Privacy of Network Traffic in Anti-Theft Software

Peter Lee

Mentor: Ming Chow

December 14, 2016

Anti-theft applications for mobile devices such as smartphones and laptops are becoming more prevalent among consumers as they recognize that the portability of their devices comes with the added risk of theft and accidental loss. Many of these applications, such as Lookout, Cerberus, Prey, and Android Lost, operate by running in the background of a device and constantly sending information to an external server. Once a device is marked as lost, these anti-theft applications have the capability of tracking the location of the device via GPS, providing hardware information and MAC addresses, take images using the device's built-in camera, and so on. Since these applications are frequently communicating with an external server in the background, it is important to ensure that any information that is sent cannot be intercepted and exploited by a third party. Furthermore, it is important to evaluate how much privacy one can expect from the application providing the service, and consider the benefits and possible risks that may result in using anti-theft software.

Introduction

As mobile devices such as smartphones, laptops, and tablet computers are increasingly ubiquitous, a growing anti-theft sector of software has developed to combat the increased likelihood of device loss and theft. Almost all of these software applications provide functionality that can locate or identify a lost device by providing information such as GPS coordinates, hardware information, webcam images, and screenshots. There have been multiple real world instances where anti-theft software has resulted in recovery of a lost or stolen device. For example, in 2011, a man in Oakland, California recovered his laptop through the use of screenshots and webcam photos taken by the anti-theft software "Hidden" on his stolen device [1].

Information from lost devices is usually communicated from the device through intermittent contact with an external server that keeps track of the data. Because of the sensitive nature of the communication between the server and the device, as well as the constant communication between the device and the server, it is important that this network traffic is secure and private.

There are several possible security concerns that stem from this type of communication: man-in-the-middle attacks via network traffic interception, eavesdropping, privacy concerning data storage by a company, and other exploits that can lead to compromises of some nature. Much of this paper discusses the relevancy of anti-theft software for the typical user, notes common security policies for anti-theft software, and takes an empirical look at security of the open-source anti-theft software Prey in relation to the security concerns noted above.

To the Community

In 2015, approximately 68% of United States adults owned a smartphone, about 70% owned a laptop, and 45% owned a tablet computer [2]. This demonstrates that anti-theft software is a concept that has relevance for a large portion of the population, as many people are likely to own some type of mobile device. With that said, those that already use anti-theft software in their mobile devices, any vulnerabilities or lapses in privacy clearly have much more direct relevance to these people than those who do not. However, many people may unknowingly be in use some form of anti-theft software. For instance, Samsung has an application that is preinstalled on its smartphones named "Find My Mobile" that provides functionality such as device location and remote wiping [3]. One case in 2014 illustrates a vulnerability found in theft protection software built into the BIOS of devices. By performing a man-in-the-middle attack on "Absolute Computrace", the anti-theft software built into the BIOS of many computers, attackers could execute arbitrary remote code on the device with full system access [4][5]. This vulnerability provides a concrete case in which an exploit in an anti-theft software resulted in potential significant compromise of a large number of devices, regardless of whether the owners had actively desired anti-theft software or not.

Overview of Security Policies of Anti-theft Software Vendors

Most anti-theft software provides claims of encryption, anonymity, and privacy in some form in their privacy policies and overall security policies. Some anti-theft software provides options for lost device information being sent to independent, client-chosen servers rather than servers provided by the anti-theft software vendor. Notably, some anti-theft software vendors

collect personally identifiable information for analytics and responsive advertising, as outlined in the Privacy Policy for Cerberus, an anti-theft software [6]. For the purposes of this paper, the policies of Prey will be scrutinized, in relation to encryption of network traffic, the possibility of man-in-the-middle attacks, and exposure of information. Prey touts network traffic encryption, and states that packet content is encrypted to prevent man-in-the-middle attacks, which should result in a secure application.

Code and Network Traffic Analysis

Source code for various versions of the client for Prey is available on Github, which allows static analysis of the code [7]. Static analysis of the Node.js code for Prey using Veracode mostly corroborates the integrity of the security of Prey, as the Veracode report gives a score of 91. While there were several medium vulnerabilities in the static analysis report, several of them appear to clearly be false positives. For instance, the report details hard-coded credentials as a security vulnerability, but names test files that contain dummy usernames and passwords that are fillers for actual credentials. Several other vulnerabilities also detail false positives in what seem to be test files unused in production. Some of the vulnerabilities detailed by the report may warrant further inspection, such as the "External Control of File Name or Path CWE ID 73" vulnerability, or "Cryptographic Issues", as these could be valid, niche vulnerabilities.

The network traffic of Prey was also observed using Wireshark when Prey was transmitting lost device reports as well as when Prey was in an inactivated state. Packet captures were saved using Wireshark with no other network applications open besides Prey and background services. The results are not particularly revealing, as all of the packets that are

connected to Prey are all encrypted using SSL/TLS. The client and the server seem to frequently perform the SSL/TLS handshake. Due to new session keys thus being created frequently, this is likely to help the security of using Prey. Overall, the network traffic produced by Prey seems to empirically be encrypted safely. However, there are two possible avenues of attack for an attack on Prey that the captured packets reveal, both involving some form of man-in-the-middle attack. One method is to intercept the network traffic of Prey, and complete the SSL/TLS handshake with the victim while redirecting the victim's network traffic, which allows an attacker to decrypt the packets sent by the victim. Another method that may facilitate this is some form of DNS spoofing or masquerading as a DNS server to provide a false IP address for "solid.preyproject.com", for which the Prey client frequently sends DNS queries for. The victim could thus potentially be directed to connect to an attacker's server that again, masquerades as an official server. Finally, one point of note is that observing network traffic reveals that someone is using Prey, due to the multiple DNS requests, which might provide a potential attacker more information.

Action Items

Based on the security policies and the observed experimental case, most anti-theft software is likely fairly secure at the moment, and is based mostly on the security of standards such as SSL and TLS. In some cases, this might result in misplaced faith in security standards, and in fact Prey was vulnerable due to the Heartbleed Bug because of its use of SSL [7]. Nevertheless, the network traffic that is produced by most anti-theft software is encrypted and unlikely to be exposed to eavesdroppers except in the case of SSL/TLS exploits. To mitigate the most risk, it is likely safest to review the security policies of chosen anti-theft software, with an

emphasis on encryption of transmitted data and validation of clients and servers to mitigate the possibility of leakage of sensitive data and masquerade attacks. To address privacy concerns related to a centralized, commercial server, having the anti-theft client communicate its information directly via email or to a specific user owned server is likely most effective if provided by the product. To protect one's self from vulnerabilities in preinstalled anti-theft applications, it is important to review such anti-theft applications and their system privileges and gauge their usefulness in comparison to their possible vulnerabilities. If a preinstalled anti-theft application is clearly vulnerable, such as with "Absolute Computrace", then removing the application is likely the safest route, although it may be difficult in some cases. Finally, to protect against man-in-the-middle and masquerade attacks, a user should be aware of their network surroundings and be watchful of any potential man-in-the-middle attacks.

Conclusion

Analysis of current anti-theft software, in particular the current program Prey, shows that anti-theft software is likely to be private and secure in most cases due to the encryption that many anti-theft software provides. However, there are some caveats to the security of anti-theft software. First, most are dependent on the security of SSL/TLS standards, which provides adequate confidence in most cases but may create vulnerabilities in cases such as the Heartbleed Bug. Another potential vulnerability to anti-theft software stems from the possibility of man-in-the-middle attacks when on a public networks. This risk may be mitigated with encryption of packet body content using a secret key shared between a predetermined client and server. Finally, a significant danger with anti-theft software stems from their system privileges, as many

instances of anti-theft software can lock and wipe devices. This means that the compromise of a system through anti-theft software is likely to have a large impact.

References

[1]S. Wash, "Macbook Sting: Computer Used to Catch Thief", *ABC News*, 2016. [Online]. Available: http://abcnews.go.com/Technology/stolen-macbook-tracked-hidden-app/story?id=13735348. [Accessed: 15- Dec- 2016].

[2]M. Anderson, "Technology Device Ownership: 2015", *Pew Research Center: Internet, Science & Tech*, 2016. [Online]. Available: http://www.pewinternet.org/2015/10/29/technology-device-ownership-2015/. [Accessed: 15- Dec- 2016].

[3]"Find My Mobile | Apps", *The Official Samsung Galaxy Site*, 2016. [Online]. Available: http://www.samsung.com/global/galaxy/apps/find-my-mobile/. [Accessed: 15- Dec- 2016].

[4]D. Storm, "Your PC or laptop may have a backdoor enabled by default, millions do", *Computerworld*, 2016. [Online]. Available: http://www.computerworld.com/article/2476651/malware-vulnerabilities/your-pc-or-laptop-may-have-a-backdoor-enabled-by-default-millions-do.html. [Accessed: 15- Dec- 2016].

[5]V. Kamlyuk, S. Belov and A. Sacco, "Absolute Backdoor Revisited", *Blackhat*, 2016. [Online]. Available: https://www.blackhat.com/docs/us-14/materials/us-14-Kamluk-Computrace-Backdoor-Revisited-WP.pdf. [Accessed: 15- Dec- 2016].

[6]"Cerberus anti theft - official website", *Cerberusapp.com*, 2016. [Online]. Available: https://www.cerberusapp.com/privacy. [Accessed: 15- Dec- 2016].

[7]"Prey", *GitHub*, 2016. [Online]. Available: https://github.com/prey. [Accessed: 15- Dec- 2016].

[8]F. Núñez, "SSL/TLS "heartbleed" vulnerability patched for all Prey services", *Preyproject.com*, 2016. [Online]. Available: https://www.preyproject.com/en/blog/2014/04/ssltls-heartbleed-vulnerability-patched-for-all-prey-services. [Accessed: 15- Dec- 2016].