate network must be clearly understood by both parties. End-user education and responsibility should include guidance on reporting procedures if a personal device is lost or stolen, device encryption requirements, device locking/screen locking, anti-virus/malware protection, and the security tips mentioned in sections 2.3 (iOS) and 3.3 (Android).

4.3. MDM Tool Minimum Requirements

An MDM tool should provide inventory management to identify the device model, device ID, firmware version, OS type, MAC address and attached memory cards. The tool should have the capability to periodically poll devices to update the inventory records. BYOD provisioning to allow the company to configure and ensure the equipment meets the company's security policy's for password length, screen Raphael Simmons, dejacpp@gmail.com

lock settings, data encryption and remote wipe. The MDM tool should allow end users to register their device and confirm their acceptance of the company's BYOD policy.

4.4. Current trends

The current state of mobile device management is evolving into a set of tools called Enterprise Mobility Management (EMM) suite. In addition to the MDM tool, the EMM suite will include a Mobile Application Management (MAM) tool to deploy in-house developed software, third-party software applications, and volume licensed applications from Apple's App Store and Google Play. A Mobile Identity (MI) is used to ensure only trusted devices and users have access to the network infrastructure. The MI tool uses the device's certificates, authentication, location, and time to determine device access.

An EMM suite will also have a Mobile Content Management (MCM) tool to provide: access controls for company files, data loss protection, file sharing rules, cut/paste restrictions, and digital rights management. The final feature of an EMM suite is Containment, which is used to encapsulate personal data from business data. The idea of containment is to create an enclave where company data and policies can reside in lieu of the device.

5. Open Source and Free MDM tools

The number of free and open source tools for MDM is small. The available tools represent a viable solution for small businesses that do not want to incur the annual cost of a vendor solution. For example, VMware AirWatch Green Management suite which includes an AirWatch MDM, AirWatch Container, and AirWatch Cataloging cost $10,400 per year to manage 200 devices or $4.33 per device a month (VMware, 2017). However, using a free and opensource tool, such as

Raphael Simmons, dejacpp@gmail.com

Miradore, Apple Configurator 2.0, or WSO2 can be used to reduce the cost significantly.

5.1. Solution for small businesses

Miradore is a free online solution for MDM management. It supports Android, iOS, Windows Phone, Windows desktops and laptops with an unlimited number of device registrations. ManageEngine MDM tool is another free tool but has a device limit of twenty-five and supports iOS, Android and Windows devices. ManageEngine has two available options: an online solution or their on-premises edition – a software download for installation on a Windows server.

Apple Configurator 2.0 is a free device management tool for Apple devices only (Apple, 2017). A computer running MacOS is required. There are two options for enrollment: automatic enrollment which requires an MDM tool or manual enrollment which requires employees to bring their device to the IT department for admission.

WSO2 is an open source MDM tool and much more (WS02, 2017). It can also be used to manage Internet of Things (IoT) devices. WSO2 can manage iOS, Android, Android Sense, Windows, Arduino, and Raspberry Pi devices. The features of WSO2 are comparable to vendor class MDM tools without the licensing and the annual cost. WSO2 can be deployed on any platform that is Java Development Kit (JDK) 8 compliant. The WSO2 website documentation page has all the information necessary to configure a server: a quick start guide, tutorials, frequently asked questions and other self-service documentation.

Raphael Simmons, dejacpp@gmail.com

6. Conclusion

Reducing the security threats caused by a BYOD policy can be expensive. For the small company, implementing a BYOD policy does not have to be a high cost. Proper planning, research, and testing of a free or open-source tool can be a viable solution for the small organization on a limited budget. Network security is a challenge for a business of any size. Excluding the external threat, the internal user threat is tough enough to monitor, detect and correct. The capabilities of mobile devices have increased the threat radius by allowing users to access a company's network from a distance. As more companies adopt the BYOD paradigm, security research will have to be at the forefront to ensure the protection of company and personal information.

Raphael Simmons, dejacpp@gmail.com

References

Ali, S., Qureshi, M. N., Abbasi, A. G. (2015). Analysis of BYOD Security Frameworks. IEEE.

Apple. (2017). iOS Security, iOS 10. Retrieved from https://www.apple.com/business/docs/iOS_Security_Guide.pdf.

Apple Configurator 2.5. (2017). Retrieved from https://support.apple.com/en-us/HT208040.

BCS, The Charter Institute for IT. (2013) Bring Your Own Device(BYOD): The Mobile Computing Challenge. London: BCS.

Center for Internet Security (CIS). (2016). CIS Apple OSX 10.12 Benchmark. Retrieved from https://www.cisecurity.org.

Center for Internet Security (CIS). (2017). CIS Google Android 7 Benchmark. Retrieved from https://www.cisecurity.org.

Center for Internet Security (CIS). (2016). The CIS Critical Security Controls for Effective Cyber Defense Version 6.1. Retrieved from https://www.cisecurity.org.

Center for Internet Security (CIS). (2016). Guide to the First 5 CIS Controls Version 6.1. Retrieved from https://www.cisecurity.org.

CVE Common Vulnerabilities and Exposures. (2017). Retrieved October 1, 2017, from https://cve.mitre.org.

Doherty, J. (2016). Wireless and Mobile Device Security. USA: Jones and Bartlett Learning.

Downer, K., Bhattacharya, M. (2016) BYOD Security: A New Business Challenge. IEEE.

Flores, D.A., Qazi, F., Jhumka, A. (2016). Bring Your Own Disclosure: Analyzing BYOD Threats to Corporate Information. (IEEE).

Google. (2017, September 1). Compatibility Definition Android 8.0. Retrieved from https://source.android.com/compatibility/android-cdd.pdf.

Raphael Simmons, dejacpp@gmail.com

Google Play. (2017, May 17). Google Play Developer Distribution Agreement. Retrieved from https://play.google.com/about/developer-distribution-agreement.html.

Jodoin, E. (2015). Accessing the inaccessible: Incident investigation in a world of embedded devices. SANS Institute InfoSec Reading Room.

Johnson, K. (2012). SANS Mobility/BYOD Security Survey. SANS Institute InfoSec Reading Room.

Kaspersky. (2017. Kaspersky Security Bulletin. Retrieved from https://kasperskycontenthub.com/securelist/files/2016/12/Kaspersky_Security_Bulletin_2016_Statistics_ENG.pdf.

Kok Kee, C. (2001). Security Policy Roadmap – Process for Creating Security Policies. SANS Institute InfoSec Reading Room.

Kim, K., Hong, S. (2014). Study on Enhancing Vulnerability Evaluations for BYOD Security. International Journal of Security and its Applications.

ManageEngine. (2017). Mobile Device Management Retrieved from https://www.manageengine.com/mobile-device-management.

Miller, C. (2009). iOS Hacker's Handbook. Wiley Publications.

Miller, K.W, Voas, J., Hurlburt, G.F. (2012). BYOD: Security and Privacy Considerations. IEEE.

Miller, M. (2015). BYOD Do You Know Where Your Backups Are Stored? Global Information Assurance Certification Paper.

Miradore. (2017). Mobile Device Management Retrieved from https://www.miradore.com/mobile-device-management.

National Institute of Standards and Technology (NIST). (2012). Special Publication 800-30 Revision 1 Guide for Conducting Risk Assessments. Retrieved from http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-30r1.pdf.

National Institute of Standards and Technology (NIST). (2013). Special Publication 800-124 Revision 1 Guidelines for Managing the Security of Mobile Devices in the Enterprise. Retrieve from http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-124r1.pdf.

Raphael Simmons, dejacpp@gmail.com

O'Donnelly, C. (2017). White Paper: The Cost of Not Acting for Managed Mobility Services. http://bluehillresearch.com.

McAfee. (2017). Mobile Threat Report: Trojans, Ghosts, and More Mean Bumps Ahead for Mobile and Connected Things. Retrieved from https://www.mcafee.com/us/resources/reports/rp-mobile-threat-report-2017.pdf.

Net Applications. (n.d.). NETMARKETSHARE. Retrieved from https://www.netmarketshare.com/operating-system-market-share.aspx?qprid=8&qpcustomd=1.

Rai, S., Chukwuma, P., Cozart, R. (2017). Security and Auditing of Smart Devices: Managing proliferation of Confidential Data on Corporate and BYOD Devices. UK: Auerbach Publications.

Samsung. (2017). Samsung Knox Security Solution. Retrieved from https://www.samsungknox.com/docs/SamsungKnoxSecuritySolution.pdf.

Smith, R., Taylor, B., Bhat, M., Silva, C., Cosgrove, T. (2017) White Paper: Magic Quadrant for Enterprise Mobility Management Suites. https://www.gartner.com/home.

Tan, V. (2016).White Paper: Bad For Enterprise, Attacking BYOD Enterprise Mobile Security Solutions. https://www.blackhat.com/docs.

Thiel, D. (2016). iOS Application Security: The Definitive Guide for Hackers and Developers. No Starch Press.

Tse, D., Wang, L., Li, Y. (2016). Mobility Management For Enterprises In BYOD Deployment. IEEE.

VMware. (2017). AirWatch Enterprise Mobility Management. Retrieved from https://www.vmware.com/products/airwatch-enterprise-mobility-management.html.

Wright, J. (2013). Fear and Loathing in BYOD. SANS Institute InfoSec Reading Room.

WS02. (2017). Open Source Mobile Device Management Software. Retrieved from http://wso2.com/platform.

Raphael Simmons, dejacpp@gmail.com

Last Updated: March 21st, 2018

Upcoming Training

Mentor Session AW - SEC504	Oklahoma City, OK	Mar 23, 2018 - Apr 27, 2018	Mentor
SANS Boston Spring 2018	Boston, MA	Mar 25, 2018 - Mar 30, 2018	Live Event
SANS 2018	Orlando, FL	Apr 03, 2018 - Apr 10, 2018	Live Event
SANS 2018 - SEC504: Hacker Tools, Techniques, Exploits, and Incident Handling	Orlando, FL	Apr 03, 2018 - Apr 08, 2018	vLive
Pre-RSA® Conference Training	San Francisco, CA	Apr 11, 2018 - Apr 16, 2018	Live Event
SANS Zurich 2018	Zurich, Switzerland	Apr 16, 2018 - Apr 21, 2018	Live Event
Community SANS Kansas City SEC504	Kansas City, MO	Apr 16, 2018 - Apr 21, 2018	Community SANS
SANS London April 2018	London, United Kingdom	Apr 16, 2018 - Apr 21, 2018	Live Event
SANS Baltimore Spring 2018	Baltimore, MD	Apr 21, 2018 - Apr 28, 2018	Live Event
Baltimore Spring 2018 - SEC504: Hacker Tools, Techniques, Exploits, and Incident Handling	Baltimore, MD	Apr 23, 2018 - Apr 28, 2018	vLive
SANS Seattle Spring 2018	Seattle, WA	Apr 23, 2018 - Apr 28, 2018	Live Event
Mentor Session - AW SEC504	Alexandria, VA	Apr 27, 2018 - May 04, 2018	Mentor
SANS Riyadh April 2018	Riyadh, Saudi Arabia	Apr 28, 2018 - May 03, 2018	Live Event
Community SANS Toronto SEC504	Toronto, ON	Apr 30, 2018 - May 05, 2018	Community SANS
Automotive Cybersecurity Summit & Training 2018	Chicago, IL	May 01, 2018 - May 08, 2018	Live Event
SANS SEC504 in Thai 2018	Bangkok, Thailand	May 07, 2018 - May 12, 2018	Live Event
SANS Security West 2018	San Diego, CA	May 11, 2018 - May 18, 2018	Live Event
SANS Melbourne 2018	Melbourne, Australia	May 14, 2018 - May 26, 2018	Live Event
Community SANS Columbia SEC504	Columbia, MD	May 14, 2018 - May 19, 2018	Community SANS
Community SANS Detroit SEC504	Detroit, MI	May 16, 2018 - May 23, 2018	Community SANS
SANS Northern VA Reston Spring 2018	Reston, VA	May 20, 2018 - May 25, 2018	Live Event
Community SANS Phoenix SEC504	Phoenix, AZ	May 21, 2018 - May 26, 2018	Community SANS
Mentor Session - SEC504	Dulles, VA	May 24, 2018 - Jun 28, 2018	Mentor
SANS Amsterdam May 2018	Amsterdam, Netherlands	May 28, 2018 - Jun 02, 2018	Live Event
SANS Atlanta 2018	Atlanta, GA	May 29, 2018 - Jun 03, 2018	Live Event
SANS Rocky Mountain 2018	Denver, CO	Jun 04, 2018 - Jun 09, 2018	Live Event
SANS London June 2018	London, United Kingdom	Jun 04, 2018 - Jun 12, 2018	Live Event
Mentor Session - SEC504	Milwaukee, WI	Jun 06, 2018 - Jul 25, 2018	Mentor
Mentor Session - SEC504	San Francisco, CA	Jun 06, 2018 - Aug 01, 2018	Mentor
SANS Crystal City 2018	Arlington, VA	Jun 18, 2018 - Jun 23, 2018	Live Event
SANS Oslo June 2018	Oslo, Norway	Jun 18, 2018 - Jun 23, 2018	Live Event