SMS, iMessage and FaceTime security

GIAC (GCIH) Gold Certification

Author: George Khalil, George@GeorgeKhalil.com
Advisor: Dale Emel

Accepted:
(Date your final draft is accepted by your advisor)

Abstract

Apple introduced FaceTime to the world with the release of iOS 4 in 2010 bringing mobile video calling to the masses. Apple introduced a new feature rich encrypted instant messaging platform (iMessage) on iOS 5 in 2011. In 2012, Apple replaced iChat with Messages in OSX mountain lion featuring iMessage and FaceTime capabilities. Steve Jobs stated that Apple products "just work" at WWDC 2011. That very same spirit of simplicity, multi-platform support and transparency introduced some vulnerabilities and exposed an attack surface. Although Apple implemented various defenses to protect users, there are attack surfaces and avenues still available, which this paper will explore. We will explore mechanisms and underlying protocol and how they functions, potential attack surface and what defenses are in place or can be easily implemented to defend the clients.

1. Introduction

With the increasing mass adoption of mobile smart devices, attackers are increasingly focusing on gaining access and visibility into the data stored and transmitted via mobile devices. With approximately 600 Million iOS devices sold (Cutler, 2013) vs the estimated 2 Billion PC's (Gartner, 2008) corporate and personal data is increasingly transmitted using mobile devices. The most common form of cellular communication is SMS messaging which works across all cellular carriers globally and across the majority of cellular devices. With the introduction of iOS 4 in 2010 Apple introduced Facetime and introduced iMessage in 2011 providing encryption and features-rich messaging to iOS based devices which were added later in 2012 to OSX Mountain Lion. Apple implemented multiple security measures to protect iMessage communication which we'll review later, however we'll first review several phyisical, social engineering and communication interception potential exploits and defense measures against mobile communication protocol attacks.

1.1. SMS

On December 3, 1992 the first text SMS message was transmitted to a mobile device via Vodafone's UK GSM network; it sent the seasonal greeting, "Merry Christmas". By 1995, the average US user sent 0.4 text messages per month. By 2005, more than 1 trillion text messages were sent globally. Currently 25,000 text messages are globally sent each second with estimate of 10 trillion messages sent in 2012 (Ericsson.com, 2013). In comparison, global e-mail traffic for 2012 is estimated at 144.8 billion messages per day with an estimated 15% spam by volume (Sara Radicati & Hoang, 2013). Historically e-mail has been the largest digital communication of choice with spam is estimated at 99% of all global email traffic, ISP's have been deploying extensive spam filters blocking 98% of transmitted spam (CloudMark, 2013). However, SMS spam filtering is still at it's infancy which leaves it as prime target for spear phishing targeted spam attacks. Corporations are deploying spam filters and human awareness training on how to spot phishing emails, but SMS is not viewed as a high security risk because it's outside of the average corporate control.

1.2. iMessage and Facetime

Apple introduced iMessage in iOS 5 for iPhone and iPad users in 2001. iMessage was introduced to enhance the traditional text messaging on "iDevices", while maintaining compatibility with traditional messaging thereby providing users with feature-rich transparent messaging. When users activate their iPhone, their number is entered in a database maintained by Apple. When a user sends a message to another number from an iPhone, it quickly verifies if the recipient is active in the Apple iMessaging user database confirming the message was only sent via apple servers, otherwise it is routed via the cellular carrier's SMS gateway. On non-cellular devices such as iPad's and later OSX messaging, as well as iPhones, the use of an email address was introduced as a recipient address. iMessage provides users with transparent encryption capabilities to protect communication in transit as well as larger quota limits in comparison to traditional messaging. iMessage is sent via the subscriber's data plan or Wi-Fi bypassing the cellular carrier's charges and limits. Users are also provided delivery and optional read notifications; both are not available via traditional messaging (Spencer, 2013).

1.2.1. FaceTime

FaceTime was introduced before iMessage with the release of iOS 4 in 2010. FaceTime and iMessage registration with Apple's servers takes place automatically when a SIM card swap occurs. On SMS capable devices, the phone registers its phone number as FaceTime address via an SMS exchanges that occur in the background without requiring user intervention or knowledge (Hollington, 2013). The SMS exchange validates that the user has a valid SIM card trusted and authenticated by the mobile carrier. This transaction alone is sufficient to proceed with sending and receiving iMessage and FaceTime communication. In addition to using a phone number, users are able to sign in using their Apple ID account and use their email as a recipient address. Using the Apple ID login is solely based on the user knowing the password for that account. A noteworthy item is that Apple allows users to log into multiple devices using the same FaceTime and iMessage account to support the Apple ecosystem of iOS and OSX devices.

2. Attacks

2.1. SMS

SMS messaging is unencrypted data from the cellular device and transmitted to the cellular carriers via their communication towers. The transmission is sent to the carriers SMS gateway, which routes it to the appropriate destination. Although SMS communication is not encrypted, the overall cellular communication is encrypted using various encryption algorithms based on the cellular communication generation technology in use.

The weakest of cellular protocol is the 2G "Second Generation" mobile phone mobile communication protocol. The 2G (Edge) protocol is supported by all modern smartphones, including iPhone 5 and the latest Android phones. 2G is available to maintain compatibility with older networks and to support worldwide roaming. 2G (Edge) encryption consists of several modes of voice and data encryption, encryption such as A5/1, which was broken using publicly available rainbow tables with 90% probability in 5 seconds by Karsten Nohl at Blackhat in 2010. An additional mode is A5/2 mode, which is vulnerable to cypher text only attacks requiring only milliseconds of over-the-air traffic and seconds on a desktop computer to break the cypher. A5/3 was recently compromised by a related-key attack, which recovered the full 128-bit key within 112 minutes with a 50% success rate (Brown, Cecchetti, 2013). The cellular carrier or the attacker have a choice to disable data encryption using GEA/0 which uses no data. A5/0 offers no voice encryption when chosen. Protocol selection is determined using the cellular carrier's or an attacker's communication tower during phone association negotiation.

Due to the built in capability in each 2G phone to support unencrypted communication, an attacker can setup a malicious cellular tower (base station) using off the shelf hardware such as USRP (Universal Software Radio Peripheral) antennas; SMS relay device, Antennas, asterisk, OpenBTS software and Internet connection as demonstrated at Def-Con 18 by Chris Paget.

Once an attacker establishes a malicious base station, he/she has full control of the subscriber phone by negotiating encryption protocols, setting or disabling frequency

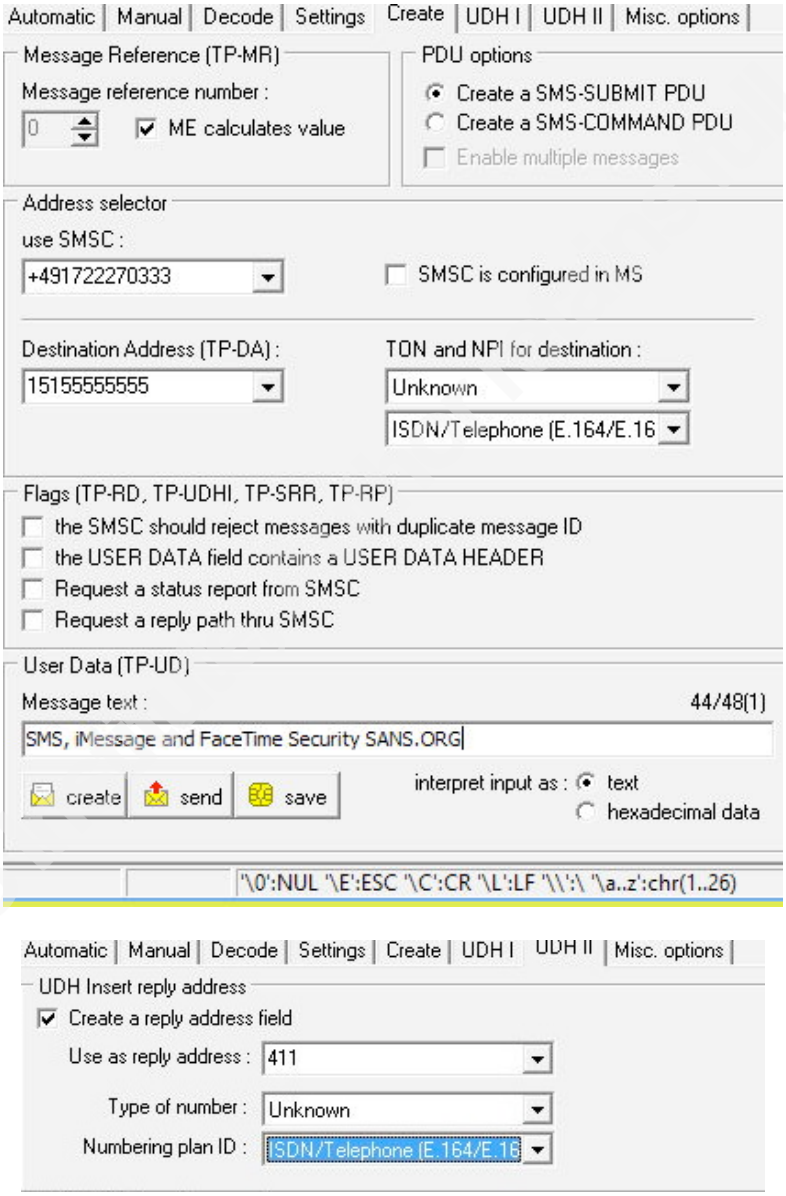
hopping, intercepting and relaying voice and SMS messages. The attacker also has the capability of capturing session keys then cracking them using rainbow tables. The attacker can choose the weakest encryption to expedite the key recovery time. The attacker can then take over the identity of the subscriber effectively executing a man in the middle attack.

2G networks are not only vulnerable to Edge type attacks, SMS traffic is sent along with the same channels used for call setup and control. An attacker can overwhelm cellular timing slots by a sustained SMS attack leading to a potential cellular denial of service. Security researchers calculate that Washington DC cellular network's deployment estimated at 120 sectors covering 68.2 square miles can suffer disruption in communication when attacked using 8,437.5 kbps of sustained SMS traffic. The estimated number of sectors deployed across a continent could be attacked using an estimated 370 Mbps of sustained SMS traffic disrupting all cellular communication (Ench, Traynor, McDaniel, La Porta 2013). The control channel SMS attack differs from jamming attack in its scalability and remote delivery capability.

SMS is a powerful platform to deliver information to end-users. Spammers and attackers have been increasingly targeting text messaging to advertise scam and defraud users. Security professionals are training users to identify malicious emails, but are mistakenly leaving out SMS and mobile communication. SMS currently is being used to deliver advertising, SPAM and a proof of concept SMS exploits. SMS transmitting botnets has been demonstrated for the Android platform (CloudMark-Blog, 2013). SMS provides no authentication of the sender with no certificate or signing capabilities. Carriers are starting to deploy spam filters, but large numbers of spoofed messaging continue to pass through and will increase in the future.

iPhone's running iOS below version 6.0 are vulnerable to an SMS spoofing flaw. Pod2G discovered that the iOS was displaying the reply to address rather than the sender's address. SMS messages sent from a personal computer, or the variety of jailbreak applications that shortly (McGee, 2013) became available, allowed the sender to configure a fake reply to email and trick the user into believing it came from someone else (Kalinchuk, 2013). An attacker can make the phishing message more plausable if

he/she can change the display name with a familiar name to the recipient, Thereby tricking the victim into provide confidential information or open a link of the attacker's choosing. Tools such as PDUSpy shown in Figure 1 demonstrate the simple steps needed to generate iPhone compatible spoofed SMS message.



Khalil, George

Figure 1: PDUSpy tool and the simple steps needed to generate iPhone compatible spoofed SMS message

After the raw code is generated using PDUSPY, it can be sent using the following command from the terminal of a jail-broken iPhone running an iOS prior to iOS 6.

./sendrawpdu [RAWDATA_from_SpyPDU]

2.2. iMessage and Facetime

2.2.1. Physical Attacks

Apple has two core methods of authenticating "iDevices" as valid on their network. The first method uses SMS capable devices (i.e. iPhones) by having a hidden 2 way SMS exchange providing the user with iMessage and FaceTime capabilities with no additional user intervention (Hollington, 2013). The second method validates the user identity using his/her Apple ID. Authentication is the only method to enable iMessage and FaceTime communication via a non-SMS capable device such as iPad's (including 3G enabled iPads) and OSX messaging.

SIM card only based authentication is extremely vulnerable to physical attacks (Ryu, 2013). The focus on simplicity and having iMessage work out of the box allows an attacker with physical access to the iPhone and its SIM card to hijack its messages. The following attack scenario has been tested and validated:

Setup:

Victim: iPhone 5, iOS 6.0 carrier locked with valid SIM card and activated iMessage; iMessage enabled using only phone number and no Apple ID. Wi-Fi connected.

Attacker: iPhone 5, iOS 6.1.3 carrier unlocked valid SIM card and activated iMessage using its unique SIM card with no Apple ID. Wi-Fi Connected.

Both iPhones were not jail broken or modified in any way from the default OEM software image. Both phones connected to the carrier networks and had internet access via Wi-Fi.

Attack:

Both phones were powered off with SIM card removed from attacker to speed up the time needed for the attack. The SIM card was removed from the victim's phone and inserted in to the attacker's phone. The attacker's phone was powered on, which registered with the victim's cellular carrier within seconds. iMessage was stuck in activating state; a second reboot resolved this issue and iMessage was active sending and receiving successful iMessages with the victim's phone number across Apple's network. The process took approximately 3 minutes. The SIM card was removed from the attacker's phone and inserted back into the victim's phone, the phone was then powered on. The victim's phone registered back with the carrier, iMessage was unaware of any changes and provided no notifications of any tampering. Both the victim and the attacker were able to send iMessages using the victim's phone number to any other iMessage capable device. The attacker's phone does not have a SIM card inserted at all, however it was connected via Wi-Fi. All iMessages sent from the victim's phone were also sent to the attacker's phone with no warning or errors to the victim. Replies from other people were only delivered to the victim's phone and not the attacker. iMessages sent from the attacker's phone were also copied to the victim's phone and appeared as if they were sent from the victim's phone. The attacker maintained the iMessage interception through a reboot and only lost it when a new SIM card was inserted or iMessage was disabled and re-enabled from the iPhone configuration. iMessages sent from the victim's device while the

attacker was offline were queued and were all delivered once the attacker came back online. The duration of the activation process varied after the swap occurred multiple times possibly due to backlog of the apple SMS validation server (Hollington, 2013) or a security triggers due to the multiple activation that occurred and switching of the same SIM card within a short period.

An attacker can have physical access to the iPhone in multiple scenarios, with the increasingly popularity of mobile payment services such as Starbucks, electronic airline tickets and other mobile payments services, users are willingly handing their devices to unknown parties with no security considerations by the users, the carriers, or Apple. With the increased adoption of "bring your own device", executives are handing their iPhones and credentials to staff for configuration or to Apple store employees for troubleshooting or mall repair stands for repairing cracked screens and cosmetic modifications. Social engineering attacks can be improvised to persuade users to hand over their device to an attacker which in turn does not need that much time to take over their iMessage communications, intercepting personal and possibly sensitive corporate data. FaceTime produced the same behavior such as the capability to make calls from the attacker's phone to a third party. This might be difficult to exploit due to the attacker being exposed alerting the receiver that they are not the person that the recipient is expecting; however, it provides a platform for further social engineering attacks.

An attacker can exploit the secondary method of authenticating users via their Apple ID, which can be obtained using shoulder surfing or getting a shared password. Unfortunately, Apple users have the tendency to share their Apple ID with other family members, teenagers or I.T. employees to manage or share their app store purchases across multiple devices. An attacker with the users' Apple ID can intercept all iMessage communications that are sent by the victim using their email address from iPhones, iPads or OSX messaging platforms. Shoulder surfing, social engineering, password guessing, possible extractions of encrypted keychain files can obtain credentials and brute forcing encrypted passwords offline (Proffitt, 2013) (Elcomsoft.com, 2013). SMS messages are also recoverable from iPhone backup files and can be recovered by having physical access to the computer that stores the iPhone backup archives.

2.2.2. Remote and network attacks

One of the first attacks that comes mind once we pivot to the network is denial-of-service. A number of iOS devices quickly became the victims of an iMessage denial of service (DOS) after Macs were allowed to send iMessages (Smith, 2013). Using a variation of Figure 2 code the Apple scripting language, an attacker can script the following, while looped, can essentially overwhelm the iMessage recipients and effectively DOS the victim.

```
set peopleIDontCareAbout to {"Pietje Piet", "Joe Anonymous"}
    tell application "iChat"
      repeat with myList in buddies
        --get properties of myList
        if full name of myList is in peopleIDontCareAbout then
          send "dfgdgdf gdg dfg dfg" to myList
        end if
      end repeat
    end tell
```

(tompaman, 2013)

Figure 2: Sample Apple script for automating iMessage transmission

Apple designed iMessage as an evolution and a fix for the lack of SMS security features. Apple's iMessage supports end-to-end encryption using TLS (Apple.com, 2013) using Apple's dedicated iMessage Certificate Authority and using a proprietary protocol developed by Apple (Green, 2013). Since iMessage is touting TLS encryption we can attempt to use known SSL/TLS man-in-the-middle attacks to decrypt the iMessage traffic. The biggest challenge remaining is the proprietary protocol that is being used which requires some reverse engineering. To prepare the victim's iDevice for the man-in-the-middle attack we need to get our attacker's certificate accepted as trusted. Security researchers have attempted to intercept and decrypt the iMessage communication protocols using the push proxy which is designed specifically to target iOS or OSX using

man-in-the-middle attacks (MEEEE, 2013). The Push Proxy author recommends using jailbroken iDevice as the method of choice of depoying the attacker's certificate. This method require us to make major modifications to the target device as well as get physical access to the device to acomplish this. However a more viable solution for remote deployment is to use Apple's build in enterprise management tool. Apple's enterprise management tool allows an administrator or an attacker to create a mobileconfig file and install their own custom push proxy certificate on the victim device. The user can be persuaded to install the mobileconfig file as a part of a web redirect on a webpage or a public Wi-Fi Acceptible Use/Terms of Service page (which is common in public locations) with the random certificate signing warning message. The mobileconfig file can be signed using any valid or compromised public certificate to get the valid green check mark on the screen and raise the victims confidence level (CRYPTOPATH, 2013).

Once the victim has connected to the attacker's Wi-Fi network and installed the push proxy certificate the attacker has full network access to perform the basics of Man-in-the-Middle attack by poisoning the victim's ARP cache and modify the DNS names of Apple's iMessage servers locally to decrypt the traffic (Burkholder, 2013). Push Proxy has some built in decryption functions as we'll see the small expert below in figure 3 from the imfreedom.org Wiki:

"Activation

This part looks very similar to the activation of iPhones: http://theiphonewiki.com/wiki/index.php?title=Activation_Token.

The connection of applepushserviced is encrypted with TLS using a client side certificate. To retrieve such a certificate, it posts to: https://albert.apple.com/WebObjects/ALUnbrick.woa/wa/deviceActivation?device=Mac OS (NOTE: this has a content type of "application/x-www-form-urlencoded", contrary to most other requests, it is shown unencoded here):

activation-info=<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">

<plist version="1.0">

<dict>

<key>ActivationInfoComplete</key>

<true/>

<key>ActivationInfoXML</key>

<data>

(again a plist, see next block)

</data>

<key>FairPlayCertChain</key>

<data>

(a certificate issued by "Apple FairPlay Certification Authority", where does this come from?)

</data>

<key>FairPlaySignature</key>

<data>

(about 3 lines, probably related to the previous certificate)

</data>

</dict>

</plist>"

(imfreedom.org, 2013)

Figure 3: Decoded iMessage protocol activation exchange

The security community is conducting Additional research to understand the content and functions of Apple's proprietary protocol after the encryption is removed. Larger decrypted communication sections and the reverse engineered functions are documented on imfreedom.org along with some unknown decoded communications. With decryption capabilities of iMessage being possible, it might be possible to return encrypted iMessage data to plain text once the Apple iMessage protocol is reverse engineered.

The previous attack is focused on the attacker having the victim connected to his or her Wi-Fi network or the attacker gaining access to the network that the user is connected. Researchers demonstrated Femto Cell modifications and capturing voice, data, SMS traffic (Ritter, DePerry, & Rahimi, 2013). If the attacker can inject spoofed arp and dns traffic into the femto cell, he or she can use that as an attack platform to launch the previously discussed attack scenario.

The Apple ID attack discussed in the physical attack section can still apply across the network either through brute forcing which will be very quickly block out by Apple. Previous opportunities presented themselves by Apple allowing an attacker to reset the victims Apple ID by merely knowning his or her birthday which can be obtained via web mining or social engineering (AppleInsider, 2013), The site was taken offline within hours after a mass public outcry. However, with new hardened Apple iForgot site, an attacker with sufficent information and knowledge about the victim can answer the password recovery question, and with access to the victim's email, reset the password. The main disadvantage of password resets revolve around the user quickly becoming aware of the attackers actions and will either contact Apple or reset the password thereby removing the attacker's access to their Apple ID.

3. Defenses

3.1. SMS

SMS is a legacy non-encrypted protocol. Most carriers rely on the underlying wireless transport protocol to provide encryption such as 3G, 4G and LTE, amongst others. The vast majority of the exploits are related to the backward compatibility with

2G protocol built into most modern phones. Carriers and manufactures of cellular devices should remove support for old non-secure 2G devices and networks. Some foreign countries prohibit encryption, but in the absence of an international standard, or treaty, other countries should not be bound to follow unilateral mandates intended to support the interception any citizens' communications. At a minimum, phone manufactures should provide software settings to enable or disable the 2G support as well as encryption functions in the firmware. Once security protocols are set, it should not be modifiable by cellular base stations, carriers or attackers for that matter. Foreign nations can set their support in firmware deployed to their users and accommodate their security requirements.

Several vendors are taking the initiative to solve SMS vulnerabilities by providing their own independent messaging applications. Although third party messaging providers are not directly modifying or altering SMS communication, they are offering a product similar to iMessage while providing their own encryption and authentication. A more elegant solution would be carriers and developers integrating a PKI encryption solution into their prospective SMS applications. For example, PGP is used to encrypt plain text email into cypher text across unsecure channels; PGP could be used to encrypt SMS messages across untrusted carrier channels while providing sender authentication and data privacy through encryption. iMessage is already providing PKI services through Apple. However, the lack of community review of the protocol and central management by Apple leaves an opportunity for community managed and user controlled PKI messaging solution.

SMS spam, phishing and denial of Service is an increasing attack vector due to its lack of authentication, exponential growth and user reach. Carriers are at varying stages of deploying send quotas, spam detection engines as well as pursuing legal avenues to shut down text messaging spammers. The protocol is inherently insecure due to its lack of sender authentication just like email. Carriers try to manage it at their SMS gateways, but there are worldwide providers willing to allow spammers to send bulk messages for minimal cost. Spoofing presents a challenge when providers attempt to track and shut down the source of SMS spam. Due to the increasing number of SMS spam fraud, Security professionals should include mobile messaging in its human awareness programs along with email. As professionals, we seem to have neglected to identify SMS

as a threat to our organizations. With the ever-increasing adoption of mobile work force and BYOD, users should be taught to question, and validate every message or prompt received. Attackers will not waste an opportunity to send a user a link, message or an exploit to any device that the user can interact with and that the attacker can exploit. SMS was successfully used to deliver a jailbreak exploit in iOS 2 due to the messaging application running with root rights. Although SMS is an older protocol with known vulnerabilities Apple is using that protocol to initiate activation of its iMessage and FaceTime applications. Developing a secure protocol must have a solid secure foundation; SMS is not a secure foundation. Using it to initiate the iPhone Messaging activation introduces an entry point for attackers. Due to the iMessage's dependency on the SMS protocol for its activation process, an attacker has the potential to utilize SMS weaknesses to achieve a full iMessage compromise without the need to do SIM card swapping once the full iMessage protocol is understood. SMS is also the primary method many social and sharing applications use to validate the user's identity such as Viber, Tango and may other social applications with millions of users vulnerable to having their data hijacked if the activation SMS is intercepted

3.2. iMessage and FaceTime

The primary and successful attack discussed against iMessage and Face Time entails SIM swapping. A short-term solution is to educate "iDevice" users not to hand their device to unknown and untrusted individuals. Development of any long-term solutions will involve coordinated efforts between carriers and Apple. Simplicity versus security is always a challenge; however, a very simple attack can compromise iMessage and introduce data leaks. To eliminate the SIM card swap attack, Apple may want to consider removing the out of the box functionality and require authentication using an Apple ID associated with the subscriber's phone number during registration, reactivation upon SIM card removal or insertion and validate the users identity through each messaging transaction. This will eliminate the SIM swapping attack in its entirety. Disabling the iPhone if the SIM card is removed or Apple ID authentication fails offers an alternative to secure the iPhones, however this response will introduce significant negative user impact. During research, the author encountered a period where none of the phones would successfully activate iMessage for more than 24 hours, leading to the

assumption that that Apple has limits or other security measures in place to slow down an attacker attempting multiple SIM swaps within a short period.

iMessage is a great solution addressing SMS related vulnerabilities, however the closed protocol and Apple's attempt at security through obscurity has been demonstrated through the years to eventually fail. Apple may want to consider allowing the cryptography community to analyze the iMessage protocol and offer feedback to improve it. With iMessage providing user and sender authentication, Apple ID security becomes much more crucial. Apple has implemented some good defenses by offering two factor authentications; but it is not required by default to users. Apple has also implemented notification when a user logs into another iDevice to the rest of the signed on devices letting the user know that the account has signed on from device "ABC". Apple should consider steps to improve the notification to give the user a way to respond if it is an unauthorized login rather than a casual note leaving the user clueless on how to act. Using Find iPhone notification email to alert the owner to a SIM card removal or an unknown device association with the user apple ID would significantly increase the user awareness and provide help to the user.

4. Detection and Incident Handling

4.1. SMS

User education is the best line of defense; security practitioners should train the community that SMS messages are not different from email. SMS is used to deliver spam, advertisement and social engineering attacks just the same way as email. If possible, SMS should not be allowed on secure devices due to the transparency of man in the middle attack, where the user does not get any notification indicating an attacker captured their messages. It is not currently possible to disable the 2G protocol on standard or jail broken iPhones, therefore all devices supporting the 2G protocol could join an attackers base station using 2G permitting them to disable encryption and man in the middle all communications without alerting the user to any anomalies.

4.2. iMessage and FaceTime

User education is the primary method to detect the interception of iMessage and FaceTime communications. Apple ID username and password protection should be taught with emphasis on strong passwords and regular password changes. A user will notice several events if their iMessage is being intercepted, first any messages sent by the attacker will also be copied to the victim's phone. Users should be trained to be vigilant to spot, report and change Apple's ID password immediately if any messages appear in their history, which the victim did not send. If the attacker logs in using the victims ID, a notice will appear on the victim's phone indicating the account signed on another device. Immediate password change should be taught to all users if they see such as message. Security professionals should secure the SIM card to prevent its removal and blocking the SIM swapping attack for high security devices.

5. Conclusion

The overwhelming theme in security vulnerabilities seems to be legacy protocols, backward compatibility, and global device support related issues. Manufactures and developers invest significant resources into improving their product's functionality, features and security. Unfortunately, they seem to have a difficult time letting go of legacy designs. As a community, we should raise awareness and lobby for the retirement of insecure legacy systems. The largest attack vector highlighted in this study deals with the 2G protocol (which has been replaced by 3G, 3.5G and LTE protocols). After the deployment of three major replacement protocols, the 2G protocol is still supported by all modern cell phone and carriers. There has been no public announcement regarding the development of roadmap leading to the de-supporting of 2G standards and protocols.

Lack of authentication leads to potential exploits, as security professionals we fight exploits and attacks centered around DNS, ARP, IP and Email spoofing to name a few. Apple may want to consider enforcing username and password authentication prior to allowing phone number based iMessage and FaceTime services. With recent reports regarding NSA accessing continental fiber and having access to ISP records as well as high profile technology company data; it is possible that intelligence agencies also have

access either with or without a warrant to unencrypted cellular data. This access could grant the government similar access to an attacker with man in the middle access. SMS, iMessage activation SMS exchange and possibly redirecting the iMessage traffic to another device by intercepting the initial activation message. Authenticating iMessage solely using an email address provides additional authentication above the current iMessage activation SMS exchange and should be encouraged. Third party messaging solutions using single use logins could provide a future secure communication platform.

6. References

Apple.com. (2012, 10). *iOS Security*. Retrieved 07 13, 2013, from Apple.com: http://images.apple.com/iphone/business/docs/iOS_Security_Oct12.pdf

AppleInsider. (2013, 03 22). *New security hole allows for Apple ID password reset using Apple's iForgot page [u]*. Retrieved 07 13, 2013, from appleinsider.com: http://forums.appleinsider.com/t/156614/new-security-hole-allows-for-apple-id-password-reset-using-apples-iforgot-page-u

Bongiorni, L. (2012). *OpenBTS*. Retrieved 06 30, 2013, from University of Ostrava/SlideShare.net: http://www.slideshare.net/iazza/open-bts-emergency-gsm-messaging-monitoring-system-for-civil-protection

Burkholder, P. (2002, 02 01). *SSL Man-in-the-Middle Attacks*. Retrieved 07 13, 2013, from sans.org: http://www.sans.org/reading_room/whitepapers/threats/ssl-man-in-the-middle-attacks_480

Cecchetti, J. B. (2012, 05 18). *Attacking the Phone*. Retrieved 06 30, 2013, from Brown.edu: http://cs.brown.edu/people/jwsbrown/Attacking_the_Phone.pdf

Charlie Miller, Blazakis, D., Dai Zovi, D., Esser, S., Iozzo, V., & Weinmann, R.-P. (2012). *iOS Hacker's Handbook.* Wiley.

CloudMark. (2011, 06). *SMS Spam Guide*. Retrieved 06 30, 2013, from Cloudmark.com: http://www.cloudmark.com/releases/docs/sms_spam_guide.pdf

CloudMark-Blog. (2013). *CloudMark Blog*. Retrieved 06 30, 2013, from cloudmark.com: http://blog.cloudmark.com/category/sms-fraud/

cryptopath. (2010, 01 29). *iPhone PKI handling Flaws*. Retrieved 07 13, 2013, from http://cryptopath.wordpress.com/2010/01/29/iphone-certificate-flaws/: http://cryptopath.wordpress.com/2010/01/29/iphone-certificate-flaws/

Cutler, K.-M. (2013, 06 10). *Apple Has Sold 600M iOS Devices, But Android Is Not Impressed*. Retrieved 06 16, 2013, from techcrunsh.com: http://techcrunch.com/2013/06/10/apple-android-2/

Elcomsoft.com. (2013, 07 13). *Elcomsoft.com*. Retrieved 07 13, 2013, from Elcomsoft Phone Password Breaker: http://www.elcomsoft.com/eppb.html

Ericsson.com. (2012, 12 30). *Twenty years of Short Message Service (SMS): how text messaging helped to change the world*. Retrieved 06 18, 2013, from http://www.ericsson.com/news/121130-twenty-years-of-short-message-service_244159017_c

Gartner. (2008, 06 23). *Gartner Says More than 1 Billion PCs In Use Worldwide and Headed to 2 Billion Units by 2014*. Retrieved 06 16, 2013, from Gartner.com: http://www.gartner.com/newsroom/id/703807

Green, M. (2012, 08 18). *Dear Apple: Please set iMessage free - A Few Thoughts on Cryptographic Engineering*. Retrieved 07 13, 2013, from cryptographyengineering.com: http://blog.cryptographyengineering.com/2012/08/dear-apple-please-set-imessage-free.html

Hollington, J. (2012, 11 09). *The Complete Guide to Face Time + iMessage: Setup, Use and Troubleshooting*. Retrieved 06 30, 2013, from ilounge.com: http://www.ilounge.com/index.php/articles/comments/the-complete-guide-to-facetime-imessage-set-up-use-and-troubleshooting/

imfreedom.org. (2013, 07 13). *iMessage*. Retrieved 07 13, 2013, from imfreedom.org: http://imfreedom.org/wiki/IMessage

Kalinchuk, A. (2012, 08 18). *Apple: We Can't protect you from fake SMS messages, just use iMessage*. Retrieved 06 30, 2013, from digitaltrends.com: http://www.digitaltrends.com/mobile/apple-address-sms-vulnerability-by-touting-imessages-security/

McGee, A. (2012, 08 18). *Try Sendrawpdu App from Pod2G to Break iPhone SMS Security*. Retrieved 06 30, 2013, from www.letsunlockiphone.com: http://www.letsunlockiphone.com/iphone-sms-security-app-sendrawpdu/

meeee. (2013, 07 13). *Github - PushProxy*. Retrieved 07 13, 2013, from Github.com: https://github.com/meeee/pushproxy

Proffitt, T. (2012, 11 05). *Forensic Analysis on iOS Devices*. Retrieved 07 13, 2013, from sans.org: http://www.sans.org/reading_room/whitepapers/forensics/forensic-analysis-ios-devices_34092

Ritter, T., DePerry, D., & Rahimi , A. (2013, 07 13). *I Can Hear You Now: Traffic Interception and Remote Mobile Phone Cloning with a Compromised CDMA*

Femtocell. Retrieved 07 13, 2013, from blackhat.com: https://www.blackhat.com/us-13/briefings.html#Ritter

Ryu, M. (2012, 02 21). *The iMessage Flaw (aka the iMessage Bug) Detailed and Explained (mostly)*. Retrieved 07 13, 2013, from macryu.com: http://macryu.com/the-imessage-flaw-aka-the-imessage-bug-detailed-and-explained-mostly

Sara Radicati, P., & Hoang, P. A. (2012, 04). *Email Statistics Report, 2012-2016*. Retrieved 06 25, 2013, from Radicati Group LLC: http://www.radicati.com/wp/wp-content/uploads/2012/04/Email-Statistics-Report-2012-2016-Executive-Summary.pdf

Smith, C. (2013, 03 30). *Selected Apple iMessage users hit by DDoS attack, forcing iOS app crash*. Retrieved 07 13, 2013, from techradar.com: http://www.techradar.com/us/news/computing/apple/selected-apple-imessage-users-hit-by-ddos-attack-forcing-ios-app-crash-1141657

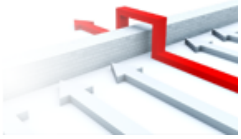
Spencer, G. (2011, 10 12). *iOS Messaging*. Retrieved 06 30, 2013, from macstories.net: http://www.macstories.net/stories/ios-5-imessage/

tompaman. (2012, 02 17). *How to send a message using iChat and AppleScript*. Retrieved 07 13, 2013, from stackoverflow.com: http://stackoverflow.com/questions/9325150/how-to-send-a-message-using-ichat-and-applescript

William Ench, P. T. (2005, 09 2). *Open Functionality in SMS-Capable Celluar Networks*. Retrieved 06 30, 2013, from smsanalysis.org: http://www.smsanalysis.org/

Last Updated: August 5th, 2020

Upcoming SANS Penetration Testing



Instructor-Led Training | Aug 10 ET	**, VA**	**Aug 10, 2020 - Aug 15, 2020**	**CyberCon**
SANS Reboot - NOVA 2020 - Live Online	**Arlington, VA**	**Aug 10, 2020 - Aug 15, 2020**	**CyberCon**
SANS Reboot - NOVA 2020	**Arlington, VA**	**Aug 10, 2020 - Aug 15, 2020**	**Live Event**
Live Online - SEC504: Hacker Tools, Techniques, Exploits, and Incident Handling	**, United Arab Emirates**	**Aug 11, 2020 - Aug 29, 2020**	**vLive**
Instructor-Led Training | Aug 17 MT	**, IL**	**Aug 17, 2020 - Aug 22, 2020**	**CyberCon**
SANS Summer Hack Europe 2020	**, United Arab Emirates**	**Aug 17, 2020 - Aug 28, 2020**	**CyberCon**
Cyber Defence APAC Live Online 2020	**, Singapore**	**Aug 17, 2020 - Aug 22, 2020**	**CyberCon**
SANS Essentials Live Online 2020	**, Australia**	**Aug 17, 2020 - Aug 22, 2020**	**CyberCon**
Instructor-Led Training | Aug 17 ET	**, DC**	**Aug 17, 2020 - Aug 22, 2020**	**CyberCon**
SANS Japan Bilingual Live Online	**, Japan**	**Aug 31, 2020 - Sep 05, 2020**	**CyberCon**
SANS London September 2020	**London, United Kingdom**	**Sep 07, 2020 - Sep 12, 2020**	**Live Event**
SANS London September 2020 - Live Online	**London, United Kingdom**	**Sep 07, 2020 - Sep 12, 2020**	**CyberCon**
SANS Philippines 2020	**Manila, Philippines**	**Sep 07, 2020 - Sep 19, 2020**	**Live Event**
SANS Baltimore Fall 2020 - Live Online	**Baltimore, MD**	**Sep 08, 2020 - Sep 13, 2020**	**CyberCon**
SANS Baltimore Fall 2020	**Baltimore, MD**	**Sep 08, 2020 - Sep 13, 2020**	**Live Event**
SANS Munich September 2020	**Munich, Germany**	**Sep 14, 2020 - Sep 19, 2020**	**Live Event**
SANS Munich September 2020 - Live Online	**Munich, Germany**	**Sep 14, 2020 - Sep 19, 2020**	**CyberCon**
SANS Network Security 2020	**Las Vegas, NV**	**Sep 20, 2020 - Sep 25, 2020**	**Live Event**
SANS Network Security 2020 - Live Online	**Las Vegas, NV**	**Sep 20, 2020 - Sep 25, 2020**	**CyberCon**
SANS Australia Spring 2020	**, Australia**	**Sep 21, 2020 - Oct 03, 2020**	**Live Event**
SANS Australia Spring 2020 - Live Online	**, Australia**	**Sep 21, 2020 - Oct 03, 2020**	**CyberCon**
SANS San Antonio Fall 2020 - Live Online	**San Antonio, TX**	**Sep 28, 2020 - Oct 03, 2020**	**CyberCon**
SANS Northern VA - Reston Fall 2020	**Reston, VA**	**Sep 28, 2020 - Oct 03, 2020**	**Live Event**
SANS San Antonio Fall 2020	**San Antonio, TX**	**Sep 28, 2020 - Oct 03, 2020**	**Live Event**
SANS Northern VA - Reston Fall 2020 - Live Online	**Reston, VA**	**Sep 28, 2020 - Oct 03, 2020**	**CyberCon**
Oil & Gas Cybersecurity Summit & Training 2020	**Virtual - US Central,**	**Oct 02, 2020 - Oct 10, 2020**	**CyberCon**
SANS Amsterdam October 2020	**Amsterdam, Netherlands**	**Oct 05, 2020 - Oct 10, 2020**	**Live Event**
SANS Tokyo Autumn 2020	**Tokyo, Japan**	**Oct 05, 2020 - Oct 17, 2020**	**CyberCon**
SANS Amsterdam October 2020 - Live Online	**Amsterdam, Netherlands**	**Oct 05, 2020 - Oct 10, 2020**	**CyberCon**
SANS October Singapore 2020 - Live Online	**Singapore, Singapore**	**Oct 12, 2020 - Oct 24, 2020**	**CyberCon**
SANS London October 2020 - Live Online	**London, United Kingdom**	**Oct 12, 2020 - Oct 17, 2020**	**CyberCon**