IRPO 304 – Portable and Secure Medical Data Management System

Spring, 2006

Project Description

The goal of this project is to design software to support the secure use of portable memory as a medium for transferring secure data. Envision yourself carrying around sensitive data (e.g., passwords, account numbers) on a flash drive that:

1. is secure from unauthorized access, and
2. will not attack another machine on installation (e.g., with a virus).

A flash drive is initialized to contain user data as well as an application that allows these data to be manipulated in a consistent way by various entities. Depending on the entity, different data are visible and changeable.

Example

Assume that the data we are storing are medical records and that the entities that are authorized to manipulate them are the patient, health care professionals, and insurance agents. These users are able to see/manipulate different data, depending on whom they are, for example:

1. The patient can update his personal demographic information, such as his address, his birth date. The patient can also input times that he took certain medications and report their effects.
2. The doctor can input diagnoses and prescribe treatments.
3. The insurance agent can examine the diagnoses and treatments, and specify coverage.
4. A pharmacist may fill prescriptions and recommend dosage levels.

In this example, there are four entities that are entitled to see different/manipulate data depending on their roles. The patient can see all data, but can only change some: he can change demographic data, but not prescription data. The pharmacist can update prescription data, but not have access to all diagnosis data, and so on.

The patient wants to be able to carry around these records on a flash drive. Alternatively, the patient can carry around a stack of paper medical records in a folder. The first option has advantages over the second in that it is more portable, (potentially) more secure, and is (potentially) more easily accessed and manipulated. The second option has the advantage of not requiring any special infrastructure (a USB-capable PC) to access.

We assume that, in the future, all of a patient's medical records will be found online, in some Web repository. However, one of the problems with this solution is that it requires even more infrastructure (a PC with Internet access). Another problem with this solution is that medical records will, in this case, be spread out over multiple data sources, making

them hard to access. This distribution is likely caused by technological reasons (it is difficult to get independently designed databases to talk with each other) and policy reasons (for privacy reasons, records from different health-care providers are kept separate).

Keeping medical records on a flash drive solves these problems. It puts the medical records at the disposal of the patient instead of the health-care providers. The patient can then choose the entities with whom he shares his medical history—a right he reserves.

To ensure security for a flash drive user's IT infrastructure, the flash drive must be guaranteed to not contain any malicious software. With such assurances, the flash drive can be confidently used to as a medium of information exchange.

Proposed Design

This secure portable memory (SPM) system can be broken up into two components: the flash-drive-to-computer interface security component, and the data access and manipulation component. The first component ensures that the flash drive will not attack the host computer, and the second component ensures the integrity of the data that are stored.

Physical Interface Security

Physical interface security is important because it protects the IT infrastructure from attacks through the USB port to which the flash drive connects—USB vulnerability is a well-known problem [Arnfield02][Labmice03].

One proposed solution to this problem is to create a physical filter that controls the data that can be accessed from a flash drive (the "hardware solution"). For example, only images with the GIF or JPG extensions can be read or written. Such filters are available for special-purpose applications, such as digital cameras that allow users to upload images to a PC or printer [citation?] This physical filter will be attached to all PCs that will be in contact with arbitrary flash drives.

The benefit of the hardware solution is that the filter should, in principle, work with all computers (not just PCs) that allow flash drives to connect to USB ports. All a user must do is plug this device into the USB port as an intermediary with the flash drive. Another benefit of the hardware solution is that hardware is harder to hack.

The problem with this solution is that it lacks flexibility. The software on the filter may have to be updated if there are changes to the application that accesses the data. This would require manual intervention.

Second, the use of a physical filter also requires an administrator to remove the filter for all other USB use.

Finally, from a business point of view, filters have minimal costs. A filter must be built and maintained for all its users.

I believe that a more elegant solution is a software-based design. In principle, all hardware can be simulated by software. We can perform whatever filtration tasks using host-resident software. Currently, there are many solutions available for USB-port security, such as DeviceWall [evers05].

The benefit with software-based security is that it is easily reconfigured, and software updates are easy to distribute via distribution channels, such as the Web. The problem with software-based security, however, is that a version of the software must be written for each application. However, we believe that, considering most computers are Windows-based PCs, using a Windows approach, at least initially, is feasible. From a business point of view, once we write one copy of this software, we can distribute it widely at almost no cost.

To determine if commercial solutions are feasible, however, we must know the particular type of filtration that must be done.

Authentication of Drive Contents

The second step of the project is to ensure that the data and code that exists on the flash drive are authentic—that is, it has been accessed and manipulated only by authorized users. This ensures that the data on the drive is authentic and the code has not been manipulated to violate the integrity of either the data or the host machine.

This authentication can be performed by creating a checksum of the drive's contents. That is, whenever the data access application shuts down, we look at the drive's contents and generate a digital signature of its current state. This signature is generated using either a well-known algorithm and a secret password, or with a secret algorithm to ensure that it cannot be generated erroneously. To authenticate the drive's contents, we check this digital signature against the drive's contents. The software to check the drive's contents should be Web accessible.

Encryption of Drive Contents

The data stored on the flash drive should be encrypted in case the drive is lost or stolen.

Data Model

For efficient data exchange, I assume that we should model data using XML. XML is not a data modeling language the way that UML or the relational model are, but, because it is a popular data interchange format, it may be a good idea to consider the ability of mapping stored data to XML as we develop a true data model.

The question is, what kind of data are we going to store? As suggested above, different parts of the data model must be accessible to different entities, depending on their access levels. What data and entities should we consider? How will they use the data?

Project Management

This IPRO consists of many independent parts:

- Physical Interface Security (PIS)
  - Examining the problems related to public USB port access—what attacks are possible?
  - Searching for suitable solutions—what solutions are there?
- Authentication of Drive Contents (ADC)
  - Studying architectures for securing the data on a flash drive—how can we ensure that the data and code have integrity? How can we ensure that confidentiality is protected?
- Application Development (AD)
  - Determining the data models necessary to support the medical records application. See www.cms.hhs.gov/hippaa/geninfo for SNIP.
  - Determining the privacy models necessary to support the medical records application. See www.cms.hhs.gov/hippaa/geninfo and www.hippa.org.
  - Studying competing products—other flash keys, Internet databases. See www.medem.com, ihealthrecord, Personal Health Key by CapMed.com).
  - Writing the application.

These three components can be worked on independently. The most difficult component is expected to be the Application Development. The Authentication of the Drive Contents is likely to be the simplest component.

The PIS component requires an understanding of the operation of the flash drive and how they can be used to undermine the security of the host computer. It is likely that this information is available on the Web as flash drives are very popular and IT managers have likely identified their flaws.

The PIS team should prepare a report about these flaws, demonstrate them, and describe possible countermeasures, including commercially available countermeasures. If no commercially available countermeasure is available or none of them are satisfactory, then the PIS team should implement one.

The ADC team should develop an architecture that will ensure the integrity of the contents of the flash drive. This system will give the user a good idea of whether the contents of the flash drive have been violated. For example, one way to ensure that an application is authentic is to compute a digital signature for the application and to compare this signature to one that can be found on the Web. The ADC team should be able to argue and demonstrate the security of this approach. The ADC team should also ensure that confidentially will not be compromised should the flash drive be lost or stolen.

The ADC team should justify whatever architecture it proposes. A good justification should cite comparable systems.

The AD team is in charge of designing and implementing the application that resides on the flash drive. This project involves designing an application that is useful and flexible. It should implement the privacy standards and should be easy to use. This design must be justified by an examination of competing products and of available standards.

A report should be prepared on medical record models, privacy concerns, and each of the competing products. All of these factors will affect the design of the application.

Each team will give weekly status reports to the team leader and the IPRO instructor. These progress reports will culminate in an end of month report each month.

Deliverables

Each group must document their work in a clear and professional manner. Good writing style and technique must be employed. All information sources should be cited.

If software is created, design documents and a user manual for this software must be maintained. The code must be documented and follow rules of style.

All written artifacts must be combined into a single integrated document.

We will publish a status report each month, and build our integrated document at the end of each month starting at the end of February.

References

Beatrice Arnfield, USB port devices pose security threat, ComputerUser.com, www.computeruser.com/news/02/05/03/news7.html, May 3, 2002.
Labmice.net, USB Flash Drives: Useful Device or Security Threat?, Labmice.net, labmice.techtarget.com/articules/usbflashdrives.htm, December 10, 2003.
Joris Evers, DeviceWall update improves USB port security, CNet News, news.com.com/DeviceWall+update+improves+USB+port+security/2110-1029_3-5916175.html, October 26, 2005
SNIP Medical Records Models, www.cms.hhs.gov/hippaa/geninfo
Medical Records Privacy Issues, www.cms.hhs.gov/hippaa/geninfo and www.hippa.org
Medical Record Internet Databases, www.medem.com, ihealthrecord
Personal Health Key, CapMed.com

Interesting reads

Michael Singer, USB drives to get smarter, CNet News, news.com.com/USB+drives+to+get+smarter/2100-1041_3-5736039.html, June 7, 2005

Notes

January 17, 2006: First draft of this document was prepared by Wai Gen Yee. I would like you to give me whatever references you have regarding this project.
January 19, 2006: Updated by Wai Gen Yee. Added project management and some references.

Meeting Log

1/17/2006: Introduced team members. Described project and debated design alternatives with students.
1/19/2006: IPRO administration day.