10 common security mistakes that should never be made

Version 1.0
August 20, 2008

By Chad Perrin

The following is a list of security mistakes I see all the time. They're not just common, though—they're also extremely basic, elementary mistakes, and anyone with a modicum of security knowledge should know better than to make them.

1 Sending sensitive data in unencrypted e-mail

Stop sending me passwords, PINs, and account data via unencrypted e-mail. Please. I understand that a lot of customers are too stupid or lazy to use encryption, but I'm not. Even if you're going to give *them* what *they* want, in the form of unencrypted sensitive data sent via e-mail, that doesn't mean you can't give *me* what *I* want—secure communications when sending sensitive data.

2 Using "security" questions whose answers are easily discovered

Social security numbers, mothers' maiden names, first pets, and birthdays do not constitute a secure means of verifying identity. Requiring an end user to compromise his or her password by specifying a question like that as a means of resetting the password basically ensures that the password itself is useless in preventing anyone who is willing to do a little homework from gaining unauthorized access.

3 Imposing password restrictions that are too strict

I've seen an unacceptable number of cases where some online interface to a system that lets you manage your finances—such as banking Web sites—impose password restrictions that actually make the interface less secure. Six-character numeric passwords are dismayingly common, and the examples only go downhill from there. See "How does bad password policy like this even happen?" for another example in more detail.

4 Letting vendors define "good security"

I've said before that there's no such thing as a vendor you can trust. Hopefully, you were listening. Ultimately, the only security a corporate vendor really cares about protecting is the security of its own profits and market share. While this may prompt a vendor to improve the security of its products and services, it sometimes prompts exactly the opposite. You must question a vendor's definition of "good security," and you must not let vendors tell you what's important to you.

5 Underestimating required security expertise

People in positions of authority in corporations often fail to understand the necessity for specific security expertise. This applies not only to nontechnical managers, but to technical IT managers as well. In fact, standards working groups such as the one that produced the WEP standard often include a lot of very smart technologists, but not a single cryptographer, despite the fact they intend to develop security standards that rely explicitly on cryptographic algorithms.

6 Underestimating the importance of review

Even those with security expertise specific to what they're trying to accomplish should have their work checked by others with that expertise as well. Peer review is regarded in the security community as something akin to a holy grail of security assurance, and nothing can really be considered secure without being subjected to significant, punishing levels of testing by security experts from outside the original development project.

7 Overestimating the importance of secrecy

Many security software developers who make the mistake of underestimating the importance of review couple that with overestimation of the importance of secrecy. They justify a lack of peer review with hand-waving about how important it is to keep security policies secret. As Kerckoffs' Principle—one of the most fundamental in security research—points out, however, any system whose security relies on the design of the system itself being kept secret is not a system with strong security.

8 Requiring easily forged identification

Anything that involves faxing signatures or sending photocopies or scans of ID cards is basically just a case of security theater—putting on a great show without actually providing the genuine article (security, in this case) at all. It is far too easy to forge such second-generation (or worse) low quality copies. In fact, for things like signatures and ID cards, the only way for a copy to serve as useful verification is for it to be a good enough copy that it is not recognized as a copy. Put another way, only a successful forgery of the original is a good enough copy to avoid easy forgery.

9 Unnecessarily reinventing the wheel

Often, developers of new security software are re-creating something that already exists without any good reason for doing so. Many software vendors suffer from Not Invented Here disease and end up creating new software that doesn't really do anything new or needed. That might not be a big deal, except that new software is often not peer reviewed, it makes security mistakes that have already been ironed out of the previous implementation of the idea, and it generally just screws things up pretty badly.

Whenever creating a new piece of software, consider whether you're replacing something else that already does that job and whether your replacement actually does anything different that is important. Then, if it is doing something important and different, think about whether you might be able to just add that to the already existing software so you will not create a whole new bundle of problems by trying to replace it.

10 Giving up the means of your security in exchange for a feeling of security

This is a mistake so absurd to make that I have difficulty formulating an explanation. It is also so common that there's no way I can leave it out of the list. People give up the keys to their private security kingdoms to anyone who comes along and tells them, "Trust me, I'm an expert," and they do it willingly, eagerly, and often without thought. "Certificate Authorities" tell you who to trust, thus stripping you of your ability to make your own decisions about trust; Webmail service providers offer on-server encryption and decryption, thus stripping you of end-to-end encryption and control over your own encryption keys; operating systems decide what to execute without your consent, thus stripping you of your ability to protect yourself from mobile malicious code.

Don't give up control of your security to some third party. Sure, you may not be able to develop a good security program or policy yourself, but that doesn't mean the program or policy shouldn't give you control over its operation on your behalf.

Additional resources

- TechRepublic's Downloads RSS Feed XML
- Sign up for the Downloads at TechRepublic newsletter
- Sign up for our IT Leadership Newsletter
- Check out all of TechRepublic's free newsletters
- 10 things you should do to ensure basic Web site security
- 10 ways to reduce insider security risks
- Avoid falling prey to these five security oversights

Version history

Version: 1.0
Published: August 20, 2008

Tell us what you think

TechRepublic downloads are designed to help you get your job done as painlessly and effectively as possible. Because we're continually looking for ways to improve the usefulness of these tools, we need your feedback. Please take a minute to drop us a line and tell us how well this download worked for you and offer your suggestions for improvement.

Thanks!

—The TechRepublic Content Team