Making Everything Easier!™

Palo Alto Networks® Edition

Mobile Security

FOR DUMMIES®

A Wiley Brand

Learn to:

- Identify threats to mobile devices
- Manage and protect mobile devices
- Control sensitive data on mobile devices

Brought to you by

paloalto networks®

Lawrence C. Miller, CISSP

About Palo Alto Networks

Palo Alto Networks is leading a new era in cybersecurity by protecting thousands of enterprise, government, and service provider networks from cyber threats. Unlike fragmented legacy products, its security platform safely enables business operations and delivers protection based on what matters most in today's dynamic computing environments: applications, users, and content.

Find out more at www.paloaltonetworks.com.

Mobile Security

FOR DUMMIES®

A Wiley Brand

Palo Alto Networks Edition

by Lawrence C. Miller, CISSP

FOR DUMMIES®

A Wiley Brand

Mobile Security For Dummies®, Palo Alto Networks Edition
Published by
John Wiley & Sons, Inc.
111 River St.
Hoboken, NJ 07030-5774
www.wiley.com

ISBN 978-1-118-91426-7 (pbk); ISBN 978-1-118-91528-8 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Publisher's Acknowledgments

Some of the people who helped bring this book to market include the following:

Senior Project Editor: Zoë Wykes

Acquisitions Editor: Amy Fandrei

Editorial Manager: Rev Mengle

Business Development Representative: Karen Hattan

Custom Publishing Project Specialist: Michael Sullivan

Project Coordinator: Melissa Cossell

Special Help from Palo Alto Networks: Brian Tokuyoshi

Table of Contents

Introduction

The explosive growth of mobile devices in the workplace creates new opportunities for business innovation, while also introducing new risks. The challenge enterprises face today is how to make mobile devices and apps secure for business use. Traditional approaches to security have not addressed the specific security challenges of the mobile enterprise. These approaches include the following:

- **Blocking mobile devices and apps:** Some organizations try to use blocking technologies in an attempt to insulate themselves from the risks that come with mobile computing. However, employees want to use their mobile devices at work, and will find ways to use them without the company's approval.
- **Attempting to protect mobile devices and apps with existing security products:** Some organizations hope that their existing security measures will protect their mobile environment. But this won't provide satisfactory results because traditional network and endpoint security measures aren't optimized for mobile use cases and may not provide adequate protection against mobile threats.
- **Using basic mobile security tools:** Not all mobile security tools are the same, and the limitations aren't always apparent at first. Mobile security tools for basic use cases (such as ActiveSync for e-mail) don't necessarily provide the essential protection for other apps and data. As organizations adopt more sophisticated mobile use cases, their security requirements will change.

Organizations must find ways to safely enable mobile computing, while satisfying end-user requirements for ease of use and platform choice. A new approach to mobile security is necessary in order to fully realize all the benefits that mobility can provide to the organization.

About This Book

This book provides an in-depth examination of mobile security challenges and solutions, including mobile Wi-Fi, mobile devices and apps, mobile threats and malware, advanced persistent threats (APTs), and mobile security solutions.

Foolish Assumptions

It's been said that most assumptions have outlived their uselessness, but I assume a few things nonetheless.

First, I assume you have a functional understanding of basic information security concepts, practices, and technologies. You should be as comfortable talking about firewalls and malware as you are talking about sports statistics or fine wines. As such, this book is written primarily for technical readers.

I also assume that you have a smartphone and/or tablet of your own and have casually noticed that smartphones and tablets are everywhere in your organization. Therefore, I don't spend much time in the pages that follow talking about the rise of mobility and the business case for mobile security. No tiptoeing at the shallow end of the pool here, you're going to dive right into the technical deep end!

Finally, I assume that you're part of an enterprise business or other large organization with mobile security challenges. Perhaps you're trying to implement a Bring Your Own Device (BYOD) or Corporate-Owned, Personally-Enabled (COPE) mobile strategy. You may just need to understand the different mobile threats and challenges that exist today. Or you may need to evaluate mobile security solutions and their implications for your business. If any or all of these scenarios apply to you, keep reading because this is the book for you!

Icons Used in This Book

Throughout this book, you occasionally see special icons that call attention to important information. You won't see any smiley faces winking at you or any other cute little emoticons, but you'll definitely want to take note! Here's what you can expect.

This icon points out information that may well be worth committing to your nonvolatile memory, your gray matter, or your noggin — along with anniversaries and birthdays!

You won't find a map of the human genome here (or maybe you will, hmm), but if you seek to attain the seventh level of NERD-vana, perk up! This icon explains the jargon beneath the jargon and is the stuff legends — well, nerds — are made of.

Thank you for reading, hope you enjoy the book, please take care of your writers. Seriously, this icon points out helpful suggestions and useful nuggets of information.

Proceed at your own risk . . . well, okay — it's actually nothing *that* hazardous. These helpful alerts offer practical advice to help you avoid making potentially costly mistakes.

Beyond the Book

Although this book is chock-full of information, there's only so much I can cover in 72 pages. So, if you find yourself at the end of this book thinking, "Gosh, this was an amazing book, where can I learn more?," just go to `www.paloaltonetworks.com`. There, you can find out more about mobile security solutions from Palo Alto Networks.

You'll also want to check out `researchcenter.paloaltonetworks.com` for access to relevant reports, white papers, articles, and blog posts about mobile security.

Where to Go from Here

With our apologies to Lewis Carroll, Alice, and the Cheshire Cat:

"Would you tell me, please, which way I ought to go from here?"

"That depends a good deal on where you want to get to," said the Cat — err, the Dummies Man.

"I don't much care where . . . ," said Alice.

"Then it doesn't matter which way you go!"

That's certainly true of *Mobile Security For Dummies* which, like *Alice in Wonderland,* is destined to become a timeless classic.

If you don't know where you're going, any chapter will get you there — but Chapter 1 might be a good place to start. However, if you see a particular topic that piques your interest, feel free to jump ahead to that chapter. Each chapter is individually wrapped (but not packaged for individual sale) and written to stand on its own, so feel free to start reading anywhere and skip around. Read this book in any order that suits you (although I don't recommend upside down or backward). I promise, you won't get lost falling down the rabbit hole.

Chapter 1

Wi-Fi Security and Mobile Devices

In This Chapter

- Understanding Wi-Fi security weaknesses
- Taking a look at man-in-the-middle attacks

In this chapter, you find out about the different Wi-Fi security protocols, some of their inherent weaknesses, and various man-in-the-middle attacks against mobile devices on Wi-Fi networks.

How Attackers Get on a Protected Network

Wireless networks first started appearing in the late 1990s as laptop computers started becoming more commonplace. With the explosive growth of mobile devices over the past decade — specifically, smartphones and tablets — wireless networks are now everywhere. Whether you're in an office, hotel, airport, school, or coffee shop, you're likely in range of a wireless network somewhere.

Of course, as a security professional, your first concern when trying to get connected is, "How secure is this wireless network?" But what about your users? Their decision process is much more methodical and deliberate. In order of priority, your users' decision process is more like this:

- "Is Wi-Fi available?"
- "Is it free?"

- "How strong is the signal?"
- "How fast is the network?"
- "Do I have to ask someone for the password?"

Need a hotel room? "Do you prefer a king or queen size bed?" Either. "Non-smoking or smoking?" Non. "Complimentary breakfast until 10 A.M." Okay. "Free Wi-Fi." Great! How early can I check in?

The unfortunate truth is that wireless connectivity is more about convenience than security for the average user. Thus, the challenge is not only to secure your corporate wireless networks but also to protect the mobile devices that your organization's employees use to perform work and access corporate data — no matter where they are or whose network they're on. In other words, you need to protect your users from themselves!

Wireless security begins with authentication. If you can't control who has access to your wireless network, then you can't protect your network. So I start with an in-depth look at the two most common Wi-Fi encryption methods in use today — WEP and WPA.

WEP "security"

The Wired Equivalent Privacy (WEP) protocol was the wireless industry's first attempt at security. It was an attempt at best. WEP isn't effective for establishing a secure wireless network. Period.

Unfortunately, you don't know how your organization's employees are setting up their wireless home networks or, for that matter, how coffee shops, hotels, and other businesses that offer free Wi-Fi to their guests are. WEP should be all but extinct given its known and well-publicized weaknesses. Unfortunately, not only is WEP still prevalent in home networks, but a 2013 *Information Week* survey found that 15 percent of business networks continue to use WEP as well.

WEP emerged as part of the IEEE (Institute of Electrical and Electronics Engineers) 802.11 wireless network standard in 1999. As its name falsely implies, WEP was intended to provide data confidentiality equivalent to the security of a wired

network. However, it wasn't long before a team of researchers identified a serious flaw in WEP. Subsequent flaws (as well as poor implementations of WEP in various products) led to even more devastating attacks — and improvements to the tools that are used in the attacks.

Standard 64-bit WEP (also known as WEP-40) uses a 40-bit key consisting of 10 hexadecimal characters with a 24-bit initialization vector (IV); 128-bit WEP (also known as WEP-104) uses a 104-bit key consisting of 26 hexadecimal characters with a 24-bit IV.

Hexadecimal (also known as base 16 or hex) characters include 0–9, representing values zero through 9, and A-F, representing values 10 through 15, respectively. Each hexadecimal character represents four binary digits (bits).

One of the critical weaknesses in WEP lies in how it handles the initialization vector (IV), which is basically a random number used in conjunction with an encryption key to protect data confidentiality for WEP's RC4 (Rivest Cipher 4) stream cipher. In WEP, the IV is a 24-bit key that is transmitted in the clear (or unencrypted). With a 24-bit key, it becomes impossible to generate unique values after sending 2^24 (or 16,777,216) packets, and the IVs will repeat. In a secure environment, the key should be replaced before exhausting the IVs, but you have no way to automate the process in WEP. Thus, given enough traffic, IV collisions will occur, which, in conjunction with other techniques, make it possible for an attacker to deduce the WEP key.

Deducing the WEP key can be done by passively monitoring the network traffic using a wireless network card in promiscuous mode. Passive monitoring leaves no indication that the wireless access point (AP) is under attack as the attacker is doing nothing more than making copies of the packets on the network. The downside to passive monitoring for an attacker is that it requires a legitimate user on the network to generate a sufficient amount of traffic to deduce the key. From the perspective of targeting a specific person, the first attacks against WEP were theoretically sound yet still somewhat impractical.

Using active techniques with an attacker sending traffic directly to the target AP, the time to break into a WEP-enabled network drops from days to minutes, thus making it very practical to

conduct a targeted attack. An attacker could literally park in front of someone's house and gain access to their WEP-enabled wireless network with relative ease.

In the past, cracking a WEP key required several tools for reconnaissance, packet collection, packet injection, and cracking the key itself. Today, the tools found in common penetration testing kits are now fully automated, with GUIs (graphical user interfaces) that make cracking a WEP key as easy as point and click.

Attacks on WEP don't depend on having a massive amount of computing power and aren't greatly affected by the size of the encryption key. The attack isn't dependent on how complex the original passphrase is either. It's simply a matter of being able to collect enough traffic.

Once it became apparent that WEP had critical, unfixable security flaws, efforts took place immediately to develop a successor. Because a replacement for WEP was urgently needed, an interim standard, Wi-Fi Protected Access (WPA) was published in 2003. WPA was further refined as WPA2 in 2004, and WEP was then deprecated as a Wi-Fi security standard.

WPA/WPA2 security

Given the inherent security problems associated with WEP, the security industry needed a replacement for WEP — and they needed it fast. WPA was published as an interim standard in 2003, quickly followed by WPA2 in 2004. WPA/WPA2 contains improvements to protect against the inherent flaws in WEP. These improvements included changes to the encryption in order to avoid many of the problems that plagued WEP.

The 802.11g wireless standard allows WEP, WPA, or WPA2 to be configured on a Wi-Fi network. The 802.11n standard requires WPA2 in order to take advantage of wireless network speeds above 54 mbps.

WPA2 can be implemented in different ways. WPA2-Enterprise, also known as WPA2-802.1x mode, uses the Extensible Authentication Protocol (EAP) and a RADIUS server for authentication. Numerous EAP types are also available for use in WPA2-Enterprise.

However, the use of a pre-shared key (PSK) is by far the most common, particularly in homes, small businesses, and guest Wi-Fi networks. WPA2-PSK can be implemented with just the AP and the client, requiring neither a third-party 802.1x authentication server nor individual user accounts.

For the most part, the Wi-Fi networks that your users connect to outside the office will likely be using WPA2-PSK.

WPA2-PSK supports 256-bit keys, which require 64 hexadecimal characters. Because it's impractical for users to enter a 64 hexadecimal character key, WPA2 includes a function that generates a 256-bit key based on a much shorter passphrase created by the administrator of the Wi-Fi network and the Service Set Identifier (SSID) of the AP used as a salt for the one-way hash function.

A *one-way hash function* is a mathematical function that creates a unique representation (a hash value) of a larger set of data in a manner that is easy to compute in one direction. The hash function can't recover the original text from the hash value. However, an attacker could attempt to guess what the original text was and see if it produces a matching hash value.

In order to make it harder to guess the original text, hash functions use a "salt." A *salt* is randomly generated data that is used to extend the input provided for a one-way hash function in cryptography. The same original text hashed with different salts results in different hash values. In WPA2, the name of the SSID is used for the salt. An easy way to make your Wi-Fi security stronger (and make rainbow table attacks impractical) is to simply change your SSID to something that isn't common or easily guessed.

A rainbow table is a pre-computed table used to find the original value of a cryptographic hash function.

To execute an attack on a WPA2 passphrase, an attacker needs to be able to test a large number of passphrase candidates. So, while WPA2 remains cryptographically secure (namely the key isn't recoverable by simply observing the traffic as with WEP), methods do exist to test passphrases offline by gathering the handshake packets between the AP and a legitimate user.

In order to collect the necessary packets to crack a WPA2 passphrase, an attacker could passively gather traffic when a legitimate user joins the network. This method requires time however, because the attacker has no idea when someone will join the network.

For an impatient attacker, the solution is to employ an active attack. As long as a legitimate user is already online, the attacker can kick the user's client device off the AP with forged deauthentication packets. After getting knocked off, the client device will automatically attempt to reconnect, thus providing the attacker with the handshake packets needed for offline passphrase analysis. Thus, unlike WEP, attacks on WPA2 can be done without spending a significant amount of time in the proximity of the target network. Once the handshake packets have been gathered, an attacker can continue his work elsewhere.

With the handshake packets in hand, what's next? The attacker must still recover (or find) the passphrase itself, which requires the following:

- **A test to check millions of potential passphrases until it finds the correct passphrase:** In order to avoid detection, an attacker can't use the actual target because the victim would be able to see this attack activity. The alternative is to use an offline method of testing, using the handshake packets.
- **A methodology to guess passphrases:** The worst-case scenario is to brute force the passphrase, that is, trying every possible combination of numbers and characters until a correct value is found. This effort can produce a correct result given enough time and computing power. However, it's much faster to take educated guesses without having to resort to brute force. By using educated guesses on possible passphrase candidates, the attacker can scan a much shorter list.

This basic process for recovering Wi-Fi passphrases is similar to cracking user passwords. In the early days of password cracking, an attacker might have knowledge of a target system's one-way hash function and a list of the system's user password hash values. However, the attacker had no way to decrypt the password, because the original text isn't recoverable from a hash — hence, a *one-way* hash!

Most cryptographic algorithms used today, including those used in one-way hash functions, are well known and published. Good cryptographic algorithms are peer reviewed to look for potential flaws. The security of the environment depends on protecting the hash values and choosing strong passwords and passphrases.

By hashing a list of words with the same one-way hash function (a dictionary attack), an attacker can then compare the resulting hash values with the hash values stored for the various user accounts on the system. So, although the password itself isn't decrypted, it is possible to find a given input that produces a given result — a password match. With the addition of more computing power, an attacker could try longer word lists and a greater number of variations of each word.

The process for attacking WPA2 passphrases is similar, but with some important distinctions that make WPA2 passphrases on a wireless network even less secure than user passwords.

Wireless Man-in-the-Middle Attacks

Instead of breaking into a wireless network, what happens when the attacker just tricks the user into connecting to a network that the attacker controls? Here's a look at what happens when an attacker inserts himself between the user and the network services he's trying to reach. These techniques are part of a larger set of attacks known as *man-in-the-middle*. In this particular scenario, it refers to what happens when the attacker controls the infrastructure that the victim is using.

Evil Twin

Perhaps the easiest way for an attacker to find a victim to exploit is to set up a wireless access point that serves as a bridge to a real network. Remember, convenience trumps all else, so an attacker can inevitably bait a few victims with "free Wi-Fi."

The main problem with this approach is that it requires someone to stumble on the access point and connect. The attacker can't be picky about who will be the victim because it depends on the victim making the connection.

A slight variation on this approach is to use a more specific name that mimics a real access point normally found at a particular location — the Evil Twin. For example, if your local airport provides Wi-Fi service and calls it "Airport Wi-Fi," the attacker might create an access point with the same name using an access point that has two radios. The average user cannot easily discern when she's connected to the real access point or the fake one, so this approach would catch a greater number of users than trying to attract victims at random. Still, the user has to select the network so a bit of chance is involved in trying to reach a particular target.

The main limitation of the Evil Twin attack is that the attacker can't choose its victim. In a crowded location, the attacker will be able to get a large number of people connecting to the wireless network, which is fine if the goal is to just grab account names and passwords. However, it's not an effective approach if the goal is to target specific employees at a specific company.

Jasager

To understand a more targeted approach than the Evil Twin attack, think about what happens when you bring your wireless device back to a location that you've previously visited. For example, when you bring your laptop home, you don't have to choose which access point to use because your device remembers the details of wireless networks to which it has previously connected. The same goes for visiting the office or your favorite coffee shop.

Your mobile device detects when it's within the proximity of a previously known wireless network by sending a beacon out to see if a preferred network is within range. Under normal conditions, when a wireless device sends out a beacon, the nonmatching access points ignore it. The beacon goes unanswered, except when it comes within the proximity of the preferred network.

The Jasager attack takes a more active approach toward beacon requests. Jasager, German for "the Yes man," responds to all beacon requests, thus taking a very permissive approach toward who can connect. The user doesn't have to manually choose the attacker's access point. Instead, the attacker pretends to be whatever access point the user normally connects to (see Figure 1-1). Instead of trying to get victims to connect at random, now the attacker simply needs to be within the proximity of his target.

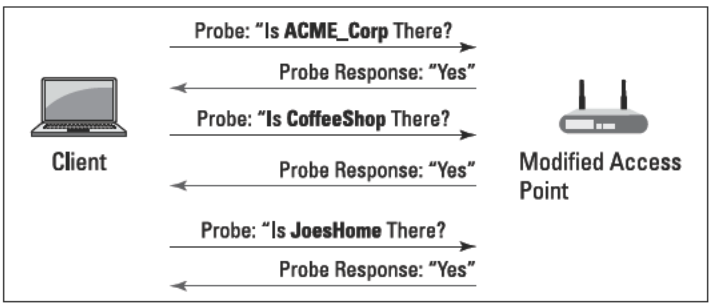


Figure 1-1: Jasager pretends to be whatever access point is requested by the client's beacon.

This process intercepts the communication from laptops, mobile phones, and tablets. Many (if not most) 3G/4G/LTE mobile devices automatically switch to Wi-Fi when they recognize that they're near a network that they know.

Using the same method to capture WPA2 handshake packets (see the earlier section in this chapter, "WPA/WPA2 security"), an attacker can kick users off a Wi-Fi network by using forged deauthentication packets; when the users reconnect, they'll unwittingly connect to the modified access point. Unlike the Evil Twin attack (see the preceding section), the attacker doesn't have to just keep his fingers crossed hoping that a victim will connect to the modified access point; with this approach, everyone who's in the vicinity will automatically connect and become a potential victim.

Jasager runs on any number of devices, but perhaps one of the most effective ways to employ it is with the Pineapple access point. The Pineapple is simply an access point with modified firmware that embeds a number of tools for wireless penetration testing. It also has a number of accessories, such as support for 3G USB cards to provide network connectivity

when it is otherwise unavailable at the target location and battery packs to operate as a standalone unit. The Pineapple is also easily concealed because it can be disguised within any number of housings typically found plugged in at the office.

Once the attacker has the victim connected to a malicious access point, the man-in-the-middle attack can proceed, and the attacker not only can observe and capture network traffic, but can modify it as well.

SSLstrip

After a user connects to a Wi-Fi network that's been compromised — or to an attacker's Wi-Fi network masquerading as a legitimate network — the attacker can control the content that the victim sees. The attacker simply intercepts the victim's web traffic, redirects the victim's browser to a web server that he controls, and serves up whatever content the attacker desires.

Take a look at how a man-in-the middle attack can be used to steal a victim's online banking or corporate e-mail account credentials. Normally, this type of traffic would be considered safe because the web page typically uses Secure Sockets Layer (SSL) encryption. Of course, all that your average users know is that if a padlock appears somewhere in the address bar of their browser, it must be secure, right?

But the padlock appears differently, and in different locations in different browsers. How does the padlock appear in Internet Explorer? What about Mozilla Firefox, Google Chrome, and Apple Safari? It appears differently on different smartphones and tablets too. It's no wonder that typical end-users — even many security professionals — can be easily tricked, which is where SSLstrip comes into play.

SSLstrip strips SSL encryption from a "secure" session. When a user connected to a compromised Wi-Fi network attempts to initiate an SSL section, the modified access point intercepts the SSL request (see Figure 1-2). The modified access point then completes the SSL session on behalf of the victim's device. Then, the SSL tunnel between the victim's device and the legitimate secure web server is actually terminated — and decrypted — on the modified access point, allowing the attacker to see the victim's credentials, and other sensitive information, in clear text.

With SSLstrip, the modified access point completes the charade by displaying a fake padlock in the victim's web browser. Web pages have the capability to display a small icon called a *favicon* next to a website address in the browser's address bar. SSLstrip replaces the favicon with a padlock that looks like SSL to an unsuspecting user.

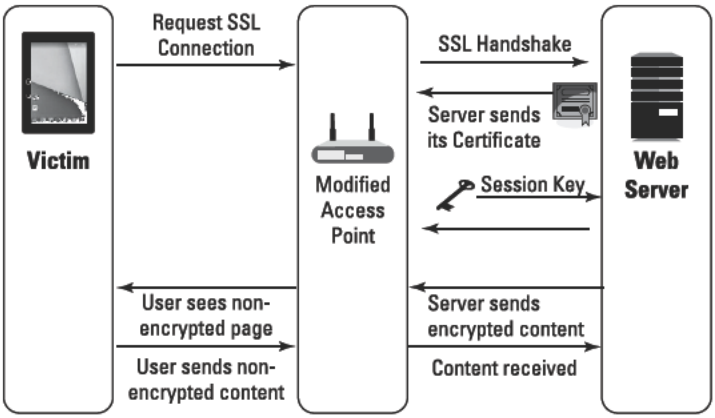


Figure 1-2: Man-in-the-middle with SSLstrip.

Man-in-the-middle malware distribution and phishing

With a man-in-the-middle exploit in place on a Wi-Fi network, an attacker can serve up practically any content. If a user attempts to download a legitimate file, then the attacker can send mobile malware instead. When a user attempts to visit a legitimate web page, the attacker can alter the content to exploit a vulnerability that exists in the device's browser, allowing the attacker to further escalate an attack (you find out more about the attack life cycle in Chapter 4). And with legitimate corporate e-mail addresses and financial account information, an attacker can create a very targeted and convincing phishing attack to trick even more users on a network into disclosing sensitive information.

In Chapter 2, you take a closer look at mobile apps and their associated security challenges.

Chapter 2

Apps Behaving Badly

In This Chapter

- Recognizing mobile app security issues
- Comparing mobile security and permission models
- Understanding why third-party libraries can be risky
- Protecting mobile data in the cloud

This chapter explores the Bring Your Own Device and consumerization trends, and how these trends have evolved the mobile app landscape. This chapter also talks about the differences in security between the Google Android and Apple iOS mobile operating systems, how features on your smartphone — such as your camera and GPS (global positioning system) — can be misused by mobile malware, and why mobile data in the cloud may put your organization at risk.

Security Challenges with Mobile Apps

As with Wi-Fi networks (see Chapter 1), when it comes to mobile app development, convenience — and time-to-market — trump security far too often. In many cases, basic data security features that you would expect in an app don't work adequately, or may not even be present at all. And there's no easy way for the average user to tell if an app is secure or not.

An app doesn't have to be designed for bad purposes to do bad things. In many cases, an app may be poorly designed, or it may be using functions or reusing code that come from a third party, such as an outsourced development team or

a third-party library. In any case, it's important to understand that for the average user, managing apps and how they behave is hard, and stopping poorly designed or otherwise risky apps from doing dangerous things is hard as well.

Over the past decade, the application landscape has changed dramatically. Business applications in the workplace have been joined by a multitude of personal and consumer-oriented applications. This convergence of corporate infrastructures and personal technologies is being driven by two trends — Bring Your Own Device (BYOD) and consumerization.

BYOD refers to the use of personally owned mobile technology — predominantly smartphones and tablets — in the workplace for both work-related and personal uses.

Consumerization is a related trend that is further driving BYOD into the workplace. The process of consumerization occurs as users increasingly find personal technology and applications that are more powerful or capable, more convenient, less expensive, quicker to install, and easier to use than corporate IT solutions. These user-centric "lifestyle" applications and technologies enable individuals to improve their personal efficiency, handle their nonwork affairs, and maintain online personas, among other things. Common examples include Google Docs, cloud-based file-syncing environments, instant messaging applications, and web-based e-mail. More often than not, the same applications used for social interaction are being used for work-related purposes. And as the boundary between work and their personal lives becomes less distinct, users are practically demanding that these same tools be available to them in their workplace.

Unsure of how to leverage the BYOD and consumerization trends in their business processes, many organizations either implicitly allow these personal devices and applications by simply ignoring their use in the workplace, or explicitly prohibit their use but are then unable to effectively enforce such policies with traditional firewalls and security technologies.

Neither of these two approaches is ideal, and both incur inherent risks for the organization. In addition to lost productivity, adverse issues for the organization include

- Creating a subculture of back-channel or underground workflow processes that are critical to the business's operations, but are known only to a few users and fully dependent on personal technologies and applications.
- Introducing new risks to the entire networking and computing infrastructure, because of the presence of unknown, and therefore unaddressed and unpatched, vulnerabilities, as well as threats that target normal application and user behavior — whether a vulnerability exists in the application or not.
- Being exposed to noncompliance penalties for organizations that are subject to regulatory requirements, such as the U.S. Health Insurance Portability and Accountability Act (HIPAA) and the Payment Card Industry's Data Security Standard (PCI DSS).

The bottom line is that if you try to block the use of mobile devices in your work environment, you may find out exactly how creative your employees can be as they find new and innovative ways to use their personal mobile devices at work anyway. But instead of having their mobile devices safely enabled for work use, they may do things that put your business information, networks, and applications at risk.

Corporate Owned, Personally Enabled (COPE) is an alternative mobile strategy to BYOD. Organizations that implement a COPE policy provide their employees with a mobile device and apps that the employee chooses from an approved list. Additionally, the employee is permitted to use the device for personal purposes, such as installing personal productivity apps, playing games, online banking, and personal calls.

The challenge is not only the growing diversity of the applications, but also the inability to clearly and consistently classify them as good or bad. Although many are clearly good (low risk, high reward), and others are clearly bad (high risk, low reward), most are somewhere in between. Moreover, the end of the spectrum that these applications fall on can vary from one scenario to the next and from user to user or from session to session.

For example, using a social networking application to share product documentation with a prospective customer would be good (medium risk, high reward), while using the same

application to forward details of an upcoming release to a Friends list that includes employees of a competitor would be not so good (high risk, no reward).

To effectively address mobile security challenges, enterprise network security solutions must be able to

- Classify different types of applications.
- Identify users regardless of what device they're using or where they're using it.
- Account for contextual variables surrounding the use of an application.
- Enforce granular mobile security policies based on the preceding factors.

Applications are evasive

Although classifying different types of applications sounds simple, it really isn't — for a number of reasons. In order to maximize their accessibility and use, many applications — including mobile apps — are designed from the outset to circumvent traditional firewalls by dynamically adjusting how they communicate. For the end-user, this means that an application can be used from anywhere, at any time, but for the security administrator, it's a major headache. Common tactics include

- **Port hopping,** where ports/protocols are randomly shifted over the course of a session.
- **Use of non-standard ports,** such as running Yahoo! Messenger over TCP port 80 (HTTP) instead of the standard TCP port for Yahoo! Messenger (5050).
- **Tunneling within commonly used services,** such as when peer-to-peer (P2P) file sharing or an instant messenger (IM) client like Meebo is running over HTTP.
- **Hiding within SSL encryption,** which masks the application traffic, for example, over TCP port 443 (HTTPS).

Many mobile apps take the appearance of a client/server application, but the underlying network communication uses web technologies. These apps access cloud-based services, use URL addresses to communicate with servers, and can be just as dynamic and elusive as applications found on a desktop or laptop computer.

Many companies use content (or URL) filtering software to control what websites their employees can access at work. For example, marketing employees may be the only employees permitted to access Facebook, and all employees are restricted from gaming and pornographic websites. But many content filtering solutions can't accurately classify the different URLs that are used by mobile apps and don't properly identify mobile device users.

Finally, many new business applications also use these same techniques to facilitate ease of operation while minimizing disruptions for customers, partners, and the organization's own security and operations departments. For example, remote procedure calls (RPC) and Sharepoint use port hopping because it is critical to how the protocol or application (respectively) functions, rather than as a means to evade detection or enhance accessibility.

The result is that HTTP and HTTPS now account for approximately two-thirds of all enterprise traffic. This isn't a problem per se, but it does exacerbate an inherent weakness of traditional security infrastructure. Specifically, the wide variety of higher-order applications riding on top of HTTP and HTTPS — whether or not they actually serve a legitimate business purpose — are practically indistinguishable for older network security solutions. The negative impact of organizations further losing control over their network communications is clear and underlines the fact that the application landscape has evolved dramatically.

Threats are coming along for the ride

The increasing prevalence of mobile application attacks is yet another disturbing trend. The two primary ways to make bad things happen on a mobile device is for an attacker to either infect a mobile device with malware, or send bad input (an exploit, discussed in Chapter 3) to take advantage of a vulnerability that exists in an app that is already installed on a mobile device.

Threat developers often take advantage of the same evasion methods (described in the previous section) that application developers utilize to promote ease of use and widespread adoption, such as tunneling within applications. The evasion techniques built in to these and many other modern applications are being leveraged to provide threats with free passage into enterprise networks. It is no surprise, therefore, that greater than 80 percent of all new malware and intrusion attempts are exploiting weaknesses in applications, as opposed to weaknesses in networking components and services. Together with the implicit trust that users place in their applications, all these factors combine to create a perfect storm.

The motivation for hackers has also shifted — from gaining notoriety to making money. The name of the game today is information theft. Consequently, it is no longer in a hacker's best interests to devise threats that are "noisy." To be successful, a thief must avoid detection, and by making as little noise as possible, the thief can often persist for longer periods of time.

For those hackers who favor speed over sophistication — speed of initial threat generation, speed of modification, and speed of propagation — the goal is to develop, launch, and quickly spread new threats immediately on the heels of the disclosure of a new vulnerability. The resulting zero-day and near-zero-day exploits then have an increased likelihood of success because reactive countermeasures, such as patching and those tools that rely on threat signatures (such as antivirus software and intrusion prevention), are unable to keep up — at least during the early phases of a new attack.

This speed-based approach is facilitated in large part by the widespread availability of threat development websites, toolkits, and frameworks. Unfortunately, another byproduct of these resources is the capability to easily and rapidly convert known threats into unknown threats — at least from the perspective of signature-based countermeasures. This transformation can be accomplished by

- Making a minor tweak to the source code of a threat.
- Using a crypter to change the signature of a threat file.
- Using an app binder to mask the underlying malicious capabilities of an otherwise benign app.
- Adding entirely new propagation and exploit mechanisms to a threat.

A *crypter* is software that is used to change the signature for a piece of malware without changing its functionality. The malware can be reused over and over again. An *app binder* is software used to hide an app with malicious functions within an otherwise harmless-appearing app.

Many of today's threats are built to run covertly on networks and systems, quietly collecting sensitive or personal data and going undetected for as long as possible. This approach helps to preserve the value of the stolen data and enables repeated use of the same exploits and attack vectors. As a result, threats have become increasingly sophisticated. Rootkits, for example, have become more prevalent. These kernel-level exploits effectively mask the presence of other types of malware, enabling them to persistently pursue the nefarious tasks they were designed to accomplish (such as intercepting keystrokes).

The increasing speed and sophistication of threats emphasize the need for proactive countermeasures with extensive visibility and control at the application layer of the network computing stack.

iOS Security Model

iOS is the mobile operating system used on Apple iPhones and iPads. According to IDC, 21 percent of smartphones run on iOS. Malware authors have typically targeted Google Android devices (discussed in the next section) rather than iOS because Android devices are more popular and because the iOS security model is more restrictive than Android's model.

Apple evaluates apps before making them available through the App Store. This process includes analyzing various criteria in the app, such as its functionality and stability, as well as looking at basic security risks within the app. It's not a perfect system, but it's generally effective because it serves as a restricted gateway for downloading and updating apps on iOS devices. You have only a few ways to install an iOS app.

- **App Store:** You can purchase apps and download them from the official Apple App Store on iTunes. This is the primary method for installing almost all apps on an iOS device.

- **Enterprise App Store:** Some companies have a fixed set of apps that they need their users to use, and these can be distributed through a private app store that's available only to their employees.
- **Ad Hoc provisioning profile:** You can deploy an app for testing to a limited group of devices using an Ad Hoc provisioning profile. This is typically used for testing code under development.
- **Jailbreaking:** Jailbreaking an iOS device allows the user to install apps from sources other than the app store. This is different than rooting an Android device, which is used for customizing or changing the operating system. (I explain jailbreaking and rooting in Chapter 3.)

To date, more than 1 million apps are available in the Apple App Store, and more than 60 billion app downloads have been made.

Android Security Model

Google Android is the open source mobile operating system used on Android smartphones and tablets. Android is based on a Linux kernel and is currently installed on the majority of mobile devices worldwide. Most mobile malware targets Android devices for several reasons:

- **Popularity:** Android has the vast majority of the worldwide market share for mobile devices, with no indication of slowing down. Malware authors typically focus on the most popular mobile devices in order to cast the widest net.
- **Open platform:** Unlike iOS devices, you can load software on an Android device in a number of ways. These include alternative app stores and apps that can be sideloaded from websites, e-mail, file-sharing apps, and memory cards.
- **No jailbreaking needed:** Loading apps from unofficial sources doesn't require jailbreaking an Android device like iOS does. A simple setting change on the device is all that's needed to permit the installation of apps from other locations.

Comparing Permission Models

You will find several stark differences in the way that iOS and Android devices handle app permissions. For example, here's a look at a hypothetical app that reviews local restaurants.

On an iOS device, such as an iPhone, you would typically download the app from the Apple App Store. The first time you run the app, it may ask if it can access the GPS feature of your iPhone so that it can locate restaurants that are near your iPhone (and presumably near you). It makes sense that this app would want to know where you are so that it can make relevant recommendations for a restaurant nearby. However, many users may be uncomfortable with an app essentially tracking their movements and may instead prefer to manually enter a location each time they want to use the app.

On iOS devices, users can easily see (and change) which apps have what permissions by going to the Privacy menu under Settings.

The same app on an Android device operates very differently. First, you can download the app from a number of official and unofficial app stores (discussed in the previous section). When you install the Android app, you're presented with a list of features that the app needs permission to access. Using the same restaurant app example, your restaurant app may ask for permission to access GPS, as well as SMS and your contacts. The user either has to select them all or choose to decline installing the app. Do users actually read all of the screens and menu options when an app is being installed, or do they just keep clicking Next until the installation is complete?

Under the Android permissions model, a user can see the permission requests upfront during the app installation, but it's really up to the app (not the operating system) to control whether you can disable or change the permissions after the app has been installed.

Google Android 4.3 had a feature known as App Ops that provided more granular control of individual app permissions. This feature was removed in version 4.4 because, according to Google, it could potentially break apps.

The following section takes a look at how third-party libraries can dramatically change what an app can do with the permissions it has been given.

App Assembly and Third-Party Libraries

Building a mobile app used to require a lot of time and effort — and a fair amount of skill. Developers needed to create everything from scratch. Besides the application's primary functions, the developer needed to create application interfaces to the various device functions, establish communication between the app and backend servers, and integrate the app with other network services. As a result, many apps today are assembled using shared libraries that come from third parties to perform common functions.

However, one consequence of the trend to use third-party libraries is the inadvertent introduction of often aggressive and sometimes malicious library functions into otherwise benign apps. The apps themselves may not have been designed for malicious purposes, but the libraries used to provide supporting functions can have unintended and potentially harmful results. For example, many developers monetize their mobile apps through pay-per-click advertising banners that are built into the apps. Rather than developing the advertising functionality themselves, app developers will use a third-party library to handle this aspect of the app. This provides a potential vulnerability within the app if the third-party library is not securely developed, or even has malicious code intentionally written into its source code.

Mobile Data in the Cloud

When using a mobile device, knowing where data actually resides can be very difficult for mobile users. Is the data stored on the device itself, a corporate server, a remote desktop computer, or in the cloud?

Sometimes the user is fully aware of where the data is being stored and whether or not it is being shared. For example, many people use file sharing SaaS (software as a service)

applications, which provide a location to make their files accessible no matter where they go. End users are often the ones who are in control of the file-sharing capabilities and will sometimes comingle their business and personal files within their file-sharing service. This creates an issue for organizations that are struggling with BYOD and regulatory challenges — namely, what can you do about employees choosing their own file-sharing services and placing your sensitive data in locations that you can't control? File-sharing apps are just one example of applications that may need to be managed more stringently on mobile devices that have access to corporate information.

But file sharing isn't the only way that data is transferred to the cloud. Many apps send data to the cloud without the user really knowing or understanding what's happening.

For example, many foreign-language character sets use cloud-based dictionaries to map non-Roman alphabet characters to their phonetic Roman equivalents on a mobile device's keyboard. Essentially, every keystroke from the mobile device is sent to the cloud to be translated. This data is therefore susceptible to compromise, both during transmission and if it's being stored in the cloud.

Don't trust the app to have the security that you need for communication and data protection. This is especially important as organizations adopt greater numbers of apps for business use, because it is difficult, if not impossible, to vet each one. The organization must take steps to provide security for the mobile device in order to make the apps safe for business use.

Chapter 3

Mobile Exploits and Malware

In This Chapter

- Recognizing unique mobile challenges
- Understanding mobile exploits
- Looking at jailbreaking and rooting
- Using an alternative app store
- Going mobile with malware

Mobile devices have many characteristics and aspects that make it challenging to keep them free of threats such as mobile exploits and malware. This chapter explains some of these characteristics and challenges, as well as the exploits and malware threats that take advantage of these unique mobile issues.

Vulnerabilities on Mobile Devices

Just like any software, finding bugs in a mobile operating system and its applications is inevitable. Some of these bugs are just annoying, but others are far more serious. A bug that can be manipulated to compromise the security of the device is a *vulnerability.* An attack used against a vulnerability is known as an *exploit.*

Staying up to date with patches to fix vulnerabilities is one of the most basic protections against exploits. However, patching a mobile device isn't always easy.

The process for patching and updating mobile devices is considerably different from traditional endpoints, such as desktop and laptop PCs. Traditional endpoints that are managed are typically updated with system management software. Unmanaged endpoints are typically updated directly by the software publisher over the Internet.

However, mobile device software is not as consistently or frequently updated by the different device manufacturers and mobile application developers as traditional software is.

For example, the responsibility for patching Android devices is layered. Google may issue a patch for an affected version of the Android operating system and make it widely available to all of the different manufacturers of Android mobile devices. However, the patches aren't automatically distributed to the devices; it's up to the manufacturers to distribute the patches for their particular devices and models. Many, if not most, device manufacturers use a customized version of the Android operating system so that they can personalize the user interface and add new features to differentiate their products in the market. As a result, patching the mobile operating system isn't as straightforward as simply rolling up all of the changes that come from Google into a software update. The changes must be validated and tested by each individual device manufacturer. Typically, device manufacturers will only validate and test on devices that are currently supported by the manufacturer, so not all mobile devices in use are necessarily patched or updated to the latest version of the operating system.

At the same time, keeping your mobile devices up to date with the most current software updates is just as important as it is for traditional endpoints. Software updates and patches are necessary to fix software bugs and vulnerabilities that can be exploited by an attacker.

An exploit targets an unpatched vulnerability in the operating system. Unlike malware, an exploit is an attack on an application or operating system that you already have installed. Using an unpatched mobile device exposes it to greater risk than a patched one.

Exploiting Mobile Vulnerabilities

An attacker can exploit a vulnerability in a mobile application or operating system for a number of purposes, such as stealing information or loading malware on the device.

What are some examples of vulnerabilities and how have they been used against mobile devices? Here's a closer look at a few recent examples of mobile vulnerabilities and exploits.

Android Master Key

The Android Master Key exploit is an example of one of the tricks used by attackers to modify otherwise benign apps. The exploit plays upon a relatively simple trick: The attacker simply puts a duplicate file inside the app that he wants to modify. The exploit is possible because affected versions of Android allow files with the same name to exist in the same directory structure. The operating system doesn't expect to see multiple copies of the same file, and so it confuses which one is the real version.

When the vulnerability is exploited, it provides a way for an attacker to modify an app without making it obvious that it's been changed. It's like being able to remove the tamper-resistant packaging on a bottle of aspirin, change the contents, and reseal it without any indication that it's been modified.

Normally, changing an app would modify the hash. However, with the Android Master Key exploit, an attacker can create modified versions of the app without breaking the tamper-resistant seal. As a result, the attacker can take an app and package it together with malware, and then distribute the app without making the app's distribution files look suspicious.

A *hash* provides a means to see whether the content of a file has changed. Changing a file changes the hash.

HeartBleed

HeartBleed is a security vulnerability that was discovered in early April 2014. HeartBleed severely impacts enterprise web servers running affected versions of OpenSSL, such as public

facing websites hosted across the Internet. The bug allows an attacker to exploit the heartbeat functionality of OpenSSL by sending a malformed heartbeat request to a vulnerable server. The server responds with random 64 kilobyte blocks of data from the server's memory that may be completely useless to the attacker — or, in a worst-case scenario, it may contain individual user names and passwords, security certificates, cryptography keys, and other sensitive data (see Figure 3-1).

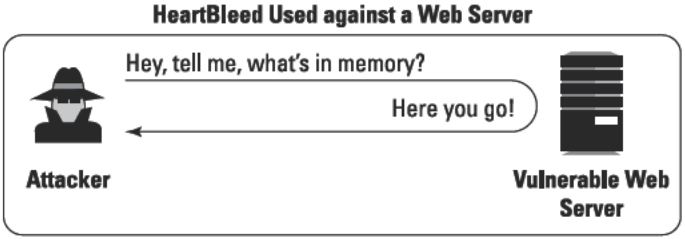


Figure 3-1: The HeartBleed attack.

With widespread media coverage focusing on the exploitation of websites, one might be misled into thinking that HeartBleed is solely a security problem for web servers. However, OpenSSL is widely used in a variety of products, not just web servers. In fact, OpenSSL is also used as the cryptographic library for clients connecting to a web server, which introduces another set of security issues: Clients that are using affected versions of OpenSSL are vulnerable to a Reverse HeartBleed attack.

In a Reverse HeartBleed attack, the attacker sets up a malicious web server to steal information from vulnerable mobile devices (see Figure 3-2).

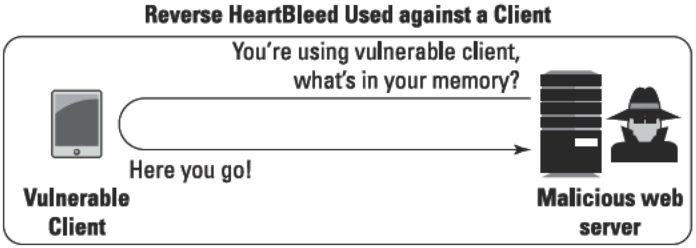


Figure 3-2: The Reverse HeartBleed attack.

With respect to mobile devices, the good news is that HeartBleed doesn't affect iOS itself, and doesn't affect the majority of Android versions. The bad news, however, is that Android 4.1.1 is vulnerable and affects anywhere between 10 and 34 percent of Android mobile devices in use today.

HeartBleed exposes a set of mobile device security challenges that many organizations had not previously considered. Mobile devices need access to applications, but some applications can malfunction and cause harm. The organization must find ways to enable access to the application while stopping the exploit from reaching these devices. Organizations must be able to identify the affected devices and provide pre-patch protection for affected devices that might otherwise never be patched.

Jailbreaking and Rooting

Some people want to have more control over their mobile devices than what they can get right out of the box. They want to customize functions that are not otherwise customizable, get certain apps without paying for them, and make their devices do things they weren't necessarily designed to do. In order to do this, some mobile device owners modify the mobile operating system on their devices to disable certain security features and restrictions.

Jailbreaking refers to the process of modifying a device so that it can use multiple app stores. This term typically refers to Apple iOS mobile devices, which by default use the Apple App Store as their primary source for loading applications. Jailbreaking requires the owner to exploit the device so that it can perform functions that are not otherwise permitted. A jailbroken mobile device can use unofficial alternative app stores to load applications and can provide owners with additional privilege levels that allow them to perform modifications to the device, file system, and operating system that are not otherwise possible.

Jailbreaking introduces a new attack vector for malware, for two primary reasons. First, a jailbroken device's security model is compromised, thus allowing programs that run on it to have greater control over the device's operating conditions. The second reason is that malware authors can more easily distribute malware through alternative app stores than they could through the official Apple App Store.

Android provides a more open environment for installing apps than iOS and doesn't need to be jailbroken. Apps from unofficial sources can be installed simply by changing a setting on the device. This allows device owners to install apps not only from Google Play but also from alternative official app stores and underground app stores, as well as from software packages distributed via e-mail and the Internet. These conditions make it easier for users to switch app stores, but it also makes it easier to load potentially malicious apps from other sources as well, such as from app stores that distribute pirated mobile apps.

While Android does provide a more open environment for the installation of apps than iOS, it is still a controlled environment that restricts direct modification of various operating system functions. The process of *rooting* an Android device gives users escalated privileges, thus permitting the user to make changes to the operating system, or even replace it entirely with a new operating system build. For some users, this provides greater flexibility to customize the device's functions, and it also provides a route to install software distributions that are otherwise not available for their device from the manufacturer. But this additional access to otherwise protected functions of the device can also make malware much more potent. So, while rooting an Android device is not a prerequisite for installing malware, it can create an environment that makes the malware more capable of performing malicious activity.

In corporate environments, rooting or jailbreaking a mobile device typically isn't necessary. Thus many organizations establish baseline security policies that forbid the use of devices that have had their security features modified in such a manner.

Alternative App Stores

Alternative app stores provide other sources for mobile apps that may not otherwise be available from an official app store. Sometimes these app stores are designed to service particular needs, such as app stores that cater to specific markets.

Some app stores provide software that is categorically in a gray area with regard to legitimate uses, such as game console emulation software.

In some cases, an app developer or author may not like the official app store's terms and conditions, and chooses to distribute its app through alternative stores.

Some app stores also focus on black market software. These app stores take commercial apps and redistribute — or pirate — them through unofficial app stores.

In all of these cases, the scrutiny given to the apps that are distributed in the app store varies considerably. In some cases, the curation process is very meticulous. This is particularly true for apps distributed through Apple's App Store. But for other app stores, little or no review process, particularly for black market apps, may exist.

Using an alternative app store with lax policies can introduce risk because there may not be any checks to see if the app contains hidden functions. This is particularly true for app stores that provide pirated software because the software may have been modified to include malware.

The State of Mobile Malware

Years ago, many security professionals viewed mobile malware as more of a theoretical construct rather than a real-world issue. While acknowledging that mobile malware was a concern, other security concerns took precedence. The issue of mobile malware was something to be addressed at some point down the road. With the explosive growth in the use of mobile devices over the past decade, there's no question that we are already much farther down the road!

Mobile malware today is a very real threat. New malicious mobile code is being created every week, and literally hundreds of mobile malware variants are released into the wild every year. While mobile malware is very similar to other malware in its objectives and techniques, some different mobile security risks do exist, particularly with regard to infection and propagation.

Mobile malware installation tricks

For years, cybercriminals have been picking apart and repackaging widely downloaded Android mobile apps with malicious code. These trends originated through the distribution of

pirated apps embedded with malware, but have evolved to other methods as well. These infected Android APKs (application package files) are then uploaded to app stores in an effort to trick users into downloading and installing the malware onto their mobile devices.

The name recognition of these apps alone can lull users into a false sense of security and into mistakenly thinking they are downloading the official apps when, in fact, the apps have already silently been compromised. For example, attackers are taking apps that are popular on other platforms and creating a build that's not officially available on Android. But these software packages may have adware/malware bundled in as well!

The trend of infecting mobile devices with malware through apps highlights the need to place greater scrutiny on the apps used on enterprise networks and mobile devices, as well as the source (the app stores) of these apps. This scrutiny must include analysis of app behaviors to identify any latent risky or malicious functions within the apps. Enterprise security practitioners need security tools that provide both the ability to identify malware and to detect devices that have these mobile malware or infected apps installed.

Finally, many infected apps will use the organization's network to download other pieces of code and to communicate with the attacker. Organizations must set policies that can break the threat lifecycle by dismantling the malware's capability to establish covert communication with hostile parties.

Malware and reverse billing

Text messaging is the single most popular application for mobile devices. It's also popular with attackers because it provides a way to make money.

Normal text messaging via SMS (Short Message Service) is relatively straightforward. If you send a text message, your carrier may charge you a fee or require that you have a texting plan. A recipient may also have to pay a nominal fee to receive the message.

Mobile content providers want to charge money for the delivery of their information, and they do this through "premium SMS" services. The recipient that wants this information pays to

receive it, so in this context, it is considered reverse billing. The recipient pays for the content in her monthly bill, and the premium goes to the mobile content provider — the message sender.

Malware authors use premium SMS services to make money on malware-infected mobile devices by tricking mobile users into subscribing to premium SMS services.

How much money does the attacker try to get from the user? An attacker will typically infect large numbers of mobile devices but only charge a small amount to each user's bill. Most mobile device owners would notice an additional $100 or $1,000 on their monthly cellular service bill and dispute the charge, but they are much less likely to notice an extra $5 or $10. By keeping the charged amount small and sustainable, the attacker can stay under the radar of device owners who typically only take a cursory glance at their monthly bill. Just like the techniques used in malware, the attackers try to maintain an undetected, sustained attack over time rather than to do anything that draws the spotlight onto their activities.

Chapter 4

Advanced Persistent Threats Go Mobile

In This Chapter

- Getting to know today's hackers
- Going after specific targets
- Understanding the basics of advanced persistent threats
- Walking through the lifecycle of an attack
- Recognizing the role of malware
- Going mobile with APTs

Advanced persistent threats (APTs) are reshaping the threat landscape and forcing enterprises to reassess their security strategies. APTs have outpaced traditional anti-malware and network security technologies and, in the process, have established a foothold within the enterprise that criminal organizations and nation-states can use to steal data and attack sensitive information assets.

The rapid rise of mobility and Bring Your Own Device (BYOD) trends in the enterprise adds yet another dimension — and another attack vector — for cybercriminals to potentially exploit. Will your users bring an infected device to work and put your entire organization at risk?

In this chapter, I talk about this new class of threat, the role of advanced malware in the APT lifecycle, and how mobility creates new security challenges for the enterprise.

The Changing Face of Hackers

Hackers have evolved from the prototypical whiz kid — sequestered in a basement, motivated by notoriety, and fueled by too much carbonated caffeine — into bona fide cybercriminals, often motivated by significant financial gain and sponsored by nation-states, criminal organizations, or radical political groups. The sophisticated hacker is no longer operating as a lone ranger, but rather

- Has far more resources available to facilitate an attack.
- Has greater technical depth and focus.
- Is supported by an organization or nation-state.
- Operates as part of a team rather than as an individual.
- Is well funded.
- Is better organized.

Why does this matter? Because to individual hackers, hacking tends to be an intellectual exercise. The ability to pull off the attack is their motivation. They're not necessarily out to profit from their ability to break into a corporate network.

On the other hand, for rogue nation-states or criminal organizations, the ability to hack is the means to get what they're really after. They want illicit access to networks, applications, and data, and they're looking to use that information to provide an advantage for their own interests or to damage the hacked company.

Additionally, criminal organizations and nation-states have far greater financial resources than independent hackers. Many criminal hacking operations that have been discovered were complete with the standard appearance of a legitimate business, including offices, receptionists, and of course cubicles full of dutiful hackers.

These are criminal enterprises in the truest sense, and their reach extends far beyond that of an individual hacker. Not only do we face more sophisticated adversaries today, but the types of information of value to them are continually expanding as well. These groups can do interesting things with the most seemingly innocuous bits of information.

Targeting the Victim

Information on the target organization and its personnel is gathered using any publicly available information, for example, from the company's website or employees' social networks (such as Facebook and Twitter). Learning about their targets' connections provides attackers with insight about their victims and provides metadata about where their targets are located and how they're organized.

Reconnaissance may also involve identifying places that employees frequently visit. For instance, even though a company may have a secure office building, it may not have a cafeteria. Employees may spend lunch breaks meeting at public restaurants and talking about their work, providing an opportunity to compromise or steal mobile devices and laptops.

If attackers want to go after the most sensitive information in the company, such as undisclosed financial reports or upcoming acquisitions, they might target the company's executives. It's not difficult to identify the executive team members and the board of directors because this information is almost always available on the company's website. It's also not too difficult to locate where people live, based on public records. Using this information, an attacker may choose to attack the home wireless network of the executive team members, which is likely to be far less secure than the corporate network (see Chapter 1).

Reconnaissance also may include traditional social engineering tactics to learn user passwords and other information about the target network. The attacker may probe the network for vulnerabilities that could provide points of entry as well.

The ABCs of APTs

APTs are the culmination of advanced malware, modern attack methods, and vast computing, networking, financial, and human resources. The garden variety type of attack tries to get a large number of victims without really discriminating about who they'll catch in their net. APTs go after a very specific target.

APTs employ a patient, multi-step attack strategy. This strategy blends exploits, malware, and evasion into a coordinated attack that specifically targets an entire network entity, such as an enterprise or government organization.

The U.S. National Institute of Standards and Technology (NIST) defines an APT as:

"An adversary that possesses sophisticated levels of expertise and significant resources which allow it to create opportunities to achieve its objectives by using multiple attack vectors (for example cyber, physical, and deception). These objectives typically include establishing and extending footholds within the information technology infrastructure of the targeted organizations for purposes of

- *Exfiltrating information*
- *Undermining or impeding critical aspects of a mission, program, or organization*
- *Positioning itself to carry out these objectives in the future*

The advanced persistent threat pursues its objectives repeatedly over an extended period of time, adapts to defenders' efforts to resist it, and is determined to maintain the level of interaction needed to execute its objectives."

Key characteristics of an APT that distinguish it from other attacks or exploits include

- **Advanced:** APTs use advanced, often customized malware to infiltrate a specific target network and evade detection.
- **Persistent:** APTs occur over long periods of time, often months or years, and employ various tools and tactics to ensure the attack can be sustained, even if certain elements of the APT are discovered.
- **Threat:** APTs are specifically designed for government, military, or corporate espionage purposes that include surveillance, information theft, and sabotage. APTs are carried out by highly trained human operators with significant abilities and resources (often with the backing of nation-states and their various covert "3-letter" agencies).

The Lifecycle of a Modern Attack

The lifecycle of a modern attack has multiple phases and uses many tools and techniques (see Figure 4-1), which include

- Infection
- Persistence
- Communication
- Command and Control

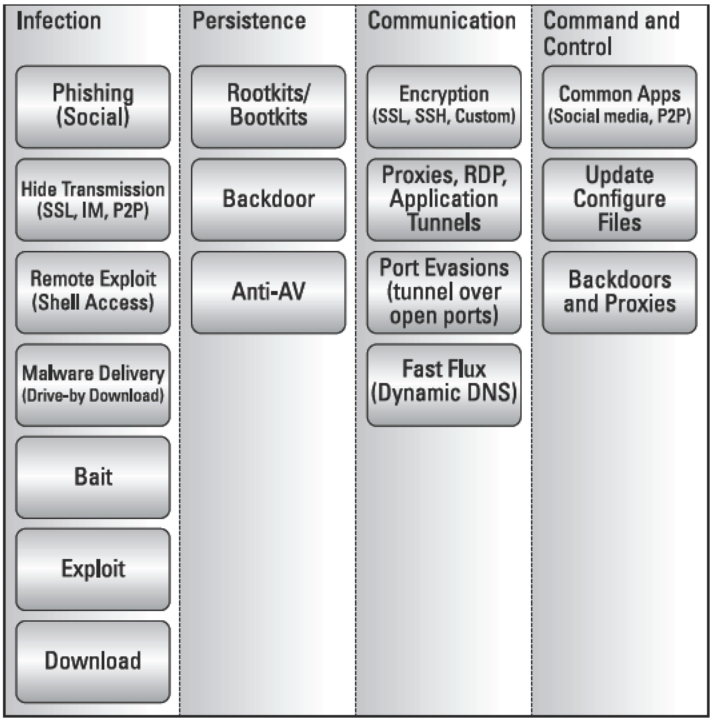


Figure 4-1: Key tools and techniques used in the modern attack lifecycle.

Infection

The infection (or intrusion) phase of an APT, like most cyber-attacks today, begins with an exploit against a vulnerability. A *vulnerability* is a programming error in an application that you have already installed or a part of the operating system itself. The attacker exploits the vulnerability by providing bad input to the application, causing the application to behave in weird, unexpected ways.

In many cases, the exploit will have a social aspect, such as getting individuals to click on a link in a phishing e-mail, luring them to a social networking site, or sending them to a web page with an infected image.

The trend today is that threats don't necessarily come as an executable attachment in an e-mail or by downloading a file from the web. A link is all that's required. This is why social media, webmail, message boards, and microblogging platforms such as Twitter are rapidly becoming favorite infection vectors for attackers.

Once attackers successfully exploit an application, they can take a number of possible actions next. One of the most common actions is to establish shell access.

Shells used for malware typically provide a command line interface that allows an attacker to type commands on the remote device. Almost anything is possible from the shell, allowing the attacker to change settings, modify programs, and install malware.

With shell access, the attacker can take complete control of the device and can deliver just about any payload.

One example of an exploit that produces a shell on mobile devices is the Webview exploit. This particular attack affects Android version 4.2 and earlier, and it simply relies on getting a user to visit a web page that has the exploit code embedded within it. Once the user visits the page, a shell opens on the attacker's computer, providing full access to the mobile device.

So how does an attacker get users to visit a malicious web page? By e-mailing the user a link, posting something to a

message board or social media, or even embedding the link in a malicious QR (Quick Response) code.

In order to infect the target device, the malware author needs to bypass security controls. The traditional way of detecting malware at scale is to identify the malware's unique file signature (typically done by creating a hash of the malware).

Malware authors who are trying to avoid detection have an easy way of defeating signature-based detection. Instead of sending malware that has already been seen in the wild, they will often develop new and unique malware that is customized specifically for the target. This method effectively subverts the malware sample-gathering process, thus keeping the sample from being analyzed and identified as malware.

Another common way to avoid detection is to infect an individual's device over a connection that traditional security solutions can't see into, such as an encrypted channel. Attack transmissions are often obscured in SSL-encrypted (Secure Sockets Layer) traffic or other proprietary encryption used in P2P (peer-to-peer) networking applications, for example.

Malware enables the next phases of the APT lifecycle (command/control and escalation, discussed in the following sections) by acting as a control point inside the network, finding other assets on the internal network, escalating privileges on infected devices, and/or creating unauthorized administrative accounts — just to name a few tactics.

The key is that instead of malware and network exploits being separate disciplines as they were in the past, they are now integrated into an ongoing process. Malware, which is increasingly customized to avoid detection, provides a remote attacker with a mechanism of persistence, and the network enables the malware to adapt and react to the environment it has infected.

Persistence

Once a target machine is infected, the attacker needs to ensure survivability of the malware so that the malware can continue to operate in the event that the machine is rebooted. In the realm of desktop operating systems, the malware may infect parts of the operating system, insert itself into the startup functions, or employ the use of rootkits and bootkits.

A *rootkit* is a piece of malware that hides in the upper functions of the operating system, thus making it inconspicuous or undetectable to the average user. A rootkit can employ masking techniques to hide itself from typical system monitoring tools, such as when reviewing a list of running processes.

A *bootkit* modifies the boot record of the hard disk, thus providing a way to insert malicious code before other parts of the operating system are loaded. Bootkits provide the attacker with access to very low-level functions of the compromised host, making it possible to perform tasks such as recovering encryption keys used for full-disk encryption.

Backdoors enable an attacker to bypass normal authentication procedures to gain access to a compromised system. Backdoors are often installed as a failover in case other malware is detected and removed from the system.

The element of persistence is somewhat different with mobile devices.

First of all, mobile malware can often persist with little difficulty, because many organizations lack the capability to detect mobile malware. Once the malware is present on the device, the organization may not have any method for identifying the devices that have malware or processes for removing it.

Mobile malware typically takes the form of an app that's installed on the mobile device. This allows the malware to stay resident on the device even if the device is rebooted.

Finally, mobile devices actually aren't stopped and restarted very often. Unless the battery dies, most people leave their mobile devices running at all times. So even if the exploit/malware didn't have any hooks into the startup process, it's still highly likely that installed malware can survive for lengthy periods of time.

Communication

Communication is fundamental to a successful attack because it allows the attacker to provide further instruction to the malware. This communication may include the establishment of a command and control channel and stolen data that is

extracted from a target system or network. Attack communications must be stealthy and cannot raise any suspicion on the network. Such traffic is usually obfuscated or hidden through techniques that include

- Encryption with SSL, SSH (Secure Shell), or some other custom application. Proprietary encryption is also commonly used, as well as tunneling fake application traffic within other applications or protocols.
- Port evasion using network anonymizers or port hopping to tunnel over open ports.
- The generation of unknown traffic; some command and control traffic hides in plain sight. Instead of creating a lot of noise that's certain to draw attention, the malware will communicate quietly and generate very small amounts of traffic, making it hard to flag and detect.
- Fast Flux (or Dynamic DNS) to proxy through multiple infected hosts, reroute traffic, and make it extremely difficult to determine where the traffic is really going.

Command and control

Command and control rides on top of the communication platform that is established but is really about making sure that the malware or attack is controllable, manageable, and updateable.

Command and control is often accomplished through common applications including webmail, social media, P2P networks, blogs, and message boards. Command-and-control traffic doesn't stand out or raise suspicion, is often encrypted, and frequently makes use of backdoors and proxies.

The Central Role of Malware

Attack techniques have also evolved and malware now plays a central role in the hacker's arsenal and in the lifecycle of an attack. Attackers have developed new methods for delivering malware (such as drive-by downloads), hiding malware communications (with encryption), and avoiding traditional signature-based detection.

Modern malware is somewhat like the pea in a shell game. A street con running a shell game on the sidewalk lures the mark (or victim) into trying to follow the pea, when actually it's an exercise in sleight of hand (see Figure 4-2).

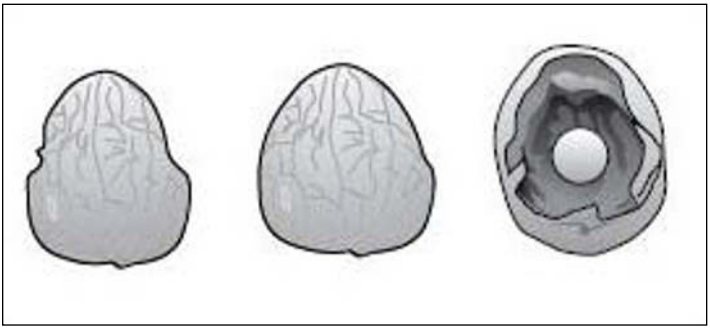


Figure 4-2: The modern threat shell game.

Similarly, the modern threat lifecycle relies on sleight of hand — how to infect, persist, and communicate without being detected.

Unfortunately, our traditional view of malware and old security habits makes us think of malware as the pea, an executable payload, perhaps attached to an e-mail. To understand, control, and successfully counter modern threats, we need to focus on not just the pea (malware) but on all the moving parts as well.

APTs and Mobile Security

With the growing use of mobile devices in the enterprise, attackers too are increasingly employing mobile devices as part of their attacks. What is interesting is that the range of services that a mobile device provides makes the device very useful for multiple parts of the attack lifecycle.

For example, an attacker could use his own mobile device for reconnaissance. With a long battery life, access to multiple networks, and the capability to reprogram its core functions, a mobile device can be a very effective tool for surveillance. Take a basic mobile phone with a prepaid plan and program it to monitor a target Wi-Fi network and send data over a

3G cellular connection back to the attacker. An attacker could simply physically mail this device to the target organization and pick up details (or even launch an attack) against the internal network from the mail room.

What about using the mobile device to gather metadata and credentials? That's easy as well, because the attacker can employ a trick to insert a man-in-the-middle to steal the employee's credentials or deliver malware, as discussed in Chapter 1.

Mobile malware can be used to infect the mobile device itself. One recent attack employed against the attendees of a political conference sent a piece of Android malware disguised as a conference proceedings tool. Once installed, the malware went after the device's contact list to gain greater knowledge of the attendee's network.

A compromised mobile device is also capable of performing further attacks once it establishes a presence within a target network. A user who brings an infected device from home may connect to the corporate network, allowing the malware to operate laterally against other machines and gather intelligence on the target network.

Not only is a mobile device the conduit for a lateral attack, but sometimes the device itself is also the target. In January 2014, Kaspersky Labs detected an extensive APT attack called Red October, where the malware incursion happened at the employees' desktop PCs but laterally moved to steal data from mobile devices in the organization, thus providing information about the users' contacts.

You can learn more about the Red October APT at `www.securelist.com/en/blog/785/The_Red_October_Campaign_An_Advanced_Cyber_Espionage_Network_Targeting_Diplomatic_and_Government_Agencies`.

This type of attack activity will become even more commonplace in the near future, as the use of mobile payment systems from mobile devices becomes more prominent, thus providing a financial incentive to use lateral movement to attack other mobile devices.

Real-world mobile malware: Dplug

In 2013, Palo Alto Networks threat researchers discovered a new kind of Android Package File (APK) mobile malware called Dplug. This malware is an example of how even basic forms of malware use communication to escalate an attack.

Dplug originated as an in-app purchase plugin for a music player app. As a result, the original host app itself doesn't do anything malicious. The plugin, however, takes advantage of the app privileges that it received from the host (including the capability to access and send SMS).

Once Dplug is installed, it communicates details about the mobile device to the attacker and intercepts all SMS messages sent to and from the mobile device. Dplug uses SMS to hijack the device's unique identifiers, subscribe to premium services, and hide this behavior from the user by blocking the premium service notifications. By subscribing the victim's device to premium SMS services, the attacker can profit from the infection.

You can learn more about the Dplug APK malware at `http://researchcenter.paloaltonetworks.com/2013/09/dplug-android-malware-discovered-by-wildfire/`.

Mobile devices have many ways to establish communication with an attacker. With ready access to cellular, Wi-Fi, and SMS, even the simplest forms of mobile malware have been able to take advantage of communicating to the attacker. When sophisticated attackers use these techniques, they are able to establish command and control and exfiltration for the data on the phone and to extract data gleaned from other parts of the network.

Chapter 5

Rethinking Mobile Security

In This Chapter

- Looking at current mobile security strategies
- Managing mobile devices for security
- Preventing mobile malware and threats
- Enforcing policies for apps and data

Before you can use mobile devices with business applications and data, you must first secure them. However, traditional mobile security approaches have limitations that you must overcome in order to safely enable the next wave of business applications. In this chapter, I survey the common approaches (and the limitations) to mobile security used by organizations today, as well as describe a comprehensive approach to mobile security that addresses three critical requirements for enterprise mobile security: managing the device, protecting the device, and controlling the data.

Recognizing Mobile Security Challenges

Mobile computing is one of the most disruptive technologies of the past decade and will likely continue to transform entire businesses and industries for the foreseeable future. Businesses are rapidly adopting mobile devices, developing mobile apps, and implementing mobile strategies to gain a competitive advantage in the global marketplace.

At its most basic level, the mobile device is a communication tool for making and receiving phone calls and accessing corporate e-mail. As enterprise mobile strategies mature, mobile capabilities become more advanced with new applications and greater access to company data, opening the door to new business opportunities and benefits — and risk (see Figure 5-1).

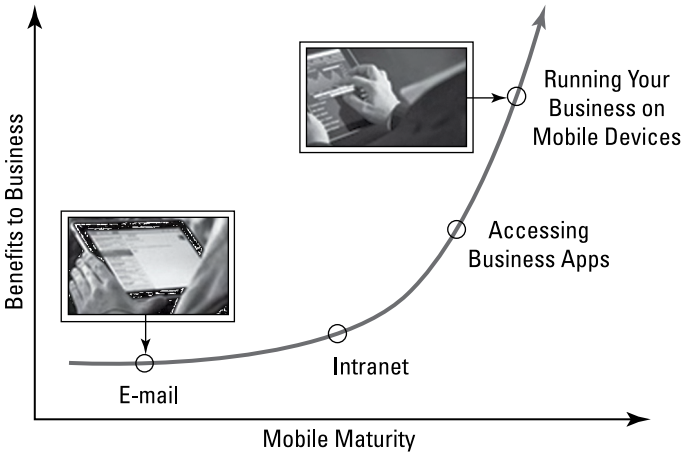


Figure 5-1: Unlocking the potential of mobile devices depends on security.

In order for organizations to adopt more sophisticated uses of mobile devices, enterprise security teams need to ensure that they address business concerns about the inherent risks to sensitive assets and confidential information that mobility brings. Unfortunately, many traditional mobile security tools tend to focus on very basic use cases and may be as limited in their security capabilities as the use cases themselves. The path to unlocking the full value of the mobile device depends on security, which provides the means to extend applications safely. Security should be seen as a way to enable mobile initiatives rather than as a limitation to mobile strategies.

By their very nature, mobile devices operate in remote locations that are not controlled by the organization. These devices, which have access to corporate applications and sensitive data, could be lost, stolen, or compromised, potentially putting sensitive business data at risk.

Security should provide the means to mitigate risk, but many common mobile strategies today are limited in scope. Today, most companies think about mobile security in terms of the following:

- **Blocking mobile devices and apps:** Some organizations attempt to insulate themselves from the effects of mobility by trying to prevent mobile devices and apps from being used at all. However, industrious employees who want to use their mobile devices at work will find creative — and risky — ways to use their mobile technology without the company's knowledge or approval.
- **Attempting to protect mobile devices and apps with existing security products:** Another common approach taken by many organizations is to simply use the same security products and technologies used to protect laptops and hope that they will secure mobile devices and apps as well. However, this approach doesn't adequately cover the threat model for mobile devices. For example:
    - Firewalls can't inspect traffic for users on a cellular network or a remote Wi-Fi network with direct connections to the Internet.
    - Anti-malware and intrusion prevention systems (IPS) may not provide any signatures to stop mobile threats.
    - Endpoint security technologies and host firewalls are not always suitable for devices with limited computing power and battery life.
- **Using basic mobile security tools:** ActiveSync is one such example of a basic mobile security tool. ActiveSync provides a system for configuring mobile devices to access corporate e-mail on Microsoft Exchange servers. It also has support for administrative functions, such as remote wipe. It is optimized to specifically address e-mail, but as mobile usage matures, more comprehensive security measures become necessary. For example, ActiveSync does not detect mobile malware. It also cannot establish policies for controlling access to applications and data other than e-mail. As a result, the need for mobile security can outgrow some of the measures that are typically used today.

These approaches will get you only so far along the mobile spectrum. At some point, you'll need to get more specific and figure out which users with mobile devices can access specific corporate applications. Namely, how do you safely enable mobile devices for business use?

To safely enable mobile devices, you need to take a simple approach and do these three important things:

- Manage the device
- Protect the device
- Control the data

Managing Devices

The first prerequisite for safely enabling mobile devices is to configure them properly for business use. Mobile devices have hundreds of settings that you can use to customize various features and functions. If set incorrectly, some of these settings could make it easier for unauthorized third parties to access apps and data on the device. Organizations should establish standard configurations that set up and enforce the usage of the correct settings in accordance with corporate standards.

Managing devices at an enterprise level typically requires mobile device management (MDM). A small client app is usually installed on an unmanaged device, and when the user logs in, the correct settings will be configured. Examples of settings that are typically configured on mobile devices include

- Requiring a PIN, passphrase, or swipe pattern to unlock the device.
- Encrypting the device's contents.
- Disabling certain device features such as the camera, GPS, or microphone.
- Accessing the corporate Wi-Fi network.
- Setting up a VPN connection.
- Creating a corporate e-mail account.
- Enabling remote support.

Managing device settings is also a prerequisite for protecting the device.

Protecting Devices

Mobile threats are everywhere, including the very environment in which they operate, so you must address the network connection first. You need to make sure that the stuff that's private stays private; otherwise, anyone who shares the same network can see what your users are doing. That's why using a VPN is important; it keeps private traffic safe, even in public places. VPNs were once used just for providing remote access to the corporate network as a way to get access to the work network when you were at home. Today, a VPN is necessary to ensure that sensitive traffic is properly encrypted to prevent unauthorized disclosure to other, less-scrupulous users on a public Wi-Fi network.

VPNs also play an important role in policy enforcement and threat prevention. A VPN helps to ensure that the same policies that are applied when employees are in the office are the same policies that are applied wherever your employees are, if appropriate. A VPN also provides comprehensive threat prevention from a number of potential attack vectors, using security features such as:

- **Vulnerability protection:** The mobile device's operating system and apps will inevitably have software bugs that can expose vulnerabilities. Exploits provide bad input to the application in order to do bad things. Vulnerability protection stops exploits from reaching applications.
- **Malware prevention:** Malware prevention blocks malware, such as viruses, worms, and Trojans, from reaching the mobile devices in order to keep them from becoming infected.
- **Malware detection:** Even with malware prevention, you need to ensure that mobile devices aren't already infected. Malware detection also analyzes normal app behavior to help detect new or unknown malware, such as advanced persistent threats (APTs).
- **Application identification, URL filtering, and DNS:** Managing applications is important to ensure that your users are adhering to policy, thus reducing the potential attack surface on mobile devices. In other words, you can reduce the amount of work required to analyze threats if you don't have to look at applications that don't belong on your network — and therefore aren't on your network — at all. In addition, network-borne attacks use network resources

> to communicate with command-and-control servers. Managing the applications you want and disrupting the network services used by malware is done with a combination of application identification, URL filtering, and DNS controls.

Threat prevention brings together multiple techniques to provide a complete system for stopping mobile threats.

Controlling Data

Your mobile security solution should enforce security policies that control access to applications and data. It's important to define policies in terms of policy criteria that map to mobile device scenarios in your actual environment. These criteria include application, user, content, device, and device state. Unfortunately, most network security products today identify port numbers rather than applications and IP addresses rather than users — and they don't understand content, devices, or device state at all.

Device-based access policies allow the organization to discern between managed and unmanaged devices. For example, an organization that is concerned about sensitive company data on personal devices may choose to establish a policy that grants access to an application on managed devices only. If a user attempts to access the same application using an unmanaged device, the policy blocks the connection.

Controlling data also means preventing access from devices that are not appropriately configured or that are infected with malware. Your mobile security solution should be able to identify the state of the device and look for dangerous conditions — such as whether malware is present — and apply policies that restrict network access until the issue has been remediated. This allows the organization to keep sensitive data away from malware-infected devices, limit the infected device's exposure to other network resources, and prevent lateral movement of any malicious activity.

The value of sensitive data on a mobile device is far greater than the value of the mobile device itself.

Chapter 6

Six Ways to Get Started with a Mobile Security Strategy

In This Chapter

- Looking at ways to safely enable mobile devices

This chapter gives you six tips to help you implement a mobile security strategy for your organization.

Define Exactly What Is Permitted in Your Mobile Environment

Not too long ago, mobile devices were pretty much all the same. Sure, they came in different shapes, sizes, and colors, and some had a few neat widgets like e-mail, calculators, and games, but they were all basically just phones. But today's smartphone is truly a smart device — with more computing power and capabilities than many desktop computers still in use today. And we can't forget about tablets. Together, the growth and ubiquity of smartphones and tablets is far outpacing desktop and laptop computers in both the consumer and business markets. As mobile devices proliferate, so too do the differences between them — and not just in terms of features and functionality, but also security.

Over time, exploits that target vulnerabilities in older mobile devices can become more serious. As discussed in Chapter 3,

many of these devices won't necessarily have patches available because the manufacturers of these devices continue to focus on delivering the "next big thing."

Organizations must define exactly what mobile devices are suitable for use with their corporate applications and what mobile devices and apps will be permitted on the corporate network.

As a security practitioner, you must help shape and inform your organization's policies by identifying clear standards for mobile security that include, for example, minimum operating system versions.

With a clearly defined policy that spells out exactly what is and isn't permitted by the organization, employees can make choices for their personal technology that are consistent with the security requirements of the organization.

Enforce Device-Based Security Policies

Defining a device-based security policy is a good start. Making sure that policy is enforced consistently is the next step. Enforcing your policy based on device criteria is critical because you may want to make certain policy decisions that are based on the type of devices being used. Is the user trying to access a critical application with an outdated version of iOS? Does the device have all of the latest critical operating system patches installed?

The operating system, version, and patch status of the device are just a few of the many things that you need to know about a mobile device that is attempting to connect to your network and applications. Has a device been jailbroken or rooted (see Chapter 3)? If so, can you control what applications and network segments these devices can access? Can you detect malware on mobile devices before permitting access to applications?

These are but a few of the things that need to be checked in order to protect your network and data center environment.

Building your internal network security policies around device-specific use cases helps you gain greater granularity and control on your network.

Implement Threat Prevention for Mobile Devices

Mobile devices and laptop/desktop computers are certainly different, but should your security measures and policies require a completely different environment for mobile devices? As explained in Chapter 5, trying to extend existing security measures to mobile use cases is largely ineffective.

Deploying an entirely new infrastructure to handle mobile devices while maintaining a separate legacy security environment for desktops and laptops presents considerable problems. Doing so increases the administrative burden for your security team and introduces additional risk because of a larger attack surface and greater opportunity for mistakes and errors. This approach also creates separate policies for what users can do on a laptop or desktop computer versus what they can do on mobile devices.

Instead, you should consider the requirements being driven by the threat model, rather than the device or platform that the threat appears on. Start by examining what's needed to deal with the sophisticated attacks that exist today. How can your organization deliver protection against both known and unknown forms of malware? What does it take to stop an advanced persistent threat (APT) that is targeting your organization?

A security environment that can deliver threat prevention (such as stop exploits on the network, block known malware, and content filtering), as well as global intelligence to identify new or previously unknown forms of malware, is most effective for protecting the entire organization regardless of platform or device.

Identify Devices Infected with Malware

Anti-virus or anti-malware software has traditionally been used to detect malware on endpoints, such as desktop and laptop computers. However, using this same approach on mobile devices is largely impractical because of factors such as computing power, battery life, cellular network bandwidth, and cellular data limits.

A better approach for identifying malware on mobile devices is to periodically check the device against a policy based on device state. This approach requires a security solution that can

- Identify malware on mobile devices and dynamically analyze apps to detect new, previously unknown malware. This identification and analysis can be accomplished with threat prevention because modern dynamic analysis of apps drives the development of better intelligence on new threats.
- Use threat information effectively, for example, by restricting the access an infected mobile device has to network resources and applications. A device-based policy (discussed earlier in this chapter) should be defined to prevent access to network resources for devices in a malware-infected state.

Segment Your Network and Begin with a Zero Trust Model

Organizations are moving away from the notion of trusting all users and devices and are instead developing their new network architecture around the concept of zero trust. Rather than provide unrestricted access to all users and devices, boundaries between various parts of the network are clearly demarcated and require specific checks before allowing the user or device to access the network segments.

These notions of zero trust are partly driven by the need to compartmentalize various functions of the network. Users

who don't have a legitimate business need to access particular applications should not have the ability to do so.

Zero trust also introduces better decision-making criteria for policy enforcement and provides ways to limit the exposure of a threat operating inside the network. A threat that establishes itself inside a target network should not have free reign of the entire network.

Enable Secure Access for Mobile and Remote Users

The modern enterprise continues to become far more distributed than in the past. Once the domain of road warriors, a significant portion of the enterprise-user population now works remotely. Users simply expect to be able to work from any location, whether at an airport, a coffee shop, a hotel room, or at home, and to connect to their applications and data via Wi-Fi, cellular network, or any means necessary.

Regardless of where users are, or even where the applications they're using might be, the same standard of control should apply. Consistent visibility and control over traffic regardless of where your users are and what devices they're using form the basis for protecting traffic and establishing a consistent set of policies that apply to both local and remote users. This is not to say that organizations will have the very same policy for users inside the office and on the road. Some organizations might want employees to use Skype when on the road, but not inside the office, whereas others might have a policy that says if outside the office, users can't download Salesforce.com attachments unless they have hard disk encryption turned on. This should be enforceable without introducing significant latency for the end-user, undue operational burden for the security team, or significant cost for the organization.

Glossary

advanced persistent threat (APT): A sophisticated cyberattack against a large enterprise or government agency that is carried out over an extended period of time by an organization with substantial financial, computing, and human resources, such as a criminal organization or nation-state, for the purpose of data theft, reconnaissance, or sabotage.

Android: A mobile operating system developed by Google.

app: An abbreviation commonly used to refer to mobile applications.

application programming interface (API): A set of routines, protocols, and tools that specify how certain software components interact with each other.

application package file (APK): The package file format used to distribute and install Android apps and middleware.

bot: A computer that has been infected with malware and is under the control of an attacker.

botnet: A large number of bots, perhaps tens of thousands, under the control of one or several attackers.

Bring Your Own Device (BYOD): A prevalent trend in the modern workplace in which employees are permitted to use their own mobile devices, such as smartphones and tablets, to perform work-related functions.

brute-force attack: A type of attack in which the attacker attempts every possible combination of letters, numbers, and symbols to crack a password, passphrase, or PIN.

Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA): A challenge-response test commonly used on websites to determine whether a user is human or an automated program.

ciphertext: A plaintext message that has been transformed (encrypted) into a scrambled message that is unintelligible. See ***plaintext***

Corporate Owned, Personally Enabled (COPE): An alternative to BYOD, the organization provides its employees with a choice of approved mobile devices and permits the devices to be used for certain personal uses. See ***Bring Your Own Device (BYOD)***

denial-of-service (DoS): The purpose of a DoS attack is to make a computer, application, or network unavailable to its intended users.

distributed denial-of-service (DDoS): DDoS is a large-scale attack that typically uses bots in a botnet to overwhelm a targeted network or server to interfere with the capability to serve legitimate users. See ***bot*** and ***botnet***

Extensible Authentication Protocol (EAP): An authentication framework commonly used in wireless networks and point-to-point connections.

Institute of Electrical and Electronics Engineers (IEEE): A professional organization that develops global standards for various industries.

IEEE 802.1x: A standard that defines encapsulation of EAP to provide authentication for devices connecting to a wired or wireless network. See ***Extensible Authentication Protocol (EAP)***

IEEE 802.11: The wireless networking standard that defines frequency, bandwidth, data rate, and range, among other things.

initialization vector (IV): A random (or pseudorandom), fixed-size input used in cryptography to ensure that different segments of an encrypted message cannot be inferred by an attacker using the same encryption key.

instant messaging (IM): A program that provides real-time text transmissions across a network, such as the Internet.

jailbreaking: The term for hacking an Apple device so that users can bypass security features for the purpose of downloading applications from alternative app stores.

malware: Malicious software that typically damages, takes control of, or collects information from a computer or device. This classification of software broadly includes viruses, worms, Trojan horses, logic bombs, spyware, and (to a lesser extent) adware. See ***Trojan horse***

National Institute of Standards and Technology (NIST): A federal agency within the U.S. Department of Commerce that is responsible for promoting innovation and competitiveness through standards, measurement science, and technology.

one-way hash function: An algorithm that can be easily computed in one direction, but not in the reverse direction.

plaintext: A message in its original readable format or a ciphertext message that's been properly decrypted (unscrambled) to produce the original readable message format. See ***ciphertext***

pre-shared key (PSK): A shared secret between two parties used in cryptography.

promiscuous mode: An operating mode for a network interface that processes or captures all traffic instead of only the traffic that is addressed to that node.

Remote Authentication Dial-In User Service (RADIUS): A network security protocol that provides centralized authentication, authorization, and accounting (AAA) for users connecting to a network.

Rivest Cipher 4 (RC4): A stream cipher commonly used in TLS and WEP. See ***Transport Layer Security (TLS)***, ***stream cipher***, and ***Wired Equivalent Privacy (WEP)***

rooting: The term used for hacking a Google Android device to obtain elevated privileges, allowing the user to replace protected functions or even the entire operating system.

salt: Randomly generated data that is used as an additional input for a one-way function. See ***one-way hash function***

Secure Sockets Layer (SSL): The predecessor to TLS, SSL is a cryptographic protocol that provides communication security over the Internet. See ***Transport Layer Security (TLS)***

Service Set Identifier (SSID): The name of a wireless network.

stream cipher: An encryption algorithm that operates on a continuous stream of data, typically bit-by-bit.

Transport Layer Security (TLS): The successor to SSL, TLS is a cryptographic protocol that provides communication security over the Internet. See ***Secure Sockets Layer (SSL)***

Trojan horse: A malware program that purports to perform a given function but that actually performs some other (usually malicious) function. See ***malware***

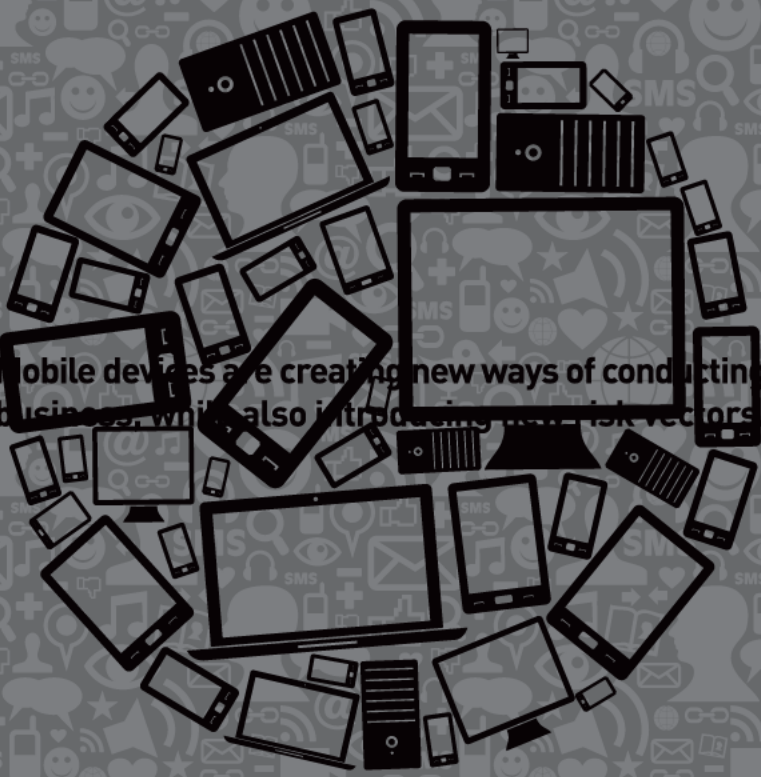
virtual private network (VPN): Securely extends a private network across a public network, such as the Internet.

Wi-Fi Protected Access (WPA/WPA2): Two wireless networking security protocols developed to address inherent weaknesses in WEP. See ***Wired Equivalent Privacy (WEP)***

Wired Equivalent Privacy (WEP): A wireless networking security protocol.

Notes

Notes



Safely Enable Mobile Devices with GlobalProtect™

GlobalProtect from Palo Alto Networks® provides a comprehensive security solution for mobile devices built upon the technologies of the Palo Alto Networks enterprise security platform and tailored to address mobile requirements. It delivers a unique combination of technology and global intelligence to secure mobile apps and to stop mobile threats. These principles enable organizations to address the three core requirements for mobile security: manage the device, protect the device, and control the data.

To learn more about how to safely enable mobile devices, visit: www.paloaltonetworks.com/globalprotect

paloalto networks®

Safely enable your mobile strategy!

Mobile technology creates new opportunities for businesses — and new risks. Traditional security products and basic mobile tools are inadequate and ineffective against mobile threats. In this book, you find out how to safely enable your mobile devices.

- ***Recognize threats to mobile devices* — learn about the risks that mobile devices face**
- ***Control mobile apps* — know the different security models and how apps are exploited**
- ***Recognize mobile threats* — understand modern malware, botnets, advanced persistent threats, and new mobile attack vectors**
- ***Rethink mobile security* — safely enable mobile devices by controlling threats**

Open the book and find:

- **Ways that attackers exploit devices on Wi-Fi networks**
- **Why protecting mobile apps is as important as protecting mobile devices**
- **Why malware and advanced persistent threats have gone mobile**
- **How security can drive mobile innovation and strategy, rather than inhibit it**

Go to Dummies.com®
for videos, step-by-step examples, how-to articles, or to shop!

Lawrence Miller has worked in information security for more than 20 years. He coauthored *CISSP For Dummies* and has written more than 50 other titles. He is also a Palo Alto Networks customer and liked it so much he bought the company — well, he's not that rich (yet) — but he did write this book!

FOR DUMMIES®
A Wiley Brand

ISBN 978-1-118-91426-7
Book not for resale

WILEY END USER LICENSE AGREEMENT

Go to www.wiley.com/go/eula to access Wiley’s ebook EULA.