Hashing Smartphone Serial Numbers

An ASLR Approach to Preventing Malware Attacks

Mark Wilson and Lei Chen
Department of Computer Science, Sam Houston State University, Huntsville, TX, USA

Abstract – *The Internet and mobile devices today have merged seamlessly, giving smartphone users access to the World Wide Web, email and other network services and resources. Due to the increased popularity of smartphones they have become a very attractive target for malware. It is predicted that smartphone users will see a multitude of different malware attacks aimed at their mobile devices in the near future. This paper presents how malware can spread to smartphones, and possible routes to safeguard smartphones against attacks. Specific defensive tactics of the Symbian Operating System will be outlined and a variation of Address Space Layout Randomization (ASLR) specifically for smartphones will be presented to prevent the spread of malware from both the Internet and other smartphones.*

Keywords: Address Space Layout Randomization, attacks, hash, malware, Smartphone, Symbian

1 Introduction

A generic cellular phone, that is only offering a phone-calling feature, has become rare and is difficult to find in many cell phone service provider storefronts. The majority of cell phone users have switched to smartphones that include not only a phone-calling feature, but also wireless Internet access to check emails, update social networking website statuses, and much more. Smartphones today often include digital audio players, high mega-pixel cameras, and either external or onscreen QWERTY keyboards for text messaging and emails [8]. Arguably the most valued feature of a smartphone is also its most detrimental: Internet access. By users having constant connectivity to the World Wide Web, the possibility of malware intrusion becomes extremely likely. Malware is identified as a piece of code that affects the behavior of the operating system (OS) or other security sensitive applications without the user's consent and by a method making the alterations impossible to detect by usual means [6].

Smartphone operating systems have been attacked and infected by multiple pieces of malware, most notably the Cabir worm that infected the Symbian OS. Symbian has employed a number of preventative measures to block the infection and spread of malware that aims to exploit weaknesses in the OS. Precautions that Symbian has installed to safeguard smartphones include a Trusted Computing Base (TCB), a Trusted Computing Environment (TCE), and Data Caging [1][7]. Though these methods hinder the infection of malware, an additional method could be introduced not only to the Symbian OS, but also to most other smartphone operating systems. A variation of Address Space Layout Randomization (ASLR) based on the hash value of each smartphone's serial number could not only prevent malware infection, but also allow telecom networks to track and trace the origin of any transmitted malicious code.

The rest of the paper is structured as follows. Next in Section 2, we survey the threats related to smartphone malware and the conventional solutions. Section 3 discusses the two major security measures, certificate signing and data caging, for Symbian Operating System. In section 4 we first briefly discuss Address Space Layout Randomization (ASLR), then introduce an ASLR based security enhancement solution to help prevent malware attacks by hashing smartphone serial number and renaming folders that need to be protected. Section 5 draws the conclusion and Section 6 proposes our future research.

2 Background

2.1 Smartphone malware threats

A number of factors contribute to weak malware protection on current smartphones. The first major reason is that a common OS is necessary for easy service creation. Unlike standard cell phones that relied solely on a proprietary OS that did not have to successfully communicate with another type of application or service, all smartphone operating systems provide the same basic foundation [2]. Powerful features such as: access to cellular networks and the Internet, multitasking for running several applications simultaneously, and data synchronization create a common ground. Unfortunately, this allows for

vast opportunities for security breaches and spread of malware infection. Most of software developers are eager to release new technologies or updated versions of current software but often neglect to properly or fully test them [10]. This failure to thoroughly harden the new software may lead to exploitation and ultimately corruption by malware. Finally, the users themselves are held accountable for some of their own habits. Similar to a PC, it is imperative that malware countermeasures such as firewalls and anti-virus monitoring be installed and working properly [4].

2.2 Solutions to safeguard smartphones

Though no single tactic is a failsafe to keep a smartphone malware-free, a combination of techniques can greatly reduce the likelihood of becoming infected. Installing and maintaining a secure firewall greatly limits the amount of traffic with internal or external peers. The user determines if another user is allowed access to a particular port, and the firewall will either grant or deny access to that port. Similar to PCs is the need for anti-virus software. By installing software to scan for malicious strings or patterns the user can be notified of a possible infection and can take proper measures in order to contain the malware and defend against any damage it may cause. Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) monitor the entire system for suspicious activity, including possible behaviors of malicious code. If abnormal behaviors are detected, closing ports or locking systems can result in order to prevent possible damage [5].

Smartphone hardening has been suggested in order to prevent the spread of malware to smartphone [4]. Simple actions, such as displaying the phone number of an incoming call and illuminating the LCD display when dialing, have already been employed by the smartphone to alert the user of suspicious activity. The smartphone's hardware itself can also play a role in reacting to malware infection. The smartphone's Subscriber Identity Module (SIM) card can be loaded with a clean, uninfected version of the smartphone OS. This tactic allows for the smartphone OS to be immediately reloaded with virtually no downtime. Finally, by turning off smartphone features that are not currently in use, such as Bluetooth or Wi-Fi, the chances of becoming infected may be reduced.

Due to smartphones having the ability to access the Internet and make calls, one smartphone has the potential of being screened by both the Internet and a Telecom network [4][5]. Smartphones being monitored by both of these types of networks can potentially protect the user from downloading or spreading malware. When many smartphones connect to the Internet, they are scanned in order to ensure that the latest security provisions are installed and that they will be shielded against possible threats. Internet access is denied if the smartphone is not patched with the most current security patches. Telecom networks are able to provide protection against the spread of malware because suspicious activity is easily identified at telecom base stations. Examples of suspicious activity monitored by telecom networks include initiating a call and immediately aborting it, connecting calls without voice traffic, or prolonged data packet transmission to or from a single user. If the telecom network determines that a smartphone is behaving suspiciously, the base station can limit the smartphone's rate of calling or transferring data, employ call filtering, or block the smartphone completely. These two networks have the ability to work together by notifying each other of abnormal network behavior. If either side were to alert the other, precautions including call filtering or denial of Internet connections could be employed.

3 Security Measures of Symbian

The Symbian OS is the most widely used smartphone operating system in use today, with Android trailing closely behind [2]. Due to the popularity of Symbian, it becomes a large target for malware, and unfortunately regardless of how secure an OS might be, there is still a potential for the contamination and spread of malware. However, Symbian employs a number of methods that enhance its security architecture.

3.1 Certificate signing

The Symbian OS requires that all applications must be digitally signed in order run on the user's smartphone. Symbian provides a security platform that is divided into three separate levels in order to maintain a secure environment. The deepest level of security allows an application to access the Trusted Computing Base (TCB) set, consisting of the kernel, the file server, and the software installer and its registry; however, in order for the application to gain access to the TCB, it must be certified and formally verified by Symbian [7][2]. The Trusted Computing Environment (TCE) set governs applications that require access to a limited amount of sensitive system resources [2]. Finally, the third level of trust is associated with third party applications that require a digital certification. Certifications can be performed in one of two ways depending upon the access level required of the application. Applications that must access core OS files in order to run properly must be submitted to the Symbian Signed program for approval [2]. On the other hand, applications that do not require access to sensitive or confidential system files can be self-signed by the developer and are granted limited access.

The requirement of verifying and signing applications before granting proper access in the Symbian environment is a strong strategy for containing possible malware. By locking sensitive files away from certain applications, malicious code would be unable to access and share a user's

personal information with others across networks. The self-signed applications, though only granted limited access, still pose a potential threat to Symbian users. Applications that are self-signed are able to access such limited privileges as making phone calls, initiating network connects, and accessing device location data. Malicious applications running only those features could potentially initiate and connect to a network, send data, sensitive or otherwise, and spread to another network [10].

3.2 Data Caging

Data Caging refers to a security architecture that divides sensitive data into separate folders each with different restrictions. The Symbian OS creates four different directories under the root file system: \sys, \resource, \private, and \(other) [7]. The \sys directory and subdirectories are only accessible by the trusted kernel. By restricting the \sys folder, it guarantees that only the trusted kernel can create executable files or load them into memory [1]. The \resource folder is used to store read-only files that will not be modified after installation, e.g. fonts and help files. All processes are able to read the files stored in this folder, but only the trusted kernel has permission to write to these files if modification is necessary. Each installed application creates a subdirectory under the \private directory in order to store files pertaining only to that particular application. The subdirectories are named using the secure identifier (SID) that identifies a running process. Applications are able to access, read, and write to their own files, but are not granted permission to read nor write to any other application's subdirectory. Finally, the \(other) directory has no permission restrictions and is designated as public, allowing the user to read and write to files housed in this directory [7].

The concept of data caging to restrict the user's accessibility to sensitive system files is a strong preemptive strike against malware intrusion. By limiting access to system files, the OS is able to retain its user-friendly status along with its integrity. However, thousands of Symbian users have posted protests in online forums speaking out against the limited accessibility to the files on their phones. Those users suggest Symbian keep the current settings as default settings, but allow power users to change those settings to allow themselves full access. The Internet is currently riddled with tutorials of how to hack Symbian and gain access to its system files using different applications. Due to Symbian being highly compatible with both C++ and Python programming languages, many power users are able to compose short scripts enabling applications to crack their Symbian-based smartphones.

4 ASLR based serial number hashing

The Symbian OS is a very strong and security conscientious operating system with very few access points for malware to breach. Nevertheless, because Symbian is ranked the most popular smartphone OS, it is likely that it will continue to be attacked. Symbian has already been attacked by such pieces of malware as the Cabir worm, CommWarrior, Skulls, and Doomboot among others. By examining methods employed by other operating systems and modifying them for smartphone usage, the digital world may be able to create not only a strong operating system that is nearly impervious to attack, but also a deterrent to those who want to employ smartphones as a means of attack against other networks.

4.1 Address Space Layout Randomization

Recently Microsoft Windows Vista and Macintosh OS X Snow Leopard edition began using a method of malware prevention called Address Space Layout Randomization (ASLR). ASLR is a security method that strengthens system security by increasing the variety of possible targets to attack [11]. This is effective because many viruses, worms, Trojan Horses and other types of malware search a computer system for a particular directory, file, or file type in order to properly infect the system. ASLR assigns random strings of characters to directories and files instead of their usual names in order to prevent possible threats from finding their target file. For example, a file contained inside of directory "\system32" is much more susceptible to infection than the same file being contained inside of a directory that has been assigned a random name. Programming a piece of malware to hunt for a particular file that has a different file path due to unpredictable folder names may prove to be a barrier that many malware authors simply do not want to invest the time into. Overcoming these obstacles will certainly slow an infection or attack, along with making it much more conspicuous [9].

In order to successfully employ an ASLR scheme and provide a distinguishable identifier of each smartphone, the serial number of the smartphone can be used. Each smartphone has its own unique serial number that is able to provide the vendor with the model number and technical specifications. By generating a hash value, using SHA-1, MD5, or a proprietary software format, from the smartphone's serial number a seemingly random number can replace common folder and file names that can still ultimately be linked to the smartphone itself.

4.2 Hashing serial number

For this research, a Nokia smartphone running the Symbian OS with the manufacturer-provided serial number "010082321439976/07951780736" will be illustrated. The smartphone's serial number generates an MD5 hash of "4abb107f8f8c4dc18482948081bdcb18" as shown in Figure 1. The checksum of the smartphone serial number allows for a more secure string of characters than simply the serial number alone. It is presumable that serial numbers are not purely random and particular digits or sets of digits signify

the make, model, capacity, original installed OS version number, and such, and could be deciphered by those familiar with Nokia smartphones. By hashing this identifying information, the likelihood of such data being deciphered is doubtful. To further randomize this hash number, eight consecutive characters from within the hash value are used as the folder names for the main three folders in the data-caging scheme already in place. By producing a 32-character hash string, it allows the eight character selection a total of 25 possibilities (4abb107f, abb107f8, bb107f8f, etc.); that is more than enough to apply to the security of one smartphone using this scheme and also allow for many other folder titles in future versions in the event of expanding this theory. Longer checksum values could also be used in the event of needing additional eight-character string titles. Using these randomized character strings based on the smartphone hardware itself would not only drastically reduce the occurrence of malware infection, but also provide information about the smartphone itself if an infection were to occur. Figures 2 illustrates the names of folder SYS before and after data caging, hashing serial number, and renaming.

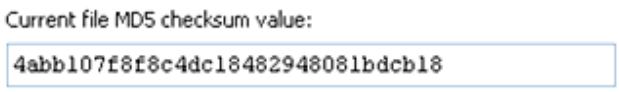

Figure 1. MD5 checksum of the smartphone serial number

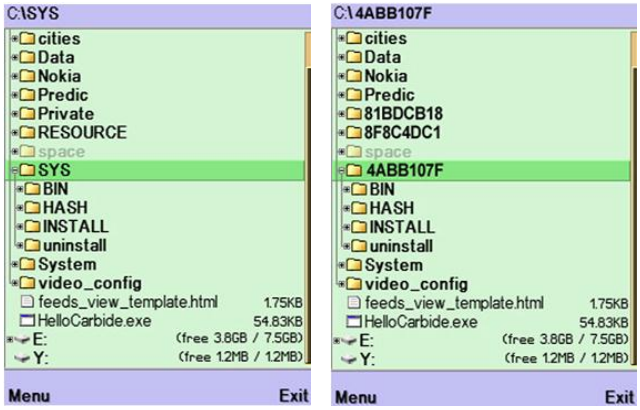

Figure 2. Folders before data caging (left) and Data caged folders renamed with eight-character hash string (right)

4.3 Security benefits

This variation of ASLR based on the smartphone hardware acts as a two-pronged attack against smartphone malware. First, the ASLR method compounds the difficulty of infecting the OS and ultimately prevents the smartphone from failing due to some type of malware attack. Second, because the ceasing of all intrusions, infections, and attacks, is unrealistic, the incorporation of the serial number and the information that is designated by it offers enough data about the smartphone to stop malware from spreading. By planting the smartphone's serial number directly into the system directory, smartphone service providers would be able to associate a particular piece of malware to a specific smartphone. Possibly infected smartphones could be reported by the owner of the smartphone itself who suspects it may be infected, technical support staff diagnosing smartphone issues over the phone or in person at a service provider storefront, or by a telecom carrier itself who suspects an attack or is currently being attacked by a particular smartphone. In the case of a telecom network identifying suspicious behavior originating from a smartphone, that network would have the ability to notify an Internet network and to take necessary countermeasures as outlined in Section 2.2.

The necessary data needed by a smartphone provider would be the reverse algorithm to use the three eight-character folder titles to successfully decode the original 32-character hash value, and to search it within the database containing the hash values of serial numbers of all smartphones sold. After the infected smartphone has been properly identified, especially if it is involved in some type of attack, the service provider would deny service to that phone. This would effectively cease any attacks launched by that smartphone along with the possible spread of infection over networks.

Giving the smartphone service provider the ability to remotely access the user's infected smartphone by serial number would also allow the smartphone's hard drive to be collected and researched. This information would be imperative for smartphone antivirus development in the case of a zero day attack. Because there are so many variations of malware in the wild today, Antivirus research would also benefit by studying and creating a virus definition to harden the system against future malware.

In the most extreme cases, malware could be linked to particular smartphones owned by particular people. If that smartphone were to be engaged in multiple attacks against a telecom or Internet carrier, the source of a particular piece of malware that intentionally infected others smartphones, had a history of engaging in suspicious activity such as multiple short calls, the proper jurisdiction could prosecute that user. The majority of states in the U.S. now have laws regarding malware and the spread of malware included in their respective penal codes. Federal law 18 U.S. Code § 1030 criminalizes computer crimes such as hacking, computer fraud and the spreading of computer malware [3]. The federal law defines computer trespass as "to knowingly access a computer without authorization or by exceeding authorized access and thereby obtain information protected against disclosure," a starting point for the introduction of malware into a system. Depending upon the behavior and target of the attack, 18 U.S. Code § 1030(5)(a) would apply to smartphones and smartphone malware transmission if the user "knowingly caused the transmission of a program, information, code, or command, and as a result of such

conduct, intentionally caused damage without authorization, to a protected computer."

5 Conclusion

In this paper a basic outline of smartphone malware was discussed along with selected security measures put into place by the popular smartphone OS, Symbian. Due to the increasing popularity of smartphones, malware specifically designed to infect, spread, and attack them is likely to be developed and put into operation. In order to successfully combat future threats of malware software developers need to remain diligent and continue to integrate counter measures into the operating system itself. In this paper, a variation of the standard ASLR scheme already employed by two major operating systems is proposed to further secure smartphones. To differentiate from the standard ASLR scheme, the hashing of the smartphone's serial number and the random selection of eight consecutive characters from that hash value allow for the smartphone's serial number to be retrieved directly by viewing the standard data caged folders in the Symbian OS. By obtaining the smartphone's serial number, it allows for possibilities of remotely ceasing attacks by disconnecting service, providing a sample of an infected smartphone for antivirus developers, and in the most extreme cases, gives law enforcement evidence to prosecute suspected malware developers, spammers, etc.

6 Future Work

Future work will include applying this model to other popular smartphone operating systems. Changes to convert this model to properly fit Windows-based, Android, or iPhone templates will vary depending upon the OS. However, because Windows and Mac OS X are already employing an ASLR technique on their desktop operating systems, a conversion to the smartphone OS employing this scheme may prove less intricate. In addition, data from Internet and telecom network providers must be gathered in order to properly test if a particular serial number can be linked to a specific person by their active account information. Furthermore, it must be researched to discover how crimes generated from smartphones are to be prosecuted in the event of a federal offense.

7 References

[1] T. Badura and M. Becher, "Testing the Symbian OS platform security architecture," 2009 international conference on advanced information networking and applications, May 2009.

[2] D. Barrera and P.C. van Oorschot, "Secure software installation on smartphones," IEEE security and privacy, December 2010.

[3] S.W. Brenner, "U.S. cybercrime law: defining offenses," Information system frontiers, 2004.

[4] C. Guo, H.J. Wang, and W. Zhu, "Smart-phone attacks and defenses," Third workshop on hot topics in networks, HotNets III, November 2004.

[5] S. Khadem, "Security issues in smartphones and their effects on telecom networks," Chalmers university of technology, August 2010.

[6] J. Rutkowska, "Introducing stealth malware taxonomy," White paper of COSEINC Advanced Malware Labs, November 2006.

[7] A. Savoldi and P. Gubian, "Symbian forensics: an overview," In Proceedings of International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Harbin, China, Proc. IEEE, 15-17 August 2008.

[8] A. Schmidt and S. Albayrak, "Malicious software for smartphones," Technische Universität Berlin, DAI-Labor, Tech. Rep. TUB-DAI 02/08-01, February 2008.

[9] P. Szor, "The art of computer virus research and defense," Upper Saddle, NJ: Pearson Education, Inc., 2005.

[10] L. Xie, X. Zhang, A. Chaugule, T. Jaeger, and S. Zhu, "Designing system-level defense against cellphone malware," 2009 28th IEEE international symposium on reliable distributed systems, December 2010.

[11] O. Whitehouse, "An analysis of address space randomization on Windows Vista," Symantec advanced threat research, March 2007.