Review of IEEE Journal - Data Exfiltration From Internet of Things Devices: iOS Devices as Case Studies

Brandon Hardy (BH00125, URN: 6314964), Bsc Computer Science, University of Surrey 2017

Abstract—**Data safety has increasingly become a concern in the world of IoT devices. Sensitive and commercial-in-confidence 'big' data is becoming increasingly accessible on devices such as our mobile phones, as a result any vulnerabilities in IoT devices may put consumer data at risk. In this paper, I will be exploring research that exposes such loop holes in IoT devices demonstrating the types of data exfiltration that could take place.**

Keywords*—*Internet of Things (IoT), Data Exfiltration, iOS Data Exfiltration, Big Data Security, iOS jailbreaking

I. INTRODUCTION

In this report, I will be analysing and discussing the said journal relating to Data Exfiltration from IoT devices and discussing their work in Data Exfiltration. I will also be sharing my view on the approach and the experience learned from related work mentioned in the journal and further related work I have explored.

Data Exfiltration is the process of gaining unauthorised access to vulnerable systems and leakage of private information [1]. Internet of Things devices are becoming increasingly rich in big data (e.g. Personal and sensitive information of an individual, company or organisation). An example that was briefly touched upon in the journal mentioned the increasingly popular bring-your-own-device policy in the workplace. Such policy could expose a higher risk to Data Exfiltration to retrieve sensitive company / organisation information; if such devices are compromised.

The security of big data stored in IoT devices can be exploited from various layers of these devices, such as the hardware, software and network layer to covertly exfiltrate data stored in the vulnerable device; which otherwise may not be immediately obvious to the user. The journal that I will be exploring in the report [1], demonstrates Data Exfiltration from an iOS device using several vulnerabilities that can be exploited to retrieve 'big-data'.

The journal explores various weaknesses in iOS devices such as iOS's pairing mode which allows establishment of a trusted relationship which is used to determine whether access to an iOS device is granted from a computer. The journal explores how data can be covertly exfiltrated from a previous trusted third party device after using the pairing mode. Various other techniques such as using Apples special service, called Apple file conduct, can be used to 'jailbreak' devices. This brings an interesting link to the further work relating to the journal, a report by Palo Alto Networks [2], as they indicate in the report that jailbroken devices have been target to malware through exploitation's of vulnerabilities that open up when a device is jailbroken; as a result, creating a severe risk to the safety and security of 'big-data' on the device.

The latter contents of this report will explore the demonstrations of attacks and Data Exfiltration presented in the journal; and bring discussion to the advantages and shortcomings of the work carried out along with my view on the approach of the work.

II. ANALYSIS OF WORK

The data exfiltration attack as a summary, uses a model based on a client-server TCP/IP architecture. This model creates the ability for 'covert' data exfiltration from an iOS device over a USB that is connected to a trusted computer. The client application has to reside on the target computers to interact with the connected iOS devices. The server application can be ran on a remote computer; this is controlled by the attacker.

The client application has the ability to "accept connections from multiple devices connected to the same computer". The remote server application that is controlled by the attacker, handles connection requests from multiple computers and allows for data exfiltration from several devices simultaneously. This is an improvement to related data exfiltration attempts [3], [4] mentioned in the journal as they only allow one-to-one connections.

Once a device is connected the client application is notified and the 'attacker' can start requesting device details. Furthermore, iOS Device Information and media file lists are included. The data exfiltration attack can be attempted once the device is paired with a computer running the client application. The attack makes use of the client application to actively listen to incoming connections. The client application is designed to run as a background process on the target computer.

Securing the communication between the client and server applications make use of RSA public-key cryptography to exchange a session key which is used in a AES-128 symmetric cipher. The Session key is needed for the exchange of device details and used by the applications to encrypt and decrypt exfiltrated data.

In the journal demonstration, the client application exfiltrates data via iTunes libraries. This involves the use of 'iTunesMobileDevice.dll', an existing library distributed with iTunes for Windows.

Similar to iOS Surveillance and Mobile Remote Access Trojans (mRATs) and mentioned in the following white-paper [5] attackers carry out attacks through jail-broken devices. In the process of jailbreaking a device all built-in iOS security mechanisms are removed, this gives the attackers the ability to install mRAT software which has the potential to 'spy on' everything stored and 'flowing through' the device; such as texts, calls and other potentially sensitive and personal information that is sent and retrieved from the device.

Related work in the study of malware in iOS devices [6] found that malware does not just attack jail-broken devices, but infections can be caused by vulnerabilities in developer certificates and cause malicious applications to be on the official app store; not just unofficial app stores like Cydia. The study showed that 30.5% of malware affects non-jailbroken devices where as the remaining being jail-broken devices. This clearly shows that having a jail-broken device significantly increases the risk of malware on a device, moreover, having a non-jailbroken device does not prevent malware from infecting your device either.

A. Attacks

In order to communicate with iOS services, the client application imports a 'set of methods' that is included in the iTunes library 'MobileDevice.dll'. Furthermore, in order to subscribe to the notification of when a device is connected to a target computer, 'AMDeviceNotificationSubscribe' is called from the client application. There are a range of methods that enable the attacker to query the targeted device. Methods such as 'AMDeviceConnect, AMDeviceStartSession, AMDeviceStartService' and a couple more which are touched upon further in the journal; are called to establish a connection. Once the connection is established, methods are used to enable the attacker to query and request details from the target device.

Attacks are separated into 'Attack I, Attack II and Attack III'. I will be analysing each attack and giving my view on the approach that the researchers took to each attack. The configuration of the attacks were designed in such a way to demonstrate the server applications ability to handle connections simultaneously and different connections would not impact the data exfiltration outcome.

B. Attack I

Attack one makes use of a *jailbroken device's file system*. To request device details, the client application calls the StartService method mentioned previously to start the 'AFC2 Service'. If successfully initiated, the target device is indicated as jailbroken; the attacker can then access the entire file system of the targeted device. Additionally, the databases that contain sensitive data such as address book contents, emails, texts (SMS) etc. can also be read and exfiltrated. This is done by the attacker sending the exfiltration request to the client. Encryption is carried out using the session AES key and the data exfiltrated is then decrypted on the attackers computer using the same key.

This is a proven highly effective method, demonstrating the ease at which an attacker can exfiltrate sensitive information from a device that has been jailbroken. Posing a risk to the victims data completely covertly, however, note that this attack is done via USB connection. Moreover, the device does not have to be unlocked in order for the attacker to exfiltrate media files and databases with the exception of the email database; it is necessary for the device to be unlocked to retrieve the email database.

C. Attack II

Attack II focuses on a device with a *non-jailbroken file system and no backup password set*. As of before, the client application calls the StartService method. If AFC2 cannot be initiated but the client successfully initiates AFC the file system can be determined to be non-jailbroken. As AFC is not initiated, access to read databases such as emails is not possible. Media files can be exfiltrated by using the same method used in Attack I via AFC.

Data can be exfiltrated from databases stored on the device, as the lack of a backup password makes the backup process possible to exploit. A new backup is made and stored in a temporary folder, which is then exfiltrated. This attack is ran in the background and therefore, this attack is covert, the likeliness of a user knowing is very low. Once a backup of the needed databases is complete the Apple process is killed and the backup folder is deleted to reduce detection.

D. Attack III

Once again, the attack used a device with a *non-jailbroken file system*, however there was a backup password set. In order for the attacker to exfiltrate important databases from the device, decryption had to be carried out using the AES-128 key that was used to encrypt the connection between client and server.

In order to discover the backup password, the attacker would have to carry out a brute force attack, or alternatively using a offline dictionary attack. Otherwise, important databases cannot be accessed. Media Files are at risk however, through AFC used in Attack II.

As demonstrated, the setup of a ***strong*** backup password is certainly recommended to protect sensitive data from attacks. Ensuring that the password is strong will delay brute-force and offline dictionary attacks considerably.

III. Discussion

In this paper, I explored a journal that demonstrated Data Exfiltration using iOS Devices. Data Exfiltration is certainly not a new problem for IoT devices. The security of these devices is becoming a largely discussed problem, which is drawing an increasing amount of interest from security experts and researchers.

The approaches taken in the journal were designed to demonstrate that features that are designed to minimise impact of data privacy in a case where the device is misplaced, can be exploited and used to assist in exfiltrating media files and sensitive information from databases. In all three of the attacks, media files were able to be exfiltrated from the device. Even the device that was not jail-broken and had a backup password set; expressing the risk to sensitive information and files even when appropriate steps are taken to reduce the risk of stolen data.

The journal made two main recommendations / suggestions to reduce the risk of Data Exfiltration. Their first recommendation was advice to Device Manufacturers. They suggested that restriction limited to paired computers should be scrapped and introduce an OAuth-like approach to register a client software to access device contents. Client software would have to be registered in order to obtain a unique token in return, which would then be authorised through iCloud to verify if the token is still valid or not. Thus when a user authorises a client to interact with their device, a record will be stored in the device keychain which, will be regularly checked on the configuration settings when an internet connection is available.

This recommendation is similar to a suggestion made in a separate journal [7] 'Device Authentication'. Elias Tabane and Tranos Zuva suggested that whenever an IoT device is plugged into a network system, it must authenticate itself before it can receive any form of data transmission, similar to the suggestion of OAuth-like authentication for iOS, where it must be authenticated before a client application can retrieve data.

The second recommendation made in the journal was aimed at device users. The recommendation was to practice 'security hygiene'. Such as not installing applications that have an unknown origin. As demonstrated, jail-broken devices can be exploited by applications running on a 'trusted computer'. The use of a strong backup password is another recommendation made in [1]. Creating a backup password can greatly hinder any brute force attacks on the backup password. Without a backup password, as demonstrated, it is easy to exfiltrate sensitive databases.

Tied to security hygiene is a recommendation made in a related work [7], in this paper the authors recommended updates and patches are carried out regularly, as iOS devices can be updated easily, their caution of doing so in a manner that does not impair the functional safety is not relevant, however.

Raising awareness is an important aspect to the journal. Demonstrating attacks and vulnerabilities that are in IoT devices that the majority of people use; is paramount in ensuring that IoT can become a safer, more secure area by raising the awareness that such attacks can happen. This is a clear advantage of the journal being published, the simplicity of the studied work can lead to a greater understanding of attacks, possibly even to those who may not be from a technological background.

It can be argued, however, that the shortcoming of the journal work involves devices and iOS versions that are now out of date, and newer versions and devices have been released since. It may be argued that the security vulnerabilities that are demonstrated in the journal have been fixed since the journal release, and therefore classing the work as outdated. However, as argued in the journal, the principle of the paper is to create a better understanding of the capabilities of an attacker if a device is connected to an otherwise 'trusted' computer. Not only in the context of iOS devices, but as a whole, recommendations in the journal and this report give insights in how to reduce the risk of Data Exfiltration from IoT devices.

Through researching and investigating the topic of the journal, I learnt a fair amount surrounding the topic of Data Exfiltration. Prior to reading the journal, I did not have an idea that non-jailbroken phones can be compromised with such ease, even with no software altering involved such as jailbreaking. The journal demonstrated the exfiltration of media files and sensitive databases on a device that was not jailbroken, using methods and built in features created by Apple. It has opened my eyes to ensure that my security hygiene is a kept up to avoid any data being lost / stolen.

IV. Conclusion

Security in IoT devices continues to draw attention of academics and researchers across the globe [7]. IoT devices such as mobile phones are seen as havens for data, thus the naivety of safety of such data can cause big problems for users and organisations. The research investigated in this report raises important questions about IoT Security and the resilience of software on devices, even those of larger companies like Apple.

The attacks demonstrated in the journal discussed and related works [3], [4], [6] have demonstrated that tampering software or file systems can cause vulnerabilities to sensitive data. However, it has also shown that data can be vulnerable despite not tampering with software on the device. As a result, recommendations were given to not only device manufacturers with suggestions on how to make sensitive data safer, but to users of devices to practice security hygiene.

As a result of investigating such work and related research around the subject of IoT security, it would be helpful to focus future work on how to appropriately address the complex issues surrounding the security of IoT.

References

[1] C. D'Orazio, K. Choo and L. Yang, "Data Exfiltration From Internet of Things Devices: iOS Devices as Case Studies", IEEE Internet of Things Journal, vol. 4, no. 2, pp. 524-535, 2017.

[2] C. Xiao, Wirelurker: A New Era in iOS and OS X Malware, Palo Alto Netw., Santa Clara, CA, USA, 2014.

[3] Q. Do, B. Martini, and K.-K. R. Choo, Exfiltrating data from Android devices, Comput. Security, vol. 48, pp. 7491, Feb. 2015.

[4] S. J. OMalley and K.-K. R. Choo, Bridging the air gap: Inaudible data exfiltration by insiders, in Proc. Americas Conf. Inf. Syst., Savannah, GA, USA, 2014, pp. 112.

[5] Threats to iOS Mobile Devices. Lacoon Security Inc., 2017. [Online]. Available: https://idency.com/wp-content/uploads/2014/08/Lacoon-White-Paper-iOS-Threats.pdf. [Accessed: 11- Nov- 2017].

[6] L. Garcia and R. Rodriguez, "A Peek under the Hood of iOS Malware", 2016 11th International Conference on Availability, Reliability and Security (ARES), 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7784623/. [Accessed: 11- Nov- 2017].

[7] E. Tabane and T. Zuva, "Is there a room for security and privacy in IoT? - IEEE Conference Publication", Ieeexplore.ieee.org, 2016. [Online]. Available: http://ieeexplore.ieee.org/document/8073758/. [Accessed: 18- Nov- 2017].