Five Mistakes of Data Encryption

Dr Anton Chuvakin @ LogLogic

If you follow the media today, you might get to a conclusion that data encryption is everywhere. However, is this “good” encryption? A classic saying “Encryption is easy; key management is hard” illustrates one of the pitfalls that await those implementing encryption enterprise-wide or even SMB-wide. This paper covers some of the other mistakes that often occur when organizations try to use encryption to protect data-at-rest and data-in-transit and thus improve their security posture.

The first mistake is not using encryption when it is easy and accepted. Yes, I am talking about those pesky plain text protocols such as telnet and FTP. One can argue that people should have abandoned the above protocols for a host of other reasons, but, as the latest Solaris telnet 0day (here) fiasco indicates, enough people are stuck using them.

Similarly, while using HTTP for sensitive online transaction is not common, one still sees such instances on lesser known e-commerce websites. Exposing sensitive information to known, actively used attacks, such as sniffing, is inexcusable. While risk of sniffing is typically overshadowed by the risks to stored data, there is indeed no excuse to not encrypting the data in transit when it is easy and does not cost any extra.

The **second mistake** has been mentioned by most cryptographers out there: inventing your own cryptographic algorithm. Sorry, but most of us are not that smart! Cryptography is a science, just like physics and mathematics (in fact, we all know that it is based on the latter) and, amateurs have no place in it. On the other hand, if you are looking to secure a highly-sought spot in Bruce Schneier’s doghouse, go help yourself to a copy of crypto for dummies and start coding right away …

An interesting extension of this mistake has to do with failing to correctly implement a well-designed cryptographic algorithm. Indeed, algorithm design is hard, but quality implementation is not an easy job as well. As a result, people who chose to re-implement a “known good” crypto algorithm might be doing themselves a disservice, if a proven implementation exists (as a cryptographic library, for example)

Every security pro worth his salt will recognize **the third mistake:** “hard-coding” secrets. As we know, security of a quality cryptographic algorithm does not depend on its secrecy, but on its key or password. If you inadvertently make such password available for attackers, the game is over. Embedding passwords in code (binaries), configuration files or other “hidden” files is just that: providing your secret to attackers. And, no, your XOR’ing the password with a string of characters does not count: it just replaces your credible “secured by AES” label by a purely humorous “protected by the power of XOR.” You can only secure a password by encrypting it and then your problem does not go away since you have to deal with a new password.

Hard-coded secrets let to many a disaster in recent history of information security. DVD and now, it seems, HD-DVD decryption is just the most famous of them.

The extension of such error, where passwords are “hidden” in files in order to enable scripting or automation, have helped attackers to extend their control over the compromised networks. One cannot argue that system administrators need to automate routine tasks, but “root” passwords in scripts and configuration files should be as archaic as double digit years. Finally, the question of whether having passwords in Javascript code that is downloaded by the web browser is a good idea is left as an exercise to the reader …

A database data encryption is all the rage now: just protect your database by encrypting it. Great! The task is done and the data is secured by a well-implemented encryption algorithm. Now, where do we put the keys? Ah, why not in the same database? Thus, **the fourth mistake** is manifested in the form of storing keys with data. The author is aware of a few organizations who sought to protect their credit card databases by encrypting the tables with sensitive data and then storing the key in another table on the same database server.

Sometimes, one hears a claim that such “protection” works against fools and low-skill attackers: they would see a string of random binary data where a credit card number should have been and then go away. However, a more correct way of putting it is that it works as a “checkbox encryption” “against” your own management – they can now claim that cryptography protects their organization’s crown jewels, while in reality it protects nothing.

While we are at it, one needs to think about the following as well: while you might not be leaving the keys in an obvious place such as a database table, do you prevent key leakage into swap files, crash dumps, logs and other areas that might be seen by attackers? This is much more insidious and a detailed discussion goes beyond the scope of this paper.

Finally, the **fifth mistake** turns encryption again the very entity that is supposed to benefit from it - your organization: not handling data recovery. So, does your crypto implementation passes “a bus test.” If whoever knows the keys to data is hit by the bus, will you be able to get your data back?

If you implemented cryptography correctly and thus there is no way to bypass the security it provides and, at the same time, you didn’t think about data recovery, your implementation will likely not pass the bus test. As a result, the data would be as good as gone in case of a key loss. Did we mention that cryptography is a science? Handling key revocation and data recovery is a critical piece of the security puzzle. For example, Windows EFS has support for such features, described in details here. Thus, protecting data from theft is only half of the challenge – you need to protect the data from loss due to “good” crypto.

To conclude, data protection is a critical piece of information security and encryption plays a major role in it. However, one should try to avoid the mentioned pitfalls and should consider data encryption to be that long-sought security “silver bullet”…

Dr Anton Chuvakin, GCIA, GCIH, GCFA (http://www.chuvakin.org) is a recognized security expert and book author. In his current role as a Director of Product Management with LogLogic, a log management and intelligence company, he is involved with defining and executing on a product vision and strategy, driving the product roadmap, conducting research as well as assisting key customers with their LogLogic implementations. He was previously a Chief Security Strategist with a security information management company.

A frequent conference speaker, he also represents the company at various security meetings and standards organizations. He is an author of a book "Security Warrior" and a contributor to "Know Your Enemy II", "Information Security Management Handbook", "Hacker's Challenge 3" and the upcoming book on PCI. Anton also published numerous papers on a broad range of security subjects. In his spare time he maintains his security portal http://www.info-secure.org and several blogs, including http://chuvakin.blogspot.com