International Journal of Research in Advanced Engineering and Technology
ISSN: 2455-0876
Impact Factor: RJIF 5.44
www.newengineeringjournal.in
Volume 4; Issue 2; May 2018; Page No. 37-40

The security of iMessage and its Metadata

Grayden Tiner, James Anderson
Florida Polytechnic University, United States America

Abstract
With hundreds of millions of users using mobile phones in todays' corporate world, the risk for security breaches has never been greater. The data that phones can leave behind can be very damaging to a company if the mobile phone is used for work, with today's companies most IT teams allow a "Bring your own device" policy. iMessage is one of the leading mobile communications application, because of this hackers are increasingly trying to steal information transmitted through iMessage. After looking at a few hypothetical ways to hijack, spoofing data from iMessage, and even investigating logs and backups, the amount of data that is left behind is more shocking than users would believe. iOS backups can leave behind old messages, contacts, and pictures transmitted through iMessage, but users can avoid this by encrypting the backups. In all iMessage is safe enough for an average user to use without worrying about data being stolen.

Keywords: iMessage, vulnerabilities, and security

1. Introduction
iMessage is used by most iPhone users today, and is considered to be one of the most feature rich application suites available on the mobile platform. Its integration with the operating system makes it an obvious favorite of the iPhone user base, but this does not necessarily mean that the iMessage suite is the most secure service available. The purpose of our paper is to evaluate the security of the iMessage suite to determine just how secure the software is, and to determine where the weak points may lie.

Given that iMessage is based off of Apple's proprietary software, it is very difficult to fully determine the security of the information being transmitted across its service. With that said, several papers have been released which have attempted to look at the information transmitted by the service and make a rough estimation of their security based on the technologies they are based on. Whether it be Face Time, iMessage, of any of the other various subcategories of iMessage, all of these applications use some form on encryption for storing and transmission of data. The introduction of iMessage was in 2011 when Apple was updating their iOS to iOS 5. With this update, two new servers were created for the iMessage protocols, Apple Push server and Apple ESS server, courier.push.com and ess.apple.com, respectively. The service was somewhat insecure until iOS6, when many changes were made to improve this aspect, and since then only a handful of minor changes have been made.

2. Related work
This work was inspired by the curiosity of the team as members of the team use Apple products and wanted to investigate the security and digital forensics of Apple's most heavily used applications. Apple has stated that the communication that is taking place over iMessage or over FaceTime is secured with the use of end-to-end encryption. Hence, only the sender and receiver can see it or read it. Even the Apple Company cannot decrypt those messages. Apple also stated that they do not keep data items related to the location of the customers, any map searches or any requests made by Siri [3]. The stack overflow [1] talks very briefly about the encryption of iMessage as well as the privacy statement given by Apple. The iMessage wiki [5] goes into great detail describing how the iMessage protocol works, as well as what servers the messages are sent to and what data is brought with it.

The sources above gave the team insight into how iMessage functions, as well as where data is being stored when packets are being transmitted from server to device. The presentation (Quicklab 2013) gave the team a lot insight into how Apple's servers communicate with devices, the presentation also covered many topics and ways on how to perform man-in-the-middle attacks, the research paper (SANS Institute 2013) went into great detail how to perform physical attacks on the iDevices; one attacked included that attacker obtaining the SIM card of the phone and loading it into their own phone and registering their phone with the iMessage credentials and was able to spoof messages to any of the contacts on the victim's phone.

Both of these papers gave great insight into multiple attacks into devices, but the goal of this research was to investigate this and what other data iMessage leaves behind. The papers lead the team with great starting points and allowed the team to further improve upon what has already been discovered.

This service is known as Apple Push Notification Service (APNS), and it uses various information, that is stored on Apple's servers, from the user such as; push-token and a push-certificate, with these things a user with a Apple ID is able to receive other messages that contain information from other Apple ID users, such as their Apple ID and which devices are connected through that Apple ID; but there is a problem, the

push and iMessage servers do not implement certificate pinning. So with the lack of certificate pinning, a hacker can gain knowledge of another user's Apple ID information, such as, as user's Apple ID password. Sadly though this is not that complicated to obtain, for a hacker to add a signing certificate to the device all they must do is "connect the device to machine running iPhone Configuration utility" [4]. Once they have the certificate the hacker is able to access Apple's push and iMessage servers.

To understand how a device is vulnerable through APNS, one must first look at how the device communicates with the service. The device will first start TLS initiation with the APNS, then APNS sends back a server certificate toe the device, the device then validates the server certificate before sending its' certificate, then finally once the APNS validates the device's certificate TLS is established. One of the papers goes extensively into how to commit man-in-the-middle attacks using spoof certificates from the ESS server to bypass the Push servers. The encryption used by iMessage is a RSA cipher text with 1280 bits which is on top of an AES session key and AES-CTR cipher text [4], see appendix F. If an attacker were to try to decrypt the traffic it would impossible as even Apple they have stated, "they cannot decrypt the messages" [3].

3. Problem Statement

How secure is iMessage? With over hundreds of millions of users, the applications need to be secure and any information that is sent between users should not be obtainable. Apple themselves have stated that their main concern is privacy of their customers Apple has stated that the communication that is taking place over iMessage or over FaceTime is secured with the use of end-to-end encryption. Hence, only the sender and receiver can see it or read it. Even the Apple Company cannot decrypt those messages. Apple also stated that they do not keep data items related to the location of the customers, any map searches or any requests made by Siri [3]. While this may be Apple's goal, if the encryption is weak or is faulty then user's data is easily available to a hacker and can be stolen.

In addition, there is the problem of the automatic data backup system employed by iTunes whenever an iDevice is connected. While this functionality is very user-friendly, it is important to ensure that this data is safe, since it includes many potentially sensitive data sources. In addition, the iTunes backup can be done on iCloud or locally. These are fundamentally different approaches which have different strengths, weaknesses and potential vulnerabilities. While issues with iCloud have become big news in recent years, issues with backup security can be potentially just as dangerous.

Both of these systems have been shown to be vulnerable to social engineering attacks, and many of the easiest attacks on these services rely on these attacks. More specifically, physical possession of the device enables SIM-switching attacks to compromise iMessage communications, and stolen iCloud passwords have caused problems for many users. Unfortunately, preventing these attacks involves user education, and this can be a very difficult thing to accomplish in the consumer market. However, there are also some concerns with the technical aspects of these services.

Specifically, there is reason to believe that the iMessage service may be vulnerable to Man in the Middle attacks. In addition, while iCloud has been updated to help prevent passwords from being phished, their encryption for the local backup system has been shown to be breakable, and the option to leave these backups unencrypted opens the way for these backups to be exploited for information. These two problems will be what we attempt to discuss in the remainder of this paper.

4. Technical Approach

To start one must understand a few things about Apple and iMessage. First, each user has a unique Apple ID. This ID is linked with all Apple devices the user may have, such as a phone, laptop, iMac, iPad, or any other device with iMessage capabilities. With how Apple handles linking of multiple devices to push data to certain information is shared among devices, when one user contacts another user and shares their Apple IDs' both parties gain access to all of the other parties devices.

For example, in the log files for Face Time, if a user calls another user through the application the Apple servers will bring back various information regarding that user such as: Apple ID, email address used to sign up, and associated phone number (although this is sometimes absent, if a cellular connected device has not yet been added to the account.) iMessage is not a fool proof application; the protocols used for encryption are secure only if the messages are being sent to the correct receiver. Apple uses a Push Client to send messages and other requests to a user's phone.

Despite remaining largely the same for some time, iMessage is considered to be a very secure service, although some security concerns remain. The servers behind this service handle all requests sent to an Apple ID, and use a variety of ports and services to accomplish their goals. The push server uses TLS port 5223 which handles all of the traffic such as iMessage, Face Time, Game Center, etc. while the ESS server handles authentication and key repository for other Apple IDs'.

This is a hypothetical way one would execute a man in the middle attack on some user. First the hacker would need a few things: a great network, the private key's of the victims' devices, the victim's Push and ESS certificates, and the victim's RSA and ECDSA private keys. Once the hacker has all of these pieces of information the hacker is then able to proxy all requests sent to the ESS server of Apple to alter all of the public keys with new keys. The hacker will also proxy all communication between the Push server, this allows the hacker to eavesdrop on all messages sent to the victim and modify them [4]. This is the basic idea for each man-in-the-middle attack, the hacker uses their network for proxy the ESS and Push servers to spoof requests and messages and then the hacker uses the certificate and keys to eavesdrop and forge messages. One problem with each of these hypotheticals is all of the requirements that are needed to properly perform these attacks. For one sided man-in-the-middle, the hacker needs both certificates as well as the keys, which are extremely difficult to procure, even if the attacker were to try two-sided man-in-the-middle they wouldn't need the keys but would

need a very powerful network to handle the network traffic.

Moving forward, our team investigated the files on a Mac computer to see if there was any data that could've been obtained if a system was compromised. Once a user shows the hidden folders on a system that has Face Time, the user is then able to view the Face Time log files. This only shows information related to that call, such as: who was being called, what type of call it was, and some other information regarding the call. Some information is not encrypted, such as plaintext, or in hexadecimal. This was the extent of the information we could find about the iMessage information as it was used in real time, and given that log files are often overwritten or deleted, it provided our first indication of where some digital forensics work can start.

Next we looked at the security implemented on the local backup. As previously stated, iTunes automatically syncs iDevices whenever they are connected to iTunes on Windows or Mac. When iTunes is first started, an option is provided if backups should be done locally or on Apple's cloud service, iCloud. In addition, if the backup is done locally (which can be common, given that iCloud storage is limited for users who do not pay for the premium service,) an option is provided to determine if the user would like to encrypt the backup.

Because the key for encryption is user-defined the backup is vulnerable to dictionary attacks, brute force attacks, and other related attacks. In addition, as previously stated, the encryption used on these backup files has been shown to be (to some extent) crackable. Given that this security is somewhat vulnerable, we decided to focus on the information in question. In the backup directory are hundreds of SQLite files, many of which are encrypted by Apple themselves. Located inside the directory are two files of particular interest, 3d0d7e5fb2ce288813306e4d4636395e047a3d28 and 31bb7ba8914766d4ba40d6dfb6113c8b614be442. To view the information located inside these extensionless files, we utilized a generic database browsing program.

The inside the database consisted of several tables. In our efforts we focused mainly on the message and handle tables from the 3d0d7e5fb2ce288813306e4d4636395e047a3d28 file and one table, AD Person Full Text Search content, from 31bb7ba8914766d4ba40d6dfb6113c8b614be442. In the message table there are the details of every SMS and iMessage communication that was saved on the iDevice, with a handle id number. This handle id number is specific to each conversation, and when the number is located in the handle table, it corresponds to the participant's phone number. Moving to AD Person Full Text Search content, users are able to correlate the phone number to any information located in the phone's contacts. This may include their name, address and other personal information. In addition to this data, the backup stored many other potentially sensitive pieces of data in a similar way, including such information as calendar information, reminders, notes, call history and even location data [f] [g] [h].

5. Results and Analysis

Since Apple uses an end-to-end encryption on devices when they send messages or Face Time requests, one must perform extensive setup to even begin one man in the middle attack to receive and forge communications. This means that this medium is, by most technical measures, very difficult to intercept. In fact, during our research for this project, we found sources claiming that even Edward Snowden used iMessage during his activities. Regardless of the political implications that statement may have, it still exemplifies that those in the field feel quite comfortable using the service as a way to discreetly transmit data.

When it comes to users' backup data, if a laptop or desktop were compromised then the hacker would have access to multiple log and backup files. These files may be unencrypted, at which point most data can be obtained without much effort [1, 2]. If the backup file is encrypted the data is much more secure, although still not entirely safe. This is particularly true when it comes to organizations who have a greater level of technical prowess. To remedy this issue, users should treat backup data like they treat any other sensitive data; it should be safeguarded as much as possible.

6. Conclusion

All things considered, iMessage is more than capable for protecting the user of an average user and needs to be hacked by a seasoned hacker for any Man-in-the-Middle attacks. While spoofing and forging attacks are possible, this is mostly preventable through user education, although alternative articles have suggested some good ways of preventing this as well. Still, nothing within the iMessage suite itself seems much more vulnerable than any of the alternative services that exist, making it seem like a good option overall. In terms of the data that is stored on local backup, while it is variably safe (depending on the settings selected) it should still be handled with extreme care and users should be aware of where files are stored. Taking some steps to ensure that this data is safe from intruders in the case of an attack is also quite vital in ensuring a small data breach does not become a case of identity theft.

7. References

1. https://security.stackexchange.com/questions/123693/how-secure-is-facetime-by-apple
2. https://www.sans.org/reading-room/whitepapers/pda/sms-imessage-facetime-security-34325
3. https://www.apple.com/apples-commitment-to-customer-privacy/
4. https://blog.quarkslab.com/resources/2013-10-17_imessage-privacy/slides/iMessage_privacy.pdf
5. http://imfreedom.org/wiki/IMessage
6. https://stackoverflow.com/questions/1498342/how-to-decrypt-an-encrypted-apple-itunes-iphone-backup
7. https://www.iphonebackupextractor.com/blog/iphone-backup-location-all-files-extension/
8. Xiaojiang DU, Gagneja KK, Nygard K. Enhanced routing in Heterogeneous Sensor Networks, IEEE Computation World. 2009; 569-574, Athens, Greece, 15-20.
9. Evanoff Lauren, Nicole Hatch, Gagneja KK. Home Network Security: Beginner vs Advanced, ICWN, Las Vegas, USA, 2015, 27-30.
10. Gagneja KK, Nygard K. Heuristic Clustering with Secured Routing in Heterogeneous Sensor Networks, IEEE SECON, New Orleans, USA. 2013, 9-16, 24-26, 20.
11. Gagneja KK. Knowing the Ransomware and Building

Defense Against it Specific to HealthCare Institutes, IEEE MobiSecServ, Miami, USA, 2017, 11-12.
12. Gagneja KK. Secure Communication Scheme for Wireless Sensor Networks to maintain Anonymity, IEEE ICNC, Anaheim, California, USA, 2015, 16-19.
13. Gagneja KK. Pairwise Post Deployment Key Management Scheme for Heterogeneous Sensor Networks, 13th IEEE Wo WMo M, San Francisco, California, USA, 2012, 1-2, 25-28.
14. Gagneja KK. Global Perspective of Security Breaches in Facebook FECS, Las Vegas, USA, 2014, 21-24.
15. Gagneja K, James L. Computational Security and the Economics of Password Hacking, Future Network Systems and Security. FNSS. Communications in Computer and Information Science. Springer, 2017, 759.
16. Javier Campos, Slater Colteryahn, Gagneja Kanwal, Pv6 transmission over BLE Using Raspberry PI 3, International Conference on Computing, Networking and Communications, Wireless Networks (ICNC'18 WN), Maui, USA, 2017-2018, 759.
17. Kanwal G. Pairwise Key Distribution Scheme for Two-Tier Sensor Networks, IEEE ICNC, Honolulu, Hawaii, USA, 2014, 1081-1086, 3-6.
18. Hill K. Gagneja K. Concept network design for a young Mars science station and Trans-planetary communication, Fourth International Conference on Mobile and Secure Services (MobiSecServ), Miami Beach, FL, 2018, 1-8.
19. Luis James, Gagneja KK. Mustapha Akbbas, Idalldes Vergara Laurens, Future Stress, Forecasting Physiological Signals, IEEE CCWC, Las Vegas, USA, 2017, 9-10.
20. Nygard K, Gagneja K. Energy Efficient Approach with Integrated Key Management Scheme for Wireless Sensor Networks, ACM MOBIHOC, Bangalore, India. 2013; 2:13-18.
21. Nygard K, Gagneja KK. A QoS based Heuristics for Clustering in Two-Tier Sensor Networks, IEEE FedCSIS 2012, Wroclaw, Poland, pages 779-784, Sept. 9-12, 2012.
22. Nygard K., Gagneja K.K., Tabu-Voronoi Clustering Heuristics with Key Management Scheme for Heterogeneous Sensor Networks, IEEE ICUFN, Phuket, Thailand, 2012; 46-51, 4-6.
23. Nygard K, Gagneja KK. Key Management Scheme for Routing in Clustered Heterogeneous Sensor Networks, IEEE NTMS, Security Track, Istanbul, Turkey. 2012; 1-5, 7-10.
24. Runia Max, Gagneja KK. Raspberry Pi Webserver, ESA, Las Vegas, USA. 2015, 27-30.
25. Singh Arvinderpal, Gagneja KK. Incident Response through Behavioral Science: An Industrial approach, IEEE CSCI, Las Vegas, USA, 2012, 7-9.
26. Singh Arvinderpal Gagneja KK. Mobile Health (mHealth) Technologies, IEEE HealthCom, Boston, USA. 2015, 14-17.
27. Gagneja KK, Ranganathan P, Boughosn S, Loree P, Nygard K. Limiting Transmit Power of Antennas in Heterogeneous Sensor Networks, IEEE EIT2012, IUPUI Indianapolis, IN, USA, 2014, 1-4, 6-8.
28. Nygard K, Bender L, Walia G, Kong J, Gagneja K, LeNoue M. Collaboration Using Social Networks for Team Projects, International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS'11), Las Vegas, USA, 2012, 18-21.