PERIMETER E-SECURITY

AUTHOR PROFILE:
ANDREW JAQUITH,
CHIEF TECHNOLOGY OFFICER

Andrew Jaquith brings 20 years of IT and information security experience to Perimeter, most recently as a senior analyst with Forrester Research. At Forrester, Andrew led team coverage for data, endpoint and mobile security topics. In his time at Forrester, he wrote 20 popular reports on data leak prevention, encryption, endpoint security, mobile security and vendor M&A. Notable recent reports include "Security in the Post-PC Era," "Apple's iPhone and iPad: Secure Enough for Business?" and "The Forrester Wave: Data Leak Prevention Suites." Andrew consulted with and assisted 300 enterprise and vendor customers annually with vendor selection, compliance, strategy and effective practices.

Prior to joining Forrester, he was program manager in Yankee Group's enabling technologies enterprise group, with coverage of client security, digital identity, and web application security. Before joining Yankee Group, he co-founded @stake, a security consulting pioneer, which Symantec acquired in 2004. Before @stake, he held project manager and business analyst positions at Cambridge Technology Partners and FedEx.

Andrew's security research has been featured in publications such as CIO, CSO, and the IEEE Journal of Security & Privacy. In addition, he is the co-developer of the Apache JSPWiki open source wiki software package, and the author of the 2007 Addison-Wesley Professional book "Security Metrics: Replacing Fear, Uncertainty and Doubt." The book has sold more than 10,000 copies and has been praised by reviewers as "one of the best written security books ever."

Andrew holds a B.A. in Economics and Political Science from Yale University.

PICKING A SENSIBLE MOBILE PASSWORD POLICY

Defining an enterprise mobile device passcode policy can be surprisingly difficult. Security managers must attempt to reconcile two opposing goals. They must:

- Create a passcode policy that is strong enough to protect the device if it is lost or stolen, while:
- Not annoying users with needless length or complexity

These goals are hard to reconcile because mobile devices like smartphones and tablets are personal, portable and convenient. Employees use their devices in places they wouldn't use a PC: in the car, during their kids' football game, and during (shall we say) otherwise unproductive periods of the day. It's tempting to simply duplicate existing network security policies. The rationale goes something like this: smartphones and tables are nothing more than small PCs with antennas, so the password policies should be the same as for PCs. It's easy to think that, but it's the wrong attitude.

I'm going to describe the passcode policy I recommend for mobile devices that comply with NIST's e-authentication Level 1 guidelines as described in Special Publication 800-63, "Electronic Authentication Guidelines"[1]. My policy is reasonable, employee-friendly and highly usable, but strong enough to protect your company's data. Here are the elements:

- 8-digit numeric PIN
- Simple PINs disallowed
- Automatic lock after 15 minutes
- Grace period of 2 minutes
- Automatic wipe/permanent lock after eight wrong tries
- No expiration

For details, read on. Warning: a tiny bit of binary math lies ahead.

THE RIGHT PASSCODE LENGTH: 8 DIGITS OR 6 CHARACTERS WITH AN AUTOMATIC WIPE POLICY

Length and composition are the most important parts of any mobile device passcode policy. The longer the passcode, the better. The "best practice" that many security admins follow in the PC world is to require a "strong" password of at least eight characters, plus at least one special character. The goal of this policy is to make the password strong enough so that an attacker wouldn't be able to guess it within an allotted time period. But what is "strong enough?" As it happens, our friends at NIST have defined this fairly precisely: for an 800-63 "Level 1" password, "strong enough" means 10 bits of guessing entropy. "Guessing entropy" comes from Claude Shannon's work on information theory. It is a probabilistic measure that an attacker will successfully guess a password over its lifetime, expressed as the number of chances the attacker

Corporate Headquarters | Perimeter E-Security | 440 Wheelers Farms Rd. | Suite 202 | Milford, CT 06461 | 800-234-2175 | 203-878-1284 Fax

PERIMETER E-SECURITY

PICKING A SENSIBLE MOBILE PASSWORD POLICY

would need. This number is measured in bits (aka powers of two). For example, two bits of guessing entropy means that an attacker would need four tries to guess the password (2^2).

For a Level 1-compliant password, NIST defines the required strength as 10 bits of guessing entropy. In other words, an attacker who knew nothing more than the employee's username would have at most 1024 (2^{10}) tries to guess the password for the entire time the password is active. For a NIST 800-63 "Level 2" password, an attacker would need an estimated 65,536 (2^{16}) guesses to break the password. (Level 3, in case you were wondering, is a Level 2 strong password plus a soft cryptographic token or certificate; Level 4 is the same but requires a hard token).

If all we need to do is pick a strong password, how do we do that? It turns out the answer to this question is, "it depends" based on complexity rules and length, plus whether the employee chooses the password or the system generates it for them. In NIST SP 800-63 Appendix A, Table A.1[2], NIST estimates the guessing entropy of various combinations. For example, in order to achieve 10 bits of guessing entropy (a Level 1 passcode), assuming the attacker had just one chance to guess it, the following types of passwords would qualify:

- a 3-digit numeric PIN that the system generated randomly. *Entropy: 10 bits of guessing entropy*
- a 5-digit numeric PIN that the employee picked themselves, disallowing simple PINs that repeat the same digit or use a sequence (e.g., 12345). *Entropy: 10 bits*
- a 2-character random password that the system generated randomly, and that used the entire 94-character keyboard (A-Z, a-z, 0-9 and the characters Ð!@#$%^&*()_-+={}[]|¥:;'<,>.?/"“). *Entropy: 13.2 bits*
- a 4-character passcode that the employee picked themselves, using the entire 94-character keyboard. *Entropy: 10 bits*

Two things jump out from these examples. First, note how much stronger the randomly-generated passcodes are than user-chosen ones. The 3-digit random PIN, for example, is as strong as a 5-digit user-generated one (both have 10 bits of guessing entropy). This is because humans aren't very good at picking random numbers. The second thing that jumps out is that these passcode lengths probably seem unnaturally short to you! Why? That's because, as I described, we assume the attacker has just one chance to guess the passcode.

But of course, the attacker never has just one chance to guess a password; they usually get *many* chances. Thus, most password strength policies contain the buried assumption that the attacker has thousands or millions of chances to guess the password over its lifetime. That's why the typical password policy calls for eight characters, with (for example) at least one upper case letter and a number, plus one special character. Per NIST, a password policy like that boosts entropy to 24 bits, which is 2^{14} times more than the 2^{10} single guess entropy estimate that Level 1 actually requires. In other words, your typical desktop password's policy essentially assumes that an attacker gets 2^{14} or 16,384 chances to guess the password. If anything, this is probably far too small a margin of safety: an attacker who has gotten access to your Windows domain controller's SAM file, for example, can execute millions of guesses in just a few seconds. Regardless, you get

Corporate Headquarters | Perimeter E-Security | 440 Wheelers Farms Rd. | Suite 202 | Milford, CT 06461 | 800-234-2175 | 203-878-1284 Fax

PICKING A SENSIBLE MOBILE PASSWORD POLICY

the idea: that long password your IT department wants you to put on your PC assumes an attacker will have many, many opportunities to break in.

In the case of smartphones and tablets, though, the operating systems typically have a feature that allows administrators to control the number of guesses an attacker gets: the automatic wipe/permanent lock setting. Put simply, modern smartphone operating systems from RIM, Apple, Microsoft and Google can require devices to turn themselves into bricks if an attacker guesses too many wrong passwords. By implementing such a policy, we can effectively shorten the number of entropy bits we would need for the smartphone compared to, say, a desktop PC that we can't turn into a brick. What this means: smartphones and tablets with an automatic wipe policy do not need passwords as long or complex as those for desktops, because the number of guesses an attacker gets is so much smaller. *They can be much shorter and simpler and still provide the same level of protection. That's what the math tells us.*

For example, if we impose a policy of eight wrong guesses before a mobile device automatically wipes or permanently locks itself (a fairly reasonable restriction), we need a guessing entropy that is only 2^{13} bits – that is, just three bits (8 guesses) higher than the single-guess entropy of 10 bits. To do that, the following types of user-chosen passcodes would qualify:

- an 8-digit numeric PIN, disallowing simple PINs (13 bits), or:
- a 6-character passcode, using the entire 94-character keyboard (14 bits)

Either one of these policies will serve our purposes nicely. Personally, I prefer the 8-digit PIN policy because it's easier to key in on some smartphone operating systems. For example, Apple's iOS (the operating system that the iPhone and iPad use) will automatically pop up a numeric keypad, instead of the full alphanumeric keyboard, if the owner initially specified a passcode that contained only numbers. It's a nice usability touch that employees like because they don't have to worry as much about fat-fingering the passcode. Even better, eight digits is still short enough to be easily remembered, and they can be tapped in quickly.

AUTOMATICALLY LOCK MOBILE DEVICES AFTER 15 MINUTES

After password length and composition, deciding how long the device can be inactive before it locks itself is the second key policy decision most firms wrestle with. Employees use their devices a lot throughout the day, but on an intermittent basis. NIST has very little to say about mobile locking policies, so use your common sense. Unless your employees carry the secret formula for Coke around on their mobile devices, I generally recommend that companies choose an inactivity timeout period that accommodates employee working styles as much as possible without opening a significant window of attack. Remember, there are less to protect on these devices than on a normal PC.

By "significant window of attack," my rule of thumb is longer than a quick trip to the break room to get a coffee (5 minutes) but shorter than a lunch break (30 minutes). A sensible inactivity timeout period is probably about 15 minutes. You can go shorter than this, of course, although I personally feel 5 minutes is a fairly employee-hostile policy that will cause you to get a lot of e-mail complaints.

Corporate Headquarters | Perimeter E-Security | 440 Wheelers Farms Rd. | Suite 202 | Milford, CT 06461 | 800-234-2175 | 203-878-1284 Fax

PICKING A SENSIBLE MOBILE PASSWORD POLICY

Many mobile devices also allow employers to implement a "grace policy" setting that will delay password-locking for a few minutes, even if the device has been put to sleep (normally, the passcode lock is switched on right away). For example, if you just checked the calendar on your mobile phone, then hit the sleep switch and put it in your pocket, you would still have a minute or two to check that other thing you just remembered without being hassled with the passcode. Again, as with the automatic lock policy, common sense should be your guide. A grace period of, say, two minutes gives your employees a little extra usability without detracting from security.

DON'T REQUIRE EMPLOYEES TO ROTATE THEIR PASSCODES

In the PC world, most security administrators, and indeed most "best practices" as enshrined in NIST, ISO 27000 advocate using password "aging" policies that cause employees to regularly rotate their passwords. The goal of this practice is to reduce the likelihood that an attacker can compromise an account over its lifetime. Despite the nearly universal acceptance of password aging practices, however, there has been surprisingly few empirical analyses showing that they actually increase security. Researchers from Microsoft, for example, suggest that, if anything, password aging policies actually detract from security. This is because employees usually resort to a variety of coping mechanisms to deal with being forced to change passwords so often. They write their passwords down on sticky notes, create easy-to-remember passwords that vary only by one digit between instances, and re-use passwords between services. Microsoft concludes that the typical password rules produce a "minor reduction of risk for a 3.9x magnification of password management effort."

I am firmly opposed to password expiration policies for most employees in most contexts, although they do make sense in certain cases: for example, for highly privileged service, admin and server accounts, or in cases where you suspect a compromise. But on the whole, I'd rather encourage employees to create *harder* passwords that don't expire rather than easier ones that do. This is nowhere more true than with mobile devices. Here are three reasons why you should never implement a password aging policy on mobile devices:

- **Guessing the device password doesn't buy the attacker anything extra.** The passcode protects the integrity of the device, not the data on your network. It's not a Windows domain password. Guessing the passcode doesn't get an attacker access to any new resources other than the ones already provisioned on the device (for example, the e-mail account). A lucky guess of a device password — which means they beat 1:1024 odds, very impressive — might mean that an attacker can now send prank e-mails on the victim's behalf. But they won't be able to mount up a new SMB share that contains your secrets, for example, or loot your payroll system.

- **Password aging policies are redundant.** Recall that a key goal of expiring passwords is to shorten the lifetime over which an attacker can compromise an account. It's a fine goal, but it is already taken care of by a sensible automatic wipe policy. Eight times to guess a password today is still just eight times, regardless of whether we're talking about a passcode the employee is using this week, last quarter or next year.

- **Forced password changes are hostile to your employees.** Trust me, your employees already hate your enterprise password aging policy, and they have the Post-Its to prove it. By plopping yet another unwanted usability obstacle onto the devices they take to birthday parties, use on the subway or show business partners in restaurants, you've just given them another thing to dislike, and another incentive to evade your well-intentioned controls.

Corporate Headquarters | Perimeter E-Security | 440 Wheelers Farms Rd. | Suite 202 | Milford, CT 06461 | 800-234-2175 | 203-878-1284 Fax

PICKING A SENSIBLE MOBILE PASSWORD POLICY

So in closing: remember that these devices are often personally owned, and they contain much less sensitive data on them than a typical PC. Instead, win friends and influence people, and don't put passcode expiration policies on mobile devices.

IMPLEMENTING MOBILE SECURITY POLICIES

The policies I've described in this article can be implemented in all iPhones and iPads running version 3 or later of the operating system, and on all BlackBerry devices. Any Windows Mobile or Windows Phone 7 device can support these policies too. Finally, Android devices running 2.2 and higher support most of these settings. In a future article, I'll describe the exact settings you should use for ActiveSync-compliant devices and for Apple's mobileconfig security policy files.

ABOUT PERIMETER E-SECURITY

Perimeter E-Security is the trusted market leader of information security services that delivers enterprise-class protection and compliance. Through its cost-effective and scalable SaaS platform, Perimeter offers the most comprehensive compliance and security solutions that include: end-to-end secure messaging solutions, managed security services, vulnerability management and security consulting services.

1. http://csrc.nist.gov/publications/nistpubs/800-63/SP800-63V1_0_2.pdf
2. NIST SP 800-63 Appendix A, Table A.1

Corporate Headquarters | Perimeter E-Security | 440 Wheelers Farms Rd. | Suite 202 | Milford, CT 06461 | 800-234-2175 | 203-878-1284 Fax