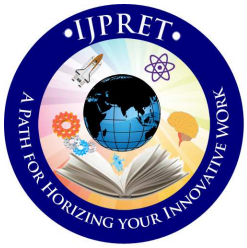
Research Article *ISSN: 2319-507X*
Trupti Lokhande, IJPRET, 2013; Volume 1(8): 926-937 *IJPRET*



INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

SMART-PHONES: A NEW APPROACH FOR SECURITY

TRUPTI S. LOKHANDE, PROF. RAJESHRI R. SHELKE

1. M.E. (CSE), H. V. P. M College of Engineering & Technology, Amravati, Maharashtra.
2. CSE Department, H. V. P. M College of Engineering & Technology, Amravati, Maharashtra.

Accepted Date:

27/02/2013

Publish Date:

01/04/2013

Keywords

Smart-phone,

GSM,

Telecom Networks,

Root kits

Corresponding Author

Ms. Trupti S. Lokhande

Abstract

Internet has been permeating into every corner of the world and every aspect of our lives, empowering us with anywhere, anytime remote access and control over information, personal communications (e.g., through smart-phones), and our environment (e.g., through the use of sensors, actuators, and RFIDs). While enabling interoperation with the Internet brings tremendous opportunities in service creation and information access, the security threat of the Internet also dauntingly extends its reach. Smart phones are increasingly being equipped with operating systems that compare in complexity with those on desktop computers. This trend makes smart phone operating systems vulnerable to many of the same threats as desktop operating systems. In this paper, we wish to Alarm the community that the long-realized risk of interoperation with the Internet is becoming a reality: Smart phones, interoperable between the telecom networks and the Internet, are dangerous conduits for Internet security threats to reach the telecom infrastructure. The damage caused by subverted smart-phones could range from privacy violation and identity theft to emergency call center DDoS attacks and national crises. We also describe defense solution space. In this paper, we focus on the threat posed by smart phone root kits. Root kits are malware that stealthily modify operating system code and data to achieve malicious goals, and have long been a problem for desktops. We use three example root kits to show that smart phones are just as vulnerable to root kits as desktop operating systems. However, the ubiquity of smart phones and the unique interfaces that they expose, such as voice, GPS and battery, make the social consequences of root kits particularly devastating.

I. INTRODUCTION

Over the last several years, the decreasing cost of advanced computing and communication hardware has allowed mobile phones to evolve into general-purpose computing platforms. Over 115 million such smart phones were sold worldwide in 2007.These phones are equipped with a rich set of hardware interfaces and application programs that let users interact better with the cyber and the physical worlds. To support the increasing complexity of software and hardware on smart phones, smart phone operating systems have similarly evolved. Modern smart phones typically run complex operating systems, such as Linux, Windows Mobile, Android and Symbian OS, which comprise tens of millions of lines of code.

Smart-phones

Smart-phone is the trend of unified communications which integrate telecom and Internet services onto a single device because it has combined the portability of cell-phones with the computing and networking power of PCs. As illustrated in Figure 1, smart-phones, as endpoints of both networks, have connected the Internet and telecom networks together.

Figure 1: Smart-phones become end-points of Both the Internet and telecom networks.

Another key reason for this trend is the ease and low cost of introducing new integrated Internet and telecom services. Easy service creation demands common operating systems (OS). Because smart-phones are typically as powerful as a few year-old PCs, their operating systems have evolved to be rather full-fledged. Smart-phone OSes today include Symbian OS, Microsoft Smart-phone OS, Palm OS, and embedded Linux. Although the detailed design And functionality vary among these OS vendors, all share the following features

• Access to cellular network with various cellular standards such as GSM /CDMA and UMTS.

• Access to the Internet with various network interfaces such as infrared, Bluetooth, and use standard TCP/IP protocol stack to connect to the Internet.

• Multi-tasking for running multiple applications

Simultaneously.

• Data synchronization with desktop PCs.

• Open APIs for application development.

While common OSes, open APIs, and sophisticated capabilities enable powerful services, they also create common ground and opportunities for security breaches and increase worm or virus spreading potentials. Given the PC-like nature of smart-phones and the trend of full-fledged OSes, software vulnerabilities seem inevitable for their OSes and applications.

II. COMPROMISING SMART-PHONES

There are three venues for a smart-phone to be compromised:

A. Attacks from the Internet: Since smart-phones are also Internet endpoints, they can be compromised the same way as the PCs by worms, viruses, or Trojan horses. The first Symbian based Trojan has recently been discovered in a popular game software.

B. Infection from compromised PC during data synchronization: Smart-phone users typically synchronize their e-mails, calendar, or other data with their desktop PCs through synchronization software like ActiveSync. There exist trust relationships between smart-phones and their respective synchronization PCs. Therefore, to ultimately infect smart-phone, attackers can first infect its synchronization PC, and

then the smart-phone will be infected at the next synchronization time.

C. Peer smart-phone attack or infection: A compromised smart-phone can actively scan and infect peer smart-phones through its Wireless Personal Area Networks (WPAN) interface such as Bluetooth or UWB (ultra wideband). Since smart-phones are mobile devices, they can infect new victims at different locations. The first smart-phone worm, Cabir , uses this method. It is also possible that a cellular phone can be crashed by a malformed SMS text message

III. ATTACKS IN SMART-PHONE

ATTACK 1:SMART-PHONE ATTACKS AGAINST THE TELECOM NETWORKS

Once a smart-phone is compromised from the Internet, it also becomes a source of malice to the telecom networks that it has access to. Before we describe the attacks, we first give a brief description of the GSM cellular network, as an example of telecom networks against which smart-phone attacks can be launched. Nevertheless, the attacks we describe here can be applied to other cellular networks, such as CDMA, as well.

1. Base Station DoS

Compromised smart-phones can easily make phone calls, say using Microsoft Smart-phone SDK API Phone Make-Call, to call other phone numbers obtained from sources like yellow pages. The radio channel of a GSM base station with n carrier frequencies can be completely exhausted by $8n$ well coordinated smart-phone zombies in the same cell initiating calls and using up all the time slots of a base station. The zombies can hang up as soon as their call setups complete and then re-initiate new calls, and so on. In the case that a callee is also subverted, the callee smart-phone can be configured deliberately not to answer the phone, occupying the time slot at both the caller and the callee side for about one minute in each call attempt. Since the callee does not accept the call, the caller would not even need to pay for this unfinished call,

2. DDoS Attack to Call Centers

This attack is similar to the previous one, but the goal is not to exhaust radio resources, but to put call centers to a halt. This is in the same spirit as the Internet DDoS attacks to web servers. Such attacks are not possible in the past with traditional

telephones because one would have to manually dial call center numbers. This requires attackers to be physically co-located with many phones. Consequently, the attackers can be easily traced back, caught, then legally prosecuted. For the case of smart-phone zombies, their owners are most likely the victims rather than the attackers themselves. Therefore, tracing back to the true attackers becomes a much more difficult task.

3. Spamming

Attackers can manipulate smart-phone zombies to send junk or marketing messages through SMS. In the case that the charging model is flat, a compromised smart-phone can spam for "free"; and therefore its owner may not even notice its bad behavior. Free SMS spamming gives attackers good incentives to compromise smart-phones.

4. Identity Theft and Spoofing

Telephone numbers or IMSIs stored on SIM cards are difficult to spoof, which is the basis of authentication and accounting mechanisms in telecom networks. In the past, researchers have attempted and successfully spoofed SIM cards. However, the procedure involves physical access to a SIM card, and requires about 150,000 queries to the stolen card, which could last as long as 8 hours. Mass cloning with attacks in this nature is hard. However, this is not the case with smart-phones. Identity theft with smart-phones is trivial — once a smart-phone is compromised, the attacker literally possesses its owner's identity for any activities in her name. This is especially alarming since in many countries, one's SIM card serves as her identity card for voting, ordering goods, or accessing her finance. Further, call center services evolve to be completely automatic, which enables attackers to carry out automatic response.

DEFENSE

We address defense for smart-phone attacks from four angles: How we may harden smart-phones themselves to be less vulnerable; Internet-side defense; telecom-side defense; and what coordination's between the Internet and telecom networks may be needed. We don't intend to give full-fledged or bullet-proof solutions, but rather to layout the landscape of the solution space, and point out interesting

and challenging topics of research in this area.

A.Smart-Phone Hardening

People have long favored functionality over security and are unwilling to pay the price and inconvenience incurred by security schemes.

• *Attack surface reduction:* One simple defense is to reduce the attack surface as much as possible. This defense mechanism has also been applied to PCs, but with limited success because it is disruptive to popular applications like file-sharing and network printer. Nevertheless, this mechanism may be more effective for smart-phones because the smart-phone usage model is different from that of PCs.

OS hardening: Smart-phone OSes can enforce some security features, such as always displaying callee's number and lighting up LCD display when dialing. This can be achieved by only exporting security enhanced APIs to applications. With hardened OSes, unless attackers can subvert the smart-phone OS without being noticed, attacking actions from malicious user-level code can be more easily detected by the smart-phone user.

• *Hardware hardening:* We believe one advantage we can leverage for smart-phone hardening is that smart-phone already has an embedded smart-card, the SIM card. The SIM card has evolved to incorporate the use of the SIM Toolkit (STK) — an API for securely loading applications to the SIM. STK allows the mobile operator to create or provision services by loading them into the SIM card without changing anything in the GSM handset.

B. Internet Side Protection

The malware defense mechanisms that have been deployed or proposed for the Internet can be readily applied to smart-phones. For example, more rigorous process in software patching or vulnerability-driven network traffic shielding will certainly strengthen the defense for smart-phones for known vulnerabilities, though not unknown ones. It would be desirable for smart-phone Internet service providers to ensure that devices that access them are properly patched or shielded — unlatched or unshielded ones should not be exposed to the wild Internet. Currently, majority of smart phones access the Internet through

telecom data networks such as GPRS or CDMA1X.

C. Telecommunication Side Protection

There will always be some subverted smart-phones no matter how much Internet-side protection there is. Telecom networks must introduce misbehavior detection and reaction mechanisms to sustain its normal operation. Fortunately, unlike the Internet traffic, telecom traffic is highly predictable and well managed since they can only be voice or SMS traffic. Therefore, it is not difficult to identify abnormal behaviors.

ATTACK 2: ROOTKITS ON SMART PHONES

The increasing complexity of smart phone operating systems makes them as vulnerable to root kits as desktop operating systems are. However, these root kits can potentially exploit interfaces and services unique to smart phones to compromise security in novel ways. In this section, we present three proof-of-concept root kits that we developed to illustrate the threat that they pose to smart phones. They were implemented by the first two authors, with only a basic undergraduate-level knowledge of operating systems. Our test platform was a Neo Free runner smart phone running the Open make Linux distribution.

A. Spying on Conversations via GSM Goal.

The goal of this attack is to allow a remote attacker to stealthily listen into or record confidential conversations using a victim's root kit-infected smart phone. *Attack Description.* The Forerunner phone is equipped with a GSM radio, which is connected via the serial bus and it is therefore available to applications as a serial device. During normal operation of the phone, user-space applications issue system calls to the kernel requesting services from the GSM device. The GSM device services the request allowing the application to access the telephony functionality provided by the device. GSM devices are controlled through series of commands, called AT (attention) commands, that let the kernel and user-space applications invoke specific GSM functions. For example, GSM devices support AT commands to dial a number, fetch SMS messages, and so on. To maliciously operate the GSM device, e.g., to place a phone call to a remote attacker, the root kit must therefore issue AT commands

from within the kernel. *Social Impact.* Snooping on confidential conversations has severe social impact because most users tend to keep their mobile phones in their proximity and powered-on most of the time. Root kits operate stealthily, and as a result, end users may not even be aware that their phones are infected. Consequently, an attacker can listening on several conversations, which violates user privacy, ranging from those that result in embarrassing social situations to leaks of sensitive information.

B. Compromising Location Privacy using GPS

Goal. The goal of this attack is to compromise a victim's location privacy by ordering the victim's root kit-infected smart phone to send to the remote attacker a text message with victim's current location (obtained via GPS). *Attack Description*. As with the GSM device, the GPS device is also a serial device. The kernel maintains a list of all serial devices installed on the system. A rootkit can easily locate the GPS device. Every serial device contains a buffer in which the corresponding device stores all outgoing data until it is read by a user-space application. Our prototype root kit uses this buffer to read information before it is accessed by user-space applications. This allows us to monitor and suppress incoming SMS messages and also query the GPS for location information. A root kit that compromises location privacy as described above must implement three mechanisms. *Social Impact*. Protecting location privacy is an important problem that has received considerable recent attention in the research community. By compromising the kernel to obtain user location via GPS, this root kit defeats most existing defenses to protect location privacy. Further, the attack is stealthy. Text messages received from and sent to the attacker are not displayed immediately to the victim. The only visible trace of the attack is the record of text messages sent by the victim's phone, as recorded by the service provider.

C. Denial of Service via Battery Exhaustion

Goal. This attack exploits power-intensive smart phone services, such as GPS and Bluetooth, to exhaust the battery on the phone. This root kit was motivated by and is similar in its intent to a previously proposed attack that stealthily drains a smart phone's battery by exploiting bugs in the MMS

interface. However, the key modifying the smart phone's operating system. *Attack Description.* The GPS and Bluetooth devices can be toggled on and off by writing a "1" or a "0" respectively, to their corresponding power device files. The root kit therefore turns on the GPS and Bluetooth devices by writing a "1" to their corresponding power device files. To remain stealthy, the root kit ensures that the original state of these devices is displayed when a user attempts to view their status. Most users typically turn these devices off when they are not in active use because they are power-intensive. *Social Impact*. This attack quickly depletes the battery on the smart phone. In our experiments, the root kit depleted the battery of a fully charged and infected Neo Free runner phone in approximately two hours (the phone was not in active use for the duration of this experiment). In contrast, the battery life of an uninfected phone running the same services as the infected phone was approximately 44 hours. We also simulated the effect of such a root kit on the Verizon Touch and ATT Tilt phones by powering their GPS and Bluetooth devices. In both cases, battery lifetime reduced almost ten-fold. Because users have come to rely on their phones in emergency situations, this attack results in denial of service when a user needs his/her phone the most.

DETECTING PHONE ROOTKITS

Root kits vary in the sophistication of the attack techniques that they use. Root kits that modify system utilities and some kernel modules often leave a disk footprint, and can possibly be detected using user-space malware detection tools. However, these tools rely on the operating system to provide critical services, such as access to files, and root kits can easily bypass them using more sophisticated techniques. More sophisticated root kits operate by modifying arbitrary data structures on the kernel's heap. It is therefore well accepted that root kit detection mechanisms must reside outside the control of the operating systems that they monitor.

Hardware-supported Root kit Detection

Hardware-assisted root kit detectors operate by using special purpose hardware to directly access kernel memory via DMA. For example, such root kit detectors can use secure co-processors or PCI cards to access

kernel memory, and ensure the integrity of kernel data structures. In this approach, the machine being monitored is equipped with the above hardware, and is physically connected to another machine, which fetches and scans its memory.

VMM-based Rootkit Detection

Virtualization offers an alternative approach to implement rootkit detection. In this approach, the smart phone's operating system and the monitor execute in separate virtual machines (VM). The monitor queries the VM that runs the phone's operating system and extracts the contents of its memory locations to perform rootkit detection. A number of commercial efforts are currently underway to build virtual machine monitors for smart phones , with the goal of allowing users to have multiple personalities on a single physical device.

IV. CONCLUSION

In this position paper, we wish to alert the community on the imminent dangers of potential smart-phone attacks against telecom infrastructure, the damages caused by which could range from privacy violation and identity theft to emergency center outage resulting in national crises. We have outlined a number of defense strategies, many of which demand much further research. We also studied Rootkits evade detection by compromising the operating system, thereby allowing them to defeat user-space detection tools and operate stealthily for extended periods of time. This paper demonstrated that kernel-level rootkits can exploit smart phone operating systems, often with serious social consequences. The popularity of the mobile platform has already attracted attackers, who have increasingly begun to develop and deploy viruses and worms that target these platforms.

V. REFERENCES

1. Ian Angus. An Introduction to Erlang B and Erlang C. *Telemanagement*, July-August 2001.

2. Live Bos and Suresh Leroy. Toward an All-IP-Based UMTS System Architecture. *IEEE Network*, January and February 2001.

3. Jian Cai and David J. Goodman. General Packet Radio Service in GSM. *IEEE Communications Magazine*, October 1997.

4. Microsoft Corporation. New Security Technologies in Windows XP Service Pack 2

(SP2). http://msdn.microsoft.com/security/productinfo/xpsp2/default.aspx?pull=/library/enusdnwxp/html/securityinxpsp2.asp.

5. Microsoft Corporation. Windows Mobile-based Smartphones. http://www.microsoft.com/ windowsmobile/smartphone/default.mspx.

6. St´ephane Coulombe and Guido Grassel. Multimidia Adaptation for the Multimedia Messaging Service. *IEEE Communications Magazine*, July 2004.

7. David Dagon, Xinzhou Qin, Guofei Gu, Wenke Lee, Julian Grizzard, John Levine, and Henry Owen. Honeystat: Local Worm Detection Using Honeypots. In Proceedings of the 7th International Symposium on Recent Advances in Intrusion Detection (RAID), 2004.

8. Trusted Computing Group. TCG TPM Specification Version 1.2: Design Principles https://www.trustedcomputinggroup.org/home.

9. Harri Honkasalo, Kari Pehkonen, Markku T. NieMi, and Anne T. Leino. WCDMA and WLAN for 3G and Beyond. *IEEE Wireless Communication Magazine*, April 2002.

10. F-secure warns of mobile malware growth. www.vnunet.com/ vnunet/news/2230481/f-secure-launches-mobile.

11. Google fixes android root-access flaw. www.zdnetasia.com/news/security/0,39044215,62048148,00.htm.

12. Mcafee mobile security report 2008. www.mcafee.com/us/ research/mobile_security_report_2008.html.

13. OKL4 embedded hypervisor: Open kernel labs. www.ok-labs. com/.

14. Openmoko Neo FreeRunner. wiki.openmoko.org/wiki/ Neo_FreeRunner.

15. Qtopia software stack (Qtextended.org). qtopia.net.

16. Smartphones will soon turn computing on its head. news.cnet. com/8301-13579_3-9906697-37.html.

17. VMware mobile virtualization platform. www.vmware.com/ technology/mobile/.

18. Rootkits, part 1 of 3: A growing threat, April 2006. MacAfee AVERT Labs Whitepaper.

19. A. Baliga, V. Ganapathy, and L. Iftode. Automatic inference and enforcement of kernel data structure invariants. In Proc. Annual Computer Security and Applications Conference, 2008.

20. A. Baliga, L. Iftode, and X. Chen. Automated containment of rootkit attacks. Computers & Security, 27(7-8):323 – 334, 2008.