Security Considerations for EFT

Introduction

The speed, cheapness and convenience of effecting payments via Electronic Funds Transfer (EFT) has made it a popular technology. However, with speed and convenience come new security threats that are often overlooked, not least because the problems of security in modern computer, and specifically PC- and network-based systems, are not generally appreciated.

The EFT Data Format

A typical EFT transaction file contains basically a list of payments to be made, identified by a recipient branch code and account number, the amount to be transferred, and various additional informational fields. However, for the purposes of effecting payment, fields other than the first three are ignored. For example here is an extract from the *Standard Bank Operational Regulations, February 2003*:

"Unlike cheque law, any name or commentary attached to an electronic payment is completely ignored."

The difficulty that this presents is that a human checking a transaction batch will typically rely solely (in terms of identifying the beneficiary) on the very information that the bank's computers ignore. In fact, the authorising person needs to be confident that the payment is going to the correct branch code/account number, regardless of the display name presented. Incorrect information, whether deliberately fraudulent or unintentional, will lead to payment going to the wrong account.

The problem that this presents is that EFTs are immediate, or at most next-day, and whilst they are quick and simple to send, the process of reversing an error (or fraud) is not. A cheque can be stopped within the holding period, and if the funds have been paid out in advance of that time then it is the bank's problem. With an EFT, once you click send to submit the file, it is instantly your problem and your risk. Even accidental re-submissions of an EFT batch will involve you having to get each payment individually reversed with the consent of the recipient.

Some precautions one can take, relating to the EFT data itself, to reduce the risk of error or fraud are:

- Separation of duties between the capture of EFT beneficiary details and the capturing and release of payments.
- Validation of EFT payment records to detect possible errors or fraud. The software Cirrus Techvue has developed will perform checks to see if multiple payments to the same account number are being made more often than expected, and also to detect if the human-readable name attached to an

account number changes from previous payments to the same account. (Exception lists for both cases can be set up to eliminate false alarms.)

Authorisation

Almost invariably, EFT payment authorisation relies on passwords. Although it is true that forging signatures is not particularly difficult, it nevertheless presents some obstacle, and in many cases a forgery can be detected and proven by expert analysis. With a password, if it is once compromised, the "forgery" is perfect, and it is impossible to prove that a given user did not key it in. There are many ways criminals can acquire passwords:

- Watching a user key it in.
- Keyboard sniffer Trojan software, introduced onto a PC by direct access or via email or web surfing.
- Looking in obvious places for jotted-down passwords.
- Guessing. Perversely, the more rigorous the password rules are in terms of complexity and frequency of changes, the more likely the users will have to write the passwords down to avoid forgetting them, increasing the risk of the previous threat.
- Trickery, or 'Social Engineering'– phoning up and posing as a support person at the bank, for example (people still fall for this).
- User carelessness: say someone is sick and the office urgently needs to process a payment. The password is given over the phone, and not promptly changed thereafter.

Very few people use what is termed two- or three-factor authorisation. The possible factors are something you know (password), something you have (smart card or token), and something you are (fingerprint, retina pattern, etc.). As a broad recommendation, passwords should be combined with, for example, fingerprint recognition, and/or a physical key like a 'dongle'.

Data Transfer

The process of transferring the data to the bank introduces fresh risks. The Internet is now the most popular medium for file transfer, and while it is true that there is the possibility of transactions being intercepted in transit, the required skills and physical access to computers (either on your premises, at an ISP, or at the bank) make this much less of a risk than other areas that are generally overlooked.

Most banks provide some form of client software through which customers can upload files. Generally, the security offered by the software is good inasmuch as once the file has been sent on its way it is extremely unlikely that it can be tampered with. However, most bank-supplied client applications concern themselves only with

protecting the bank from risk, ensuring that the bank can prove that the file they receive and act on is indeed the file sent from the customer. Unfortunately, they do not do anything to help the security at the customer end, and indeed often force an insecure step in the process.

The problem is that usually they are PC-based applications that require the user either to key in information transaction-by-transaction, or to provide a file for uploading.

The first case is not suitable for entry of large numbers of transactions, and in any event introduces another point at which errors (deliberate or otherwise) can occur. Most larger organisations generate the payment data from their own accounting systems, and do not, indeed should not, manually transfer the data to the EFT client.

The second case, providing a file for uploading, typically requires that the file is written to a PC or network drive, and then is imported into the client software. The PC/network environment is notoriously insecure: even in the best case where there is reasonable control over access to drives and directories, it is still the case that anyone with the administrator password has the ability to access, edit and duplicate the transaction files, either before transmission, or for re-transmission.

Ideally, banks should provide one or more of the following:

- An Application Programming Interface (API) to their client software, allowing customers or third-party suppliers to invoke their client directly and pass payment data directly to the EFT client.
- A means for the customer's software to establish a direct connection to the bank system over TCP/IP and transmit information directly, preferably digitally signed (see below).
- A mechanism for customers to digitally sign the transaction file, so that integrity can be assured. The simplest approach would be:
    - A cryptographically-strong hash of the transaction file is calculated (using an algorithm like SHA-1 or MD5).
    - The hash is signed with the customer's private key (RSA or Elliptic Curve, for example).
    - The bank calculates the hash on the transaction file.
    - The bank decrypts the signed hash with the customer's public key, and compares the result to it's own calculation.

  This process provides a means where any dispute as to the content of the file sent to the bank can be resolved provided the parties have kept copies of the files sent and received.

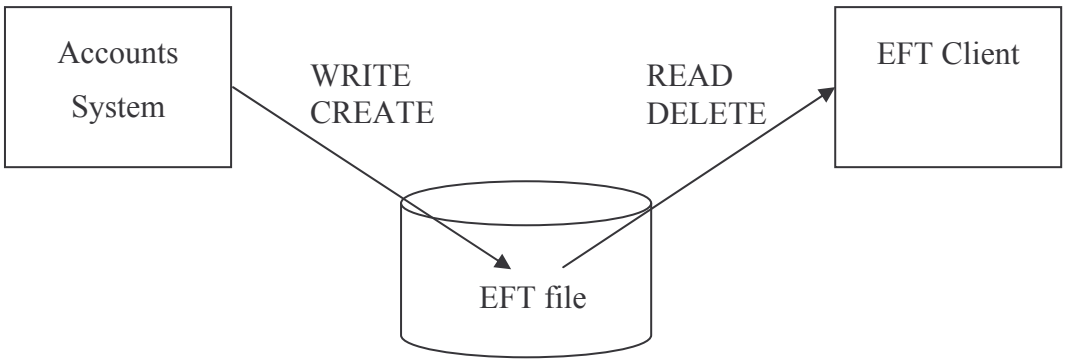
Note that the above deals primarily with providing a way of ensuring the data is not tampered with and certifying the origin of the file. Additionally, files can be

encrypted in transit if confidentiality must also be assured, but confidentiality and data integrity are two separate issues.

Mitigating Data Transfer Risks

Where files need to be uploaded via the intermediate step of a file on disk, steps can be taken at least to make life harder for would-be criminals by setting up access controls on the file locations involved.

Here is a suggested setup:



The workstations/applications would have no permissions other than the ones shown: specifically, no edit permissions.

In cases where the payments are first being passed to Pandora DB (for integrity checking as outlined above, and authorisation), then being passed in turn to an EFT client, the arrangement would be repeated with a second directory, Pandora being able to create and write files, and the EFT client reading and deleting: