CHAPTER

Mobile Applications and Devices

10

INFORMATION IN THIS CHAPTER

- Why Mobile Applications?
- Platforms
- Trust
- Attacks
- Securing Mobile Devices
- Secure Mobile Applications

From providing data access to allowing remote control of systems, mobile applications will play a large role in the smart grid. As mobile devices continue to grow in popularity, consumers and utility personnel alike will integrate mobile devices more with their daily lives and use them to interact with smart grids. As a result, attackers view mobile devices as higher value targets.

Mobile devices will access smart grid resources through mobile applications designed specifically for mobile devices. However, these mobile applications will be attacked by every type of device capable of accessing the applications. Since most of the mobile applications will be accessible via the Internet, attackers will be able to use any type of device to attack these applications. This chapter will describe the possible attacks against mobile applications and mobile devices, and then explore how to protect these applications and devices.

WHY MOBILE APPLICATIONS?

Cell phones and other mobile devices have become prevalent in today's society. At the end of February 2010, comScore (www.comscore.com) estimated that roughly 234 million Americans (aged 13 years or older) were mobile subscribers.[1] Thus, mobile devices provide a convenient platform that utility companies can utilize to distribute energy consumption information to their consumers in a manner that consumers are comfortable with. Additionally, mobile devices will allow consumers to rapidly adapt to consumption information by enabling consumers to change their energy consumption from anywhere.

Although smart grid functionality will allow utility companies to remotely diagnose and fix most problems in electric grids, there will still be situations that require

A Guide to Kernel Exploitation. DOI: 10.1016/B978-1-59749-570-7.00010-8

employees to physically visit the location of the problem. Mobile devices will enable these employees to stay connected with their organization's network and resources while out in the field diagnosing and fixing issues. Some of these devices will be able to interact flawlessly with existing applications; however, some devices will require applications specifically designed to allow access from mobile devices.

PLATFORMS

Mobile application developers seeking to reach the broadest audience will have to support multiple platforms. The following list contains some of the more common mobile platforms:

- Cell phones (smart and regular)
- Mobile Internet devices (MID)
- Portable media players (PMPs)
- Laptops
- Netbooks
- Ultra mobile personal computers (UMPC)
- Tablet PCs

From a functionality standpoint, developing mobile applications for mobile platforms is traditionally difficult due to how varied the devices are. The application will need to display correctly on devices ranging from smart phones with three-inch screens to laptops with 17-inch screens. Similarly, the security requirements for these platforms will vary depending on the specifications for each supported mobile device.

TRUST

Chapter 9, "Third-Party Services," discussed the trust relationships between utility companies and third-party service providers, and this chapter will describe the trust relationship between the application developers and the consumers. However, in this case, the consumers will be taking a further leap of faith when using the mobile applications. For the most part, the consumers will just click the **I Agree** button and blindly accept the developer's terms of service.

EPIC FAIL

Have faith in consumers spending the time to make informed decisions regarding what applications to install? Consider the Lose/Lose video game (www.stfj.net/art/2009/loselose/), developed by Zach Gage, as an example. The video game closely resembles Space Invaders, but with one important difference. For every alien space ship the user destroys, the video game deletes a file in the user's home directory.[2] On his Web site, Zach clearly warns that the application will permanently delete files. Additionally, when you start the video game, it will warn the user that killing the alien ships will permanently delete files on the hard drive. Yet, on the video game's Web site, the list of high scorers contains over a thousand players.

Trusting Strangers

Would you trust a stranger with your online banking password? If not, why would you trust an application developed by a stranger with your online banking password? Without performing a source code review of the application, how can a consumer be assured their information will not be compromised? The application could provide a back door that allows the malicious developer to access the consumer's mobile device remotely or the application could be sending data to the developer's malicious server. From a smart grid perspective, would you provide a stranger with a key to your house to adjust your thermostat's setting to reflect changing electric prices? If not, why would you trust an application developed by a stranger with the password to control smart devices in your home?

One argument to trust the application is that you trust the parent company to provide the proper oversight, as well as the dozens of strangers who have posted reviews. Let's take Apple's iTunes (www.apple.com/itunes/) App Store as an example, although Google's Android Market (www.android.com/market/) and BlackBerry's App World (http://na.blackberry.com/eng/services/appworld/) could be used in this example as well. Most consumers have grown to trust Apple through using iTunes to download music, movies, and television shows. With the tremendous success of their iPhone, iPod Touch, and iPad, developers have created an overwhelming number of applications for these mobile devices. However, Apple does not blindly allow any application to be distributed through their App Store. For example, Apple has prohibited applications that include sexual content and has pretty thoroughly rejected or removed most applications with sexual content on that basis.[3] So how thoroughly has Apple screened the applications to prevent malicious applications from being distributed in the App Store? Although consumers can rest assured that any respectable company would remove malicious applications, finding sexually explicit material in an application is a lot easier than finding back doors in an application. Performing a thorough security assessment of each application would be too time consuming and thus unrealistic. So the question then changes from is there an app for that to is that app secure, and should you trust the new smart grid mobile applications on your device?

Code Signing

Code signing is one method the software industry has used to address the issue of trust with applications downloaded from the Internet. Code signing uses digital signatures to confirm the authenticity and integrity of the application. Specifically, the author of the application will digitally sign the executable(s) using their private encryption key, or signing key, and any recipients of the executable can use either the author's public key or a certificate authority (CA) to validate the signature. Theoretically, this method would prevent anyone from tampering with the application, which should instill trust in the user.

NOTE

A digital signature is a cryptographic transformation that allows the verification of origin authentication, data integrity, and signatory nonrepudiation.[4] For more information on digital signatures, NIST FIPS PUB 186-3 provides a very detailed perspective on how digital signatures work. If you are unfamiliar with digital signatures, or public key infrastructure (PKI) in general, a beginner's guide to PKI can be found at: http://articles.techrepublic.com.com/5100-10878_11-5839988.html.

Although code signing can help mitigate the risk involved with downloading applications from the Internet, the risk is not completely eliminated. It is important to note that code signing does not prevent the original developer from inserting malicious code into their application. Additionally, the user needs to trust that the developer has protected their private encryption key. If a third party has compromised the developer's private key, then that third party would be able to tamper with the application and digitally sign the executable as the original developer.

As a real-world example of code signing, iPhones require all applications to be signed by Apple.[5] Thus, developers will need to request a signing certificate from Apple before their application can run on iPhones and be distributed in the App Store. So in this scenario, Apple acts as the CA, who can issue and revoke signing certificates. If an application is later determined to be malware, Apple could revoke the signing certificate for that developer and application to prevent their application from further spreading. Additionally, Apple could refuse to issue further signing certificates to that developer, effectively banning that developer from distributing more malware through the App Store.

Apple's code signing technique is susceptible to attacks, though. At the 2010 CanSecWest conference, Halvar Flake compromised an iPhone in the Pwn2Own contest on the first day using an exploit developed by Vincenzo Iozzo and Ralf-Philipp Weinmann.[5] The attack was performed by browsing a malicious Web site on the iPhone, which forced the iPhone to disclose the contents of its SMS database.[5] The exploit circumvents the code signing security control by using return-oriented programming techniques, which manipulate the function call stack to convert valid code into a malicious payload.[5]

ATTACKS

Mobile applications and devices will be targets in the smart grid. Attackers generally take the path of least resistance to achieve their goal, and mobile applications provide an easily accessible target. Attacks against certain mobile devices, such as cell phones and PMPs, have been limited in the past; however, this will dramatically change as their integration into society and level of connectivity continue to increase.

Why Attack the Handset?

Mobile devices, such as laptops, netbooks, and tablet PCs, generally run traditional operating systems, applications, and share the same connection speeds as their nonmobile variants. Thus, the reasons for attacking these devices should not be a surprise. Cell phones, MIDs, and PMPs have only recently become major targets for attackers. Faster Internet connections and the sensitive data now stored on these types of devices will compel attackers to consider them as better targets.

Cell phones used to store a very small amount of information, including names and phone numbers, which provided little to attackers. The same can be said for PMPs, which would traditionally store media files and maybe some personal information about the owner. However, these devices have been transformed into multipurpose computers that can perform the same functions as other mobile devices such as laptops. The following list contains examples of what can be found on mobile devices:

- VPN keys
- Corporate e-mail passwords
- Authentication tokens
- GPS location
- Sensitive information cached by applications

Additionally, most cell phones and PMPs will sync with other devices, such as laptops or desktops. So, if an attacker compromises a cell phone or a PMP, they could use that compromised device to attack the system it syncs with.[6] When the user plugs in the device or connects it to the host system wirelessly, the device could compromise the host system's operating system.[6] As a recent example, consider the incident where Vodafone was shipping HTC Magic cell phones with malware pre-installed.[7] Specifically, when a user connected their new phone to their computer, the phone would attempt to install a Mariposa bot client, as well as a Confiker and Lineage password stealing program onto their computer.[7]

SMS

Short Message Service (SMS) messages are often seen as harmless text messages; however, attacks can be carried out through the SMS messaging systems on cell phones. For example, Charlie Miller and Collin Mulliner demonstrated at the Black Hat (www.blackhat.com) USA 2009 conference an attack against iPhones using malicious SMS messages.[8] By sending specially crafted SMS messages, an attacker could obtain complete control over the iPhone.[8] Additionally, further research showed that other cell phones were also partially vulnerable to this same attack.[8]

E-mail

If an attack can occur via SMS, then attacks will also occur over e-mail. E-mail attacks targeting mobile devices may take the same form as the example SMS attack; however, they will also take other forms, such as phishing attacks that attempt to spoof messages from the utility companies in order to get sensitive information out of their targets.

NOTE

Other variants of phishing attacks include spear phishing and whaling attacks. Spear phishing refers to phishing attacks that are targeted at certain groups. So, for example, the attacker would first attempt to identify a group of e-mail addresses for a specific utility company and then craft an e-mail that spoofs that particular utility company, thus, improving the likelihood of success in their attack. Whaling attacks are similar to spear phishing except that they further refine the target list to senior executives or other high-profile targets.

As an example, consider a two-vector attack that targets a business for ransom or blackmail. The attacker initially compromises the smart meter that controls various smart devices (such as HVAC, cooling, and of course, power for the company's servers) within the company's data center. The attacker then sends e-mails to each member of the board of directors for the company, whose e-mail addresses are listed publicly on the company's Web site. The e-mails state that if the company does not pay the attacker $5 million (or some appropriate amount), the attacker will turn off power to the mission critical systems, increase the temperature to unsafe levels, and wreak havoc with the HVAC. The company's backup power generators may provide temporary support, but will the board want to risk the company's business operations on that chance?

Malicious Web Sites

Mobile Web browsers can be attacked just like traditional Web browsers, thus, following links sent via e-mail on your cell phone does not provide any more protection than on your desktop. Most attackers will create malicious Web sites that target the broadest audience, which usually ends up being desktop Web browsers such as Internet Explorer or Safari. However, an attacker who is determined to target your mobile device will be able to setup a malicious Web site to compromise mobile Web browsers.

Physical

When discussing threats to mobile devices, the physical security threat usually rises to the top of the list. By their very nature, mobile devices will go outside the protective physical security layers of fences, security guards, and security cameras. Thus, these mitigating controls will cease to reduce the risk of physical security threats.

The primary physical security threat to mobile devices will be theft. Mobile devices are already seen as a target for thieves for their resale potential, but attackers will also view these devices as methods to bypass perimeter security controls. As an example, consider utility company maintenance workers who spend most of their days out in the field-fixing issues that cannot be fixed remotely, such as a downed power line. The maintenance worker carries around a laptop with an internal 3G wireless card that provides Internet access over a cellular network. The maintenance worker connects to the utility company's network through the corporate VPN to check their e-mail and check the sensor networks that detect when there is a problem within the electric grid. When the maintenance worker arrives at the downed power line, he leaves the laptop in the car. If an attacker can steal the laptop that is still connected to the VPN, then he has most likely gained initial access to internal resources in that organization. Hopefully, the organization has compensating controls that would prevent unrestricted access to the internal network from a VPN connection, but the attacker would still have gained an access point to that organization.

From a consumer perspective, consider the upcoming Chevy Volt OnStar mobile application as another example. Mobile phone applications are currently being developed for the iPhone, BlackBerry, and Android platforms, in addition to a Web interface.[9] These mobile applications will allow Chevy Volt owners to remotely control the following features:

- Display charge status
- Provide flexibility to "Charge Now" or schedule charge timing
- Display percentage of battery charge level, electric, and total ranges
- Allow owners to manually set grid-friendly charge mode for off-peak times when electricity rates are lowest
- Send text or e-mail notifications for charge reminders, interruptions, and full charge
- Show miles per gallon, EV miles and miles driven for last trip and lifetime
- Remotely start the vehicle to precondition the interior temperature
- Unlock/lock the car doors[9]

The application integrates nicely into the smart grid goal to reduce consumer energy consumption during peak times by allowing the owner to set the charging schedule to off-peak times. If an attacker were able to steal the device, they could wreak havoc on the owner. Consider the example of a Chevy Volt owner who lives in a warm place, such as Florida. During the summer, cars that are parked outside can reach unbearable temperatures that exceed 100 degrees Fahrenheit. So, the natural elements alone can cause damage to items inside the car. Then, consider if the attacker were to turn the car on remotely. The car will try to adjust the temperature to be comfortable, thus causing the battery and gas tank to drain. The attacker could continue to drain the battery by repeatedly turning on the car until the mobile application informs the attacker the battery is empty. Alternatively, the attacker could just use the mobile application's ability to unlock the doors and steal the car.

Mobile devices are often recycled, resold, or thrown away without properly removing sensitive information. EBay (www.ebay.com) has made selling these devices an easy transaction but if the information is not removed properly, the buyer could become access to the information and resources stored on the mobile device. According to a recent research survey, 40 percent of hard drives bought on eBay

Table 10.1 Sample secure deletion resources[12]

Tool	Device support	License type
Dariks boot and nuke (DBAN – www.dban.org/)	ATA/IDE, SATA	Free
Roadkill's datawipe (www.roadkil.net/)	ATA/IDE, SATA Flash	Free
Secure erase 4.0 http://cmrr.ucsd.edu/people/Hughes/SecureErase.shtml)	Solid-state drives (SSD)	Free
Wireless recycling (www.recellular.com/recycling/data_eraser/)	Cell phones	Free

contain sensitive information from their previous owners, including corporations and government agencies.[10] In 2009, data for a U.S. missile air defense system were discovered on a hard drive bought on eBay.[11]

TIP

The process to securely erase flash memory is different than erasing traditional storage devices and will most likely require different applications.[12] Alternatively, degaussing or physically destroying the device memory can also achieve the same goal. See Table 10.1 for a list of several examples of secure deletion tools.

SECURING MOBILE DEVICES

Mobile applications will mainly be developed to support access through two mediums: a mobile Web site and a client application installed on the mobile device. In either case, the mobile device will need to be protected to prevent unauthorized access to the mobile application. Traditional security controls can be extended to protect many mobile devices; however, new issues that arise from mobile devices will need to be addressed by additional security controls.

The following best-practice guides and tips have been developed for mobile device security:

- US-CERT Cyber Security Tip ST06-007: Defending Cell Phones and PDAs Against Attack – www.us-cert.gov/cas/tips/ST06-007.html
- US-CERT Cyber Security Tip ST05-017: Cybersecurity for Electronic Devices – www.us-cert.gov/cas/tips/ST05-017.html
- US-CERT Cyber Security Tip ST04-020: Protecting Portable Devices: Data Security – www.us-cert.gov/cas/tips/ST04-020.html
- NIST SP 800-124: Guidelines on Cell Phone and PDA Security – http://csrc.nist.gov/publications/nistpubs/800-124/SP800-124.pdf
- NIST SP 800-101: Guidelines on Cell Phone Forensics – http://csrc.nist.gov/publications/nistpubs/800-101/SP800-101.pdf

Traditional Security Controls

Most mobile devices can utilize the same controls as nonmobile devices, especially laptops and tablet PCs. Firewalls, host-based IDS/IPS, antivirus programs, and patch management programs can be extended to mobile devices to provide host-based protection; however, not every device will support these traditional security controls. Thus, additional security controls will need to be implemented to fill in the gaps where appropriate.

Secure Syncing

Certain portable devices, such as PMPs and cell phones, can be synced to a host system, which allows the user to manage the device via a management interface. As discussed earlier in the section "Why Attack the Handset" of this chapter, the device can be used to attack the host system or conversely, an infected host system could attack the mobile device. In order to carry out this type of attack, an attacker would exploit the automatic actions that occur when a user connects their mobile device to the host system. In Microsoft Windows, these features are called Autorun and AutoPlay. Autorun starts programs automatically when a device is inserted or connected to a computer, and AutoPlay can be configured to automatically select a program to start based on the type of media inserted into a computer.[13] Generally speaking, these features should be disabled in a corporate environment, especially considering Conficker and other malware have recently exploited this functionality.[14] Instructions on how to disable these features are provided in the following Microsoft Support article: http://support.microsoft.com/kb/967715. Other operating systems have similar features and should be disabled if at all possible.

Disk Encryption

Sensitive data, such as credit card numbers or social security numbers, should always be encrypted. It does not matter if the data is sitting on a desktop inside the utility company's office building or on a laptop that is taken home by an employee every night. However, since mobile devices tend to leave the physical security barrier of most office buildings, the need for full disk encryption on mobile devices is greater. Unfortunately, there is not a full disk encryption solution for every mobile device. So, mitigating controls such as requiring any sensitive data to be stored inside an encrypted file container or prohibiting the storage of sensitive data on mobile devices, such as cell phones, can help mitigate the risk.

TOOLS

TrueCrypt (www.truecrypt.org) is a free, open-source tool that provides support for encrypted file containers, partitions, and full disk encryption. The full disk encryption is only supported on Windows; however, the other features are supported on Windows, Mac OS X, and numerous Linux distributions.

Screen Lock

Most mobile devices can be configured to require a username/password combination, PIN, or some other form of authentication credential to be entered before granting access to the device. From an organization's perspective, mobile devices should follow the policy that covers screen locking. Typically, this policy dictates the following:

- Screen is automatically locked after 5 to 15 minutes of inactivity
- User will manually lock the screen when leaving the system unattended
- Authentication is required to unlock the screen

For mobile devices, such as cell phones and handhelds, these settings are usually configurable; however, the management is more difficult. On Windows systems, group policy can be set to automatically configure the device to implement the screen lock policy. Other devices do not support centralized management and must be configured manually. Additionally, some devices are hard to type on and will frustrate users who need to type in a complex password. Thus, some users may choose a password/PIN that is easier to type … and of course easier to guess.

Screen locking on mobile devices can be defeated by attackers in multiple ways and should not be solely relied upon. For example, some touch screen mobile devices pose a risk due to finger smudges. Oil from the user's fingers can create a fingerprint trail that reveals the characters of the password/PIN. An attacker could then try the different combinations of the characters to unlock the device. If the device is configured to use an unlock pattern to unlock the screen, then it can be even easier for the attacker. For example, Android phones can be configured to use an unlock screen that displays nine dots on the touch screen, and the user must draw a pattern with the nine dots to unlock the screen.[15] The pattern can be any series of lines with the following constraints:

- Between four and nine dots must be used
- A dot can only be used once
- A dot cannot be crossed, unless it has already been used
- A line between dots can be horizontal, vertical, diagonal, or similar to the way a chess knight piece moves[16]

Figures 10.1 through 10.4 provide example unlock patterns that comply with these constraints. Considering users routinely choose poor passwords, the pattern chosen will most likely be a simple line that has two distinct end points, which is illustrated in Figures 10.1 and 10.2. For the examples in this chapter, consider an end point to be a point with only one line connected to it. In Figure 10.1, the points in the upper left corner and the bottom right corner would be considered end points. If drawing the pattern leaves a smudge trail on the screen, there will only be two possible unlock combinations for Figures 10.1 and 10.2. In this scenario, the attacker simply needs to start at one end point and traverse the line, or smudge trail, until they have reached the other end point. If that does not work, the only other possible combination is to follow the line in reverse. Even if the user chooses a more complex pattern, such as those

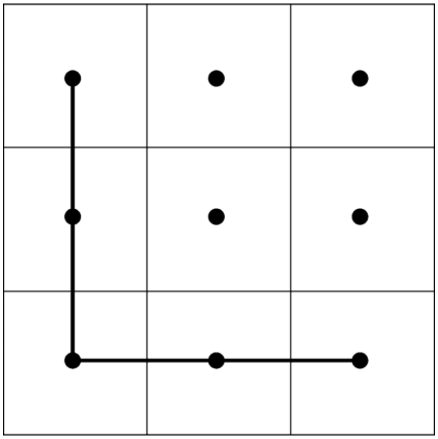

FIGURE 10.1

Simple android unlock pattern.

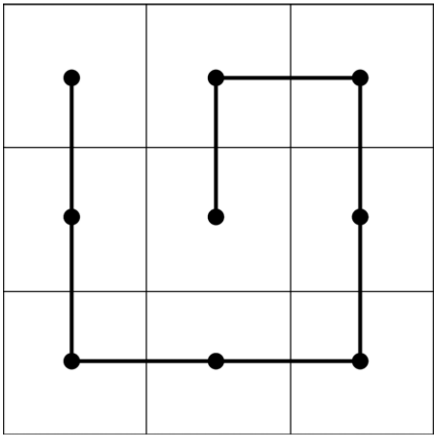

FIGURE 10.2

Simple android unlock pattern using all nine points.

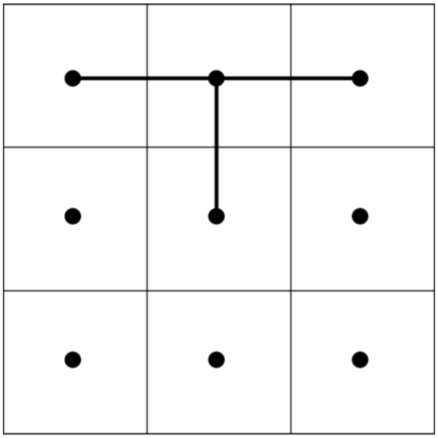

FIGURE 10.3

More complex android unlock pattern with three potential end points.

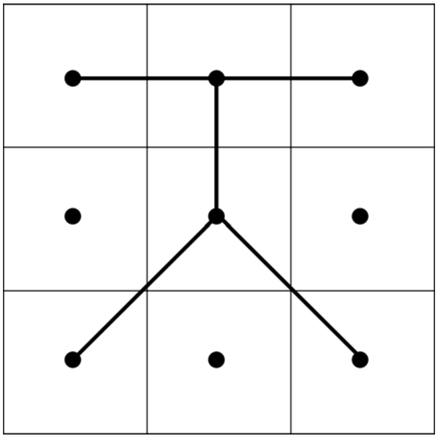

FIGURE 10.4

More complex android unlock pattern with four potential end points.

illustrated in Figures 10.3 and 10.4, the number of distinct end points will not increase significantly. For example, the pattern in Figure 10.3 has three potential end points and has only six possible unlock patterns. Thus, drawing the pattern may leave a smudge trail that has only a few different combinations. The pattern in Figure 10.4 is a little more complex with 4 potential end points and 24 possible unlock patterns; however, performing 24 combinations would still not take a very long time.

Wiping the Device

Mitigating the attacks described in the section "Screen Lock" can be done by securely erasing all the data on the mobile device after a defined number of unsuccessful authentication attempts. This number is usually set to be between five and 10. Setting this number to less than five would be ill advised since users tend to mistype their password on the smaller keyboards of mobile devices.

Additionally, the ability to remotely wipe the mobile device will help mitigate the risk of theft. Several commercial solutions exist that will enable administrators to remotely wipe the mobile device after it has been reported lost or stolen. Additionally, some of these solutions also provide a phone-home function that attempts to alert the owners of the device's location via coordinates provided by a GPS signal.

WARNING

Wiping the device can help mitigate the risk of several attack vectors, but should only be considered a last resort security control. In the situation where the device is remotely wiped, the attacker will have time between when they stole the device and when the owner reports the device stolen to obtain the sensitive information stored on the host. Thus, wiping the mobile device may only remove that copy of the sensitive information. Each security control discussed in this chapter should be part of a defense in depth approach to security.

Recovery

Some attackers may not be interested accessing the sensitive information and only want to cause damage by taking their anger out on your property. Additionally, the attacker may be the mobile device owner's dog that causes the owner to trip and smash the mobile device. In a more common scenario though, utility company maintenance workers may need to work during harsh weather storms that batter the mobile devices with rain and debris.

Regardless of the scenario, there will be instances where the data on the mobile device becomes inaccessible. Recovering the data may be possible, but if the organization or consumer was maintaining backups of the mobile devices, they may not need to go through the often-painstaking process of trying to recover data from a corrupt storage disk. For cell phones and PMPs, most syncing applications provide the functionality to backup the mobile device to the host system. Users who take advantage of this functionality will be grateful when their mobile device's data is deleted or becomes inaccessible.

Forensics

Forensics is a difficult process on even nonmobile systems, but one thing that can help an investigation process is by keeping detailed logs. Access to sensitive data should be logged to assist the forensics process in the case of an incident.

Although not every mobile device will have logging capabilities, those that do should be enabled. For more information on forensics for mobile devices, please see NIST SP 800-101: Guidelines on Cell Phone Forensics.

Education

Users are almost always the weakest link in security, which is why educating mobile device users is so important. Mobile devices are stolen frequently when an owner leaves them in the car to buy a cup of coffee or go to the bathroom. Thus, regular training on at least an annual basis should be done to reinforce the importance of security awareness, which includes the following:

- Never leave the mobile device out of reach or sight in an untrusted environment
- Manually lock the screen, logout, or shut down the mobile device
- Do not visit untrusted Web sites on the mobile device or open messages from unknown e-mail addresses or phone numbers
- Backup the device regularly

SECURE MOBILE APPLICATIONS

In most cases, mobile applications are developed to be an interface to the standard application. The mobile application sits between the standard application and the mobile client, and it handles communications between the mobile client and the standard application. There are, of course, exceptions where a mobile application is developed independently, but the security controls will remain the same.

Mobile Application Security Controls

One of the biggest mistakes that mobile application developers make is assuming that only mobile devices will interact with the mobile application. Assuming the mobile application server is network accessible, any system with access to the network will be able to attack that application server. So, for example, let's consider the Chevy Volt OnStar mobile applications again. Users will be able to use an iPhone, BlackBerry, Droid, or most mobile Web browsers to remotely control certain functions in their Chevy Volt car from anywhere.[9] In order to provide this level of access, the mobile application server(s) will be Internet accessible. Thus, any system with an Internet connection will be able to attack that application server.

Mobile applications will need to be able to defend against traditional application attacks, including those described in Chapter 7, "Attacking the Utility Companies." The following resources provide detailed information regarding how to develop secure applications:

- Open Web Application Security Project (OWASP) – www.owasp.org
- Web Application Security Consortium (WASC) – www.webappsec.org

WARNING

The OWASP Top 10 (www.owasp.org/index.php/Category:OWASP_Top_Ten_Project) can be used as a good starting point to understand the types of attacks that the mobile applications will face. Many organizations use the OWASP Top 10 as the only criteria for assessing their application security posture and only concern themselves with the ten items in the list. The OWASP Top 10 is intended to provide awareness on the top 10 Web application security flaws,[17] thus it is not intended to be a comprehensive list.

Encryption

When developing mobile applications, it is tempting to offload encryption to the network provider. So, for example, if the developers intend to support only cell phones, they may make the justification that the cellular network will encrypt the data in transit, thus implying that SSL will be a waste of resources. However, Chapter 7, "Attacking the Utility Companies," discussed attacks against encryption used in GSM networks. Additionally, a large number of cell phones now include Wi-Fi radios, so there is no assurance that cell phones will even be using the cellular networks to communicate with the mobile application server. Making these types of assumptions can lead to critical vulnerabilities in applications.

SUMMARY

Mobile applications and devices will be used extensively to increase the functionality and reach of smart grids, which is also why they will introduce greater risk to smart grids. They provide a conduit for attackers to easily bypass an organization's physical and virtual perimeter security controls. Additionally, they will provide attackers with another attack vector to compromise user's security.

Due to the extensive variety of platforms, applications, and inherent mobility, mobile devices are more difficult to manage and oftentimes lack centralized management capabilities. This leads to a greater reliance on individual users to implement secure practices, such as screen locking and regularly making backups. However, the applications will face familiar attacks that organizations should already be prepared to face. By implementing the security controls discussed in this chapter, organizations can greatly reduce the risk that mobile applications and devices present to their security posture.

Endnotes

1. comScore. comScore reports February 2010 U.S. Mobile subscriber market share [document on the Internet]. www.comscore.com/Press_Events/Press_Releases/2010/4/comScore_Reports_February_2010_U.S._Mobile_Subscriber_Market_Share; 2010 [accessed 11.04.10].

2. Raywood D. Secure Computing Magazine. Online game disguised as space invaders hits mac users with trojan [document on the Internet]. www.securecomputing.net.au/News/159811,online-game-disguised-as-space-invaders-hits-mac-users-with-trojan.aspx; 2009 [accessed 19.05.10].
3. VanHemert K. Gizmodo. Apple says no more titillating apps, period [document on the Internet]. www.gizmodo.com.au/2010/02/apple-says-no-more-titillating-apps-period/; 2010 [accessed 12.04.10].
4. NIST. FIPS PUB 186-3 Digital Signature Standard (DSS) [document on the Internet]. http://csrc.nist.gov/publications/fips/fips186-3/fips_186-3.pdf; 2009 [accessed 20.05.2010].
5. Goodin D. The Register. iPhone, IE, Firefox, Safari get stomped at hacker contest [document on the Internet]. www.theregister.co.uk/2010/03/25/pwn2own_2010_day_one/; 2010 [accessed 20.05.10].
6. Garfinkel S. CSO. Attack of the iPods! [document on the Internet]. www.csoonline.com/article/220868/Attack_of_the_iPods_; 2006 [accessed 12.04.10].
7. Zorz Z. HelpNet Security. Mariposa bot distributed by Vodafone's infected phone [document on the Internet]. www.net-security.org/secworld.php?id=8991; 2010 [accessed 20.05.10].
8. Mills E. CNET. Researchers attack my iPhone via SMS [document on the Internet]. http://news.cnet.com/8301-27080_3-10299378-245.html; 2010 [accessed 29.07.10].
9. General Motors. Chevrolet and onstar give volt owners 24/7 connection and control via wireless smartphone application [document on the Internet]. http://gm-volt.com/p/OnStar%20Mobile%20release%20CES.pdf; 2010 [accessed 05.04.10].
10. Mearian L. Computerworld. Survey: 40% of hard drives bought on eBay hold personal, corporate data [document on the Internet]. www.computerworld.com/s/article/9127717/Survey_40_of_hard_drives_bought_on_eBay_hold_personal_corporate_data; 2009 [accessed 20.05.10].
11. Llewellyn G. The Independent. US missile data found on eBay hard drive [document on the Internet]. www.independent.co.uk/news/world/americas/us-missile-data-found-on-ebay-hard-drive-1680529.html; 2009 [accessed 20.05.10].
12. Soper M. MaximumPC. Leave No Trace: how to completely erase your hard drives, ssds and thumb drives [document on the Internet]. www.maximumpc.com/article/howtos/how_complete_destroy_your_data; 2010 [accessed 12.04.10].
13. Microsoft. How to disable the AutoRun functionality in Windows [document on the Internet]. http://support.microsoft.com/kb/967715; 2009 [accessed 13.04.10].
14. Microsoft. Autorun changes in Windows 7 [document on the Internet]. http://blogs.technet.com/srd/archive/2009/04/28/autorun-changes-in-windows-7.aspx; 2009 [accessed 13.04.10].
15. Arrington M. TechCrunch. Android's Login Is Cool, But Is It Secure? [document on the Internet]. http://techcrunch.com/2008/10/12/androids-login-is-cool-but-is-it-secure/; 2008 [accessed 13.04.10].
16. Beust C. Otaku, Cedric's weblog. Android's Locking Pattern [document on the Internet]. http://beust.com/weblog2/archives/000497.html; 2008 [accessed 20.05.10].
17. OWASP. OWASP Top Ten Project [document on the Internet]. www.owasp.org/index.php/Category:OWASP_Top_Ten_Project; 2010 [accessed 13.04.10].