Portable Storage and Data Loss

Stephen Farrell • *Trinity College Dublin*

Data loss or leakage occurs in many organizations, frequently with significant impacts, both in terms of incident-handling costs and of damage to the organization's reputation. In this installment of Practical Security, I consider information leakage related to portable storage — for example, your laptop hard-disk — and what might best mitigate that. I briefly consider some recent incidents, describe practical mitigation steps, and look at how we might plan, in advance, for handling such events.

Recent Incidents

Information leakage events have been widely reported in recent years. In both May and August 2006, for example, the US Department of Veterans' Affairs (VA) suffered significant data losses (www.usa.gov/veteransinfo.shtml). In the first case, an intruder burgled a VA employee's home and stole, among other items, a laptop that (in breach of VA policy) contained identifying information about millions of veterans. Subsequently, some of those affected launched lawsuits against the organization.

In January 2007, the VA suffered another data loss when one of its IT specialists couldn't locate an external hard disk, which the VA ultimately presumed was stolen. This case is informative because the VA Office of the Inspector General thoroughly documented it.[1] From this description, we can see that a sequence of policy breaches, each arguably relatively minor, resulted in the IT specialist eventually accumulating significant amounts (more than 1.3 million healthcare provider records) of personally identifying information (PII), without strong protection and, in this case, without being fully authorized to possess that data. Even after this sequence of data loss events, the VA, and several other organizations, continue to have difficulty preventing further incidents.

However, organizations can also become sensitized by non-data-loss events. The Irish Blood Transfusion Service (IBTS) has had to deal with various infection scandals, so when they recently suffered an information leakage event, they felt they had to contact all 170,000 people potentially affected. The laptop stolen in this incident contained data used for testing. Because the IBTS was using real records rather than fake data, it sent out notification letters (www.autoschism.com/images/out0014.jpeg), even though, according to news reports, the data were strongly encrypted (see www.independent.ie/opinion/analysis/ibts-faces-massive-costs-if-personal-data-is-leaked-1293907.html). In an echo of the VA case, the stolen laptop belonged to a contractor who was somewhat detached from normal day-to-day IBTS operations — in fact the data had been exported to another country for interoperability testing when the loss occurred, bringing up potential cross-border issues related to data export.

As a follow-up to another recent incident — in which millions of UK taxpayer details on a CD-ROM went missing (www.publications.parliament.uk/pa/cm200708/cmhansrd/cm071217/debtext/71217-0006.htm) — spammers targeted those affected, offering them a fake tax rebate in an effort to get users to enter further personal details (www.theregister.co.uk/2008/02/22/hmrc_phishing_attack/). Such so-called spear-phishing has been highly effective in other circumstances.[2]

In all these cases, the lost data was kept on mobile storage (laptops, removable hard-disks, or CDs). Today's smaller mobile devices (phones or PDAs) can store gigabytes of data and might

1089-7801/08/$25.00 © 2008 IEEE

easily become vectors for data loss. However, few publicized cases of massive data loss via such devices have occurred to date. Whether that will change probably depends on the types of applications that become common on such devices.

Prevention and Mitigation

The first way to prevent such incidents is to leave the data at "home," which in the case of most data, means the office network, not the employee's house. This might seem obvious, but it can be challenging because it could require significant changes to an organization's enterprise IT infrastructure. Generally, mobile workers have become used to being able to access data from their office desktop, from home, while traveling, and, crucially, while offline (such as on a flight). The enterprise IT infrastructure has now caught up with these trends and can offer this kind of access, but it frequently requires storing copies of documents, databases, and the like on portable devices.

If an organization's main defense against data loss is centralized storage, then that organization clearly must properly control access to the data centrally stored. For example, if deleting a Web server access-control file (such as the ".htaccess" file for Apache) would expose sensitive data, then an organization must also control OS-level access to the server in question, and should also look for any cases in which administrators mistakenly grant access (which can be hard to notice given that authorized users might see no change to their access to data). Essentially this calls for the use of enterprise access control and audit trails.

Anonymization

Another useful step is to anonymize sensitive data (in one VA case, the organization was storing some US social security numbers in an obfuscated form), but you must do this with care, particularly if another public data set could be correlated with the sensitive information. Take, for example, the anonymized data set published by Netflix; this was correlated with a public movie-review database so that time-based correlation exposed the identities of some anonymized records.[3] Organizations might also be able to mask certain fields, such as credit-card numbers, by storing only their last four digits in the data set. The Payment Card Industry (PCI) has produced a Data Security Standard (PCI-DSS)[4] for handling account information that requires, for example, that merchants never store card verification values (CVVs). For merchants, breaching this standard might be extremely expensive because credit-card associations impose penalties for noncompliance.

Data Encryption

One very effective mitigation against data loss is to encrypt the data store, and many free and commercial tools can do this. Of course, you should use such tools only when they properly utilize well-known encryption algorithms, such as the Advanced Encryption Standard (AES).[5] However, caveat emptor, as usual — some products have been found to use both strong and weak encryption algorithms, with the result that an "encrypted" hard-disk can be trivially decrypted (see www.heise-online.co.uk/security/Enclosed-but-not-encrypted--/features/110136). In addition, if decryption depends on a user-memorized key (that is, a password), we must assume in the event of a loss that the adversary can brute-force attack the encryption. Thus, from an incident-handling viewpoint, organizations must really react as if the data weren't encrypted, as the IBTS did by sending letters to all concerned.

Based on this, we might conclude that full disk encryption with a token-based key store is the way forward. If we lost the store but not the token, brute-force attacks would fail to decrypt the data. However, we must also consider how many users would keep the store and the token separately — generally, people keep the usual range of wires, USB tokens, and other associated paraphernalia in their laptop bags, and might do the same with the token. Again, if both store and token are missing, we must react as if the data were lost in cleartext.

Some might go further and suggest using tokens that require biometric activation. Products like this do exist (see www.raidon.com.tw/content.php?sno=0000088&p_id=5), but given that fingerprint readers in particular are vulnerable to well-known attacks,[6] and that a laptop is a good source of its owner's fingerprints, such schemes might not offer as much benefit as initially appears to be the case.

Object-Level Encryption

Most information on a laptop hard-disk isn't actually sensitive (such as copies of executables), so object-level encryption might sometimes be preferred over full-disk encryption. In fact, much stored data these days is imagery, music, and video, rather than really sensitive information. So, assuming we can identify the truly sensitive data, would it make more sense to encrypt only that, and, if so, what issues arise?

Object-level encryption can potentially address many of the requirements we might pose when considering information leakage. If I encrypt only the sensitive data, then I could more easily make the objects themselves "live" on a server, to be downloaded to the portable store only as needed (via some Web interface, for example) and decrypted only while being used (via client software or some Web 2.0 client scripting). Of course, this assumes that we can identify and manage the sensitive data, and absent a fully fledged multilevel secure operating system, data objects are highly vulnerable to users copying (parts of) them into insecure storage via file move/save or cut-and-paste operations.

Object-level encryption, however, offers better object portability — if an object is encrypted, you can move it to a USB stick and back to a laptop without further exposing the data. Depending solely on full-disk encryption means that you're much more likely to expose data during such transitions.

Usability

One lesson we might learn here is that, in real systems, convenience will always win out over security (see www.infoworld.com/article/08/03/06/10NF-data-loss-prevention-problem_1.html). So, unless the "secure" objects are as easy to use as insecure equivalents, the system probably won't succeed in mitigating information leakage risks. Similarly, users need training and guidance — for example, many users might not realize that they're putting sensitive data at risk while working from home.

Although it's useful to compile security recommendations in this area (for example, one US National Institute of Standards and Technology [NIST] report covers issues related to home network configuration for teleworkers[7]), we can't assume that users have, or can take, all "sensible" precautions — a hotel guest can't influence how the hotel network is set up. Having a simple Web site that explains policies and offers guidance should be a minimum for any larger organization; for example, the Rutgers University "RU secure" site (http://rusecure.rutgers.edu/nppi) nicely describes their policies in this area. The bottom line here is that usability always wins over security, but that user education is also never wasted.

Temporary Storage

Many forms of temporary storage, such as browser caches and firewall logs, can also contain sensitive data. For example, while reporting a recent personal firewall bug, the firewall vendor asked me to send them some files they could use to analyze the problem. On inspection, those files turned out to contain a substantial portion of the Windows registry as well as firewall logs going back several months. At that point, I had to decide whether to trust the vendor and their support partners with details of all my recent Web searches. The lesson we can learn is that many traces of our activities are visible on our machines, and although it's easy to derive sensitive information from these traces, it's very hard to expunge that sensitive information. In some cases (related to a company's IPR, for example, rather than customer records), such temporary stores might be the highest priority. Disk-level encryption seems (modulo key-management concerns) to be a good countermeasure here.

Credential Storage

Lastly, we should also consider that some forms of sensitive data typically carried on portable stores are really credentials that enable access to other sensitive systems and data. A typical browser will store various username-password combinations, and although many of those aren't really sensitive, some will be. However, in this case, it's at least possible to cut off access to centrally stored sensitive data after a theft, and logging can determine whether someone has accessed the sensitive data between when the theft occurred and when it was reported.

In many cases, laptops come with vendor-supplied single-sign-on (SSO) type systems that aim to improve this situation. The general argument for such tools is that they claim to do a better job of securing the local credential store and thus improve security, relative to using a browser to store credentials. However, you have to wonder whether it actually makes sense to tie credential storage to a particular portable storage vendor — although a laptop might come with a reasonable set of SSO tools, having to return to the same vendor for your next upgrade isn't desirable, so you should use only SSO tools with good credential export features.

Incident Handling

Regardless of whether organizations use mitigations such as the ones I've discussed, they should plan for incidents that will inevitably occur — even in the best cases, some user will always fail to follow policy.

Handling data loss incidents is clearly a subset of more general computer security incident handling, so many considerations that apply to other incidents (such as malware or denial-of-service attacks) apply here; you can find a good NIST report on general security incident handling elsewhere.[8] However, as a community, we've yet to incorporate handling data loss into mainstream security incident handling, as shown by the fact that the NIST report doesn't consider data loss incidents in detail. (Nonetheless, it's excellent in terms of the generic aspects of incident handling.) A related memorandum,[9] however, does set out general principles and some specific requirements for US government agencies specifically concerned with the loss of PII.

There are differences, however, between the level of costs acceptable to government agencies and, in particular, smaller enterprises (whether commercial or not). Smaller enterprises also have difficulty accessing appropriate expertise, both for incident handling and for choosing mitigations. The best path forward for them might lie in selecting a good security partner with incident-handling capabilities, as is often the case with more traditional network security.

Ultimately, I would make three main recommendations when considering data loss:

- Plan for handling incidents — they will happen.
- Identify sensitive data, and keep that at "home." Where such data might leave your control (such as when being processed by a partner), specifically consider the potential for data leakage before allowing such use.
- Do use disk- and object-level encryption where possible, and with the best key management you can afford; don't assume, though, that this solves the entire problem — it doesn't.

In addition, I would encourage system developers to better consider the trade-offs relating to ease of use and potential information leakage incidents when planning new systems (for example, during database design), selecting vendors, or deploying systems. Although we can't remove the possibility of bad things happening, proper design can reduce the number of occurrences and their impact. Designers should also consider building object-level encryption into systems from the start — even if it appears that most initial data won't be sensitive, organizations often use systems in ways designers didn't plan for, so the ability to turn on object-level encryption could be an important feature as more forms or more capable portable storage become common.

References

1. *Administrative Investigation Loss of VA Information – VA Medical Center Birmingham, AL*, report no. 07-01083-157, VA Office of Inspector General, June 2007; www.va.gov/oig/51/FY2007rpts/VAOIG-07-01083-157.pdf.
2. S. Engleman et al., "You've Been Warned: An Empirical Study of the Effectiveness of Web Browser Phishing Warnings," *Proc. Human-Computer Interaction* (CHI 08), 2008; www.guanotronic.com/~serge/chi1210-egelman.pdf.
3. A. Narayanan and V. Shmatikov, "How To Break Anonymity of the Netflix Prize Dataset," Oct. 2006, http://arxiv.org/abs/cs/0610105v2.
4. "Payment Card Industry Data Security Standard," version 1.1, Sept. 2006; www.pcisecuritystandards.org/pdfs/pci_dss_v1-1.pdf.
5. "Specification for the Advanced Encryption Standard (AES)," Federal Information Processing Standards Publication 197, 26 Nov. 2001; www.csrc.nist.gov/publications/fips/fips197/fips-197.pdf.
6. T. Matsumoto et al., "Impact of Artificial 'Gummy' Fingers on Fingerprint Systems," *Proc. Optical Security and Counterfeit Deterrence Techniques IV Conf.*, Soc. of Photo-Optical Instrumentation Engineers, vol. 4677, 2002, pp. 275–289; www.lfca.net/Fingerprint-System-Security-Issues.pdf.
7. *User's Guide to Securing External Devices for Telework and Remote Access*, NIST special publication 800-114, Nov. 2007; http://csrc.nist.gov/publications/nistpubs/800-114/SP800-114.pdf.
8. *Computer Security Incident Handling Guide*, NIST special publication 800-61, revision 1, Mar. 2008; http://csrc.nist.gov/publications/nistpubs/800-61-rev1/SP800-61rev1.pdf.
9. "Safeguarding Against and Responding to the Breach of Personally Identifiable Information," US Office of Management and Budget (OMB) memorandum M-07-16, May 2007; www.whitehouse.gov/omb/memoranda/fy2007/m07-16.pdf.

Stephen Farrell is a research fellow at Trinity College Dublin. His research interests include security and delay/disruption-tolerant networking. Farrell has a Joint Honors B.Sc. in mathematics and computer science from University College Dublin. Contact him at stephen.farrell@cs.tcd.ie.