The Real *Inside Man*

Brian Hayes Matt Marshall
Redspin, Inc.
{bhayes, mmarshall}@redspin.com

Abstract

The purpose of this effort is to elevate the awareness of the risks associated with automated teller machines (ATMs) connected to a bank's internal network. Given the number and severity of current debit/ATM card incidents, it is time to re-examine existing security controls. While some in the banking industry know that the personal identification number (PIN) is the only encrypted data sent from an ATM, many others falsely assume all data is encrypted [13]. This has only recently become a problem when those that are not aware of that fact are connecting their ATM's to their internal network. Because these networks are often connected to the Internet, this introduces their customer's sensitive data to greater risk [1]. This paper will address vulnerabilities based on this configuration and conclude with several mitigation strategies.

1. Introduction

ATM technology has struggled to keep pace with ATM fraud [10]. Attackers most recently gained access to account numbers and PINs forcing the reissue of hundreds of thousands of ATM cards [11]. Recent ATM changes include strengthening PIN encryption techniques, replacing antiquated operating systems, and upgrading legacy transmission mediums [16]. However, these improvements actually introduce new vulnerabilities involving a fundamental component of original ATM technology that remains unchanged: the message format or application protocol used by the ATM formerly protected by these antiquated operating systems and legacy transmission mediums is now exposed. This paper will explain these architectural changes and how, in combination with an unencrypted protocol, several network-based attacks are now possible.

Section 2 defines the terms and concepts used throughout the rest of this paper. Section 3 discusses both past and present ATM network architectures. Section 4 details the current application protocol used by ATMs worldwide. Section 5 addresses vulnerabilities with this protocol. Section 6 lists possible mitigations for these vulnerabilities, and section 7 concludes this paper.

2. Terminology

Cardholder An individual to whom a debit card is issued. Typically, this individual is also responsible for transactions made with that card.

Issuer — An institution that issues debit cards to cardholders. This institution is responsible for the cardholders account and authorizes all transactions. Banks and credit unions are typical issuers and will be used interchangeably throughout this paper.

Processor — An organization that provides services such as core data processing, electronic funds transfer (EFT), etc. to issuers. EFT allows an issuer to access regional and national networks that connect point of sale (POS) devices and ATMs worldwide. Examples of processing companies include Fidelity National Financial and Jack Henry & Associates.

Visa — An association owned by thousands of issuers.

3. Architecture

Most small to mid-sized issuers contract processors to provide core data processing and EFT/ATM services. Each service typically requires a dedicated data connection between the issuer and the processor. Outsourcing these services allows issuers to provide 24/7 service to their customers without having to support this level of operations. For example, processors allow ATMs to operate even when an issuer is closed.

Historically, ATMs interfaced with processors directly rather than with the issuer that owned the ATM, through a leased or virtual leased line using a proprietary transport protocol. This point-to-point connection made it difficult to maliciously intercept transferred data. Therefore, most issuers had at least two connections to the processor, one for core data processing and one for each ATM the issuer owned (Figure 1). A number of changes have recently been made to ATMs to include an upgrade of PIN encryption strength from data encryption standard (DES) to Triple DES [1] and the replacement of IBM's OS/2 operating system with Microsoft Windows [8]. The use of transmission control protocol/Internet protocol (TCP/IP) as a network transport is now standard as well. Because issuers often run their own networks with TCP/IP, ATMs are being connected to these issuer networks and a single connection

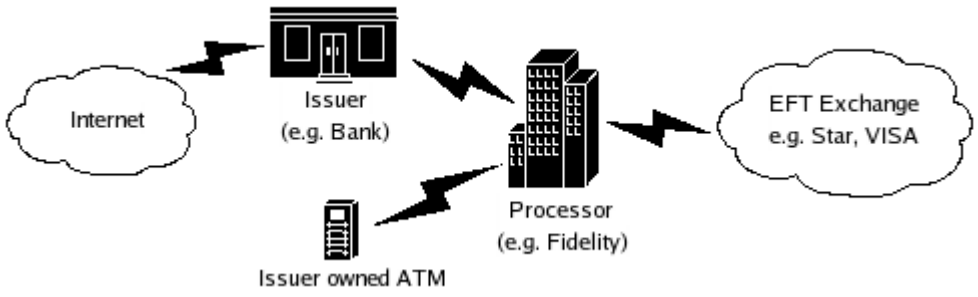

Figure 1. ATM architecture with point-to-point connection to processor.

back to the processor is now utilized for both services (Figure 2). Not only does this save expensive monthly circuit fees, but connecting all the ATMs together provides easier management. The problem, addressed in section 5, is that these issuer networks are often connected to the Internet without proper security controls in place, thereby providing access to ATM traffic by an individual with access to the issuer network, such as an attacker or malicious employee.

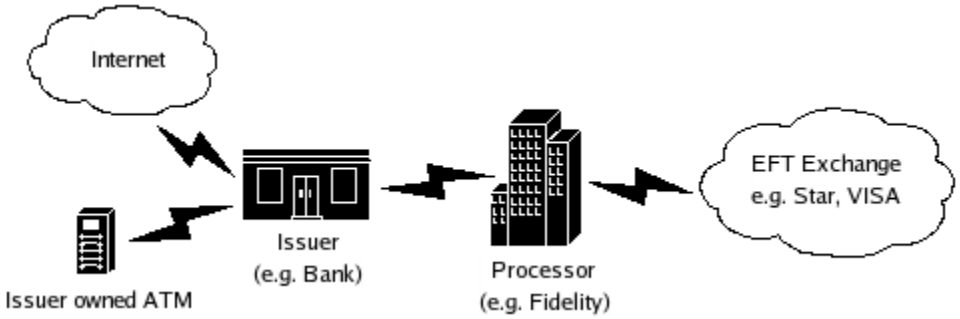

Figure 2. ATM architecture with shared connection to processor.

Processors have connections to EFT exchange networks to allow cardholders access to accounts from any ATM (assuming the processor and ATM are connected to the exchange). The exchange consists of a number of regional networks connected together by national networks; for example, "Visa and MasterCard own and operate the two national EFT/POS networks: Visa's Plus and MasterCard's Cirrus ATM networks and Visa's Interlink and MasterCard's Maestro POS networks. These national networks serve as a bridge between regional networks, and permit transaction information to be routed from one regional network to another [9]." Examples of regional networks include Star, NYCE, and Pulse.

Using the architecture shown in Figure 1, an ATM transaction would begin with a user swiping her card and inputting her PIN. The ATM would encrypt the PIN and send an authorization request to the processor. The processor would answer and update the customer's information. This differs from Figure 2 in that the information sent from the ATM to the processor now traverses the issuers network before being sent to the processor. It is this time on the network that the data is vulnerable. A further look into the application layer protocol will reveal these vulnerabilities.

4. Protocol Analysis

The connection between the ATM and the processor gateway is a single TCP session between two fixed, high-numbered ports. If no transactions occur within a certain period of time, the ATM sends a TCP keep-alive message to the gateway to maintain the connection. The reason for the continuous session is twofold: 1) initiating a TCP connection for every transaction introduces a delay, albeit minimal, for the ATM user,

and 2) a break in this session may indicate a problem with the connection, possibly malicious, such as a “man-in-the-middle” attack.

The complete ATM transaction is typically completed in three packets: 1) request, 2) response, 3) acknowledgment. An example transaction is shown with the card number 1234567890987654 expiring on January 2007 highlighted (Figure 3). The message format is loosely based on International Organization for Standardization (ISO) 8583

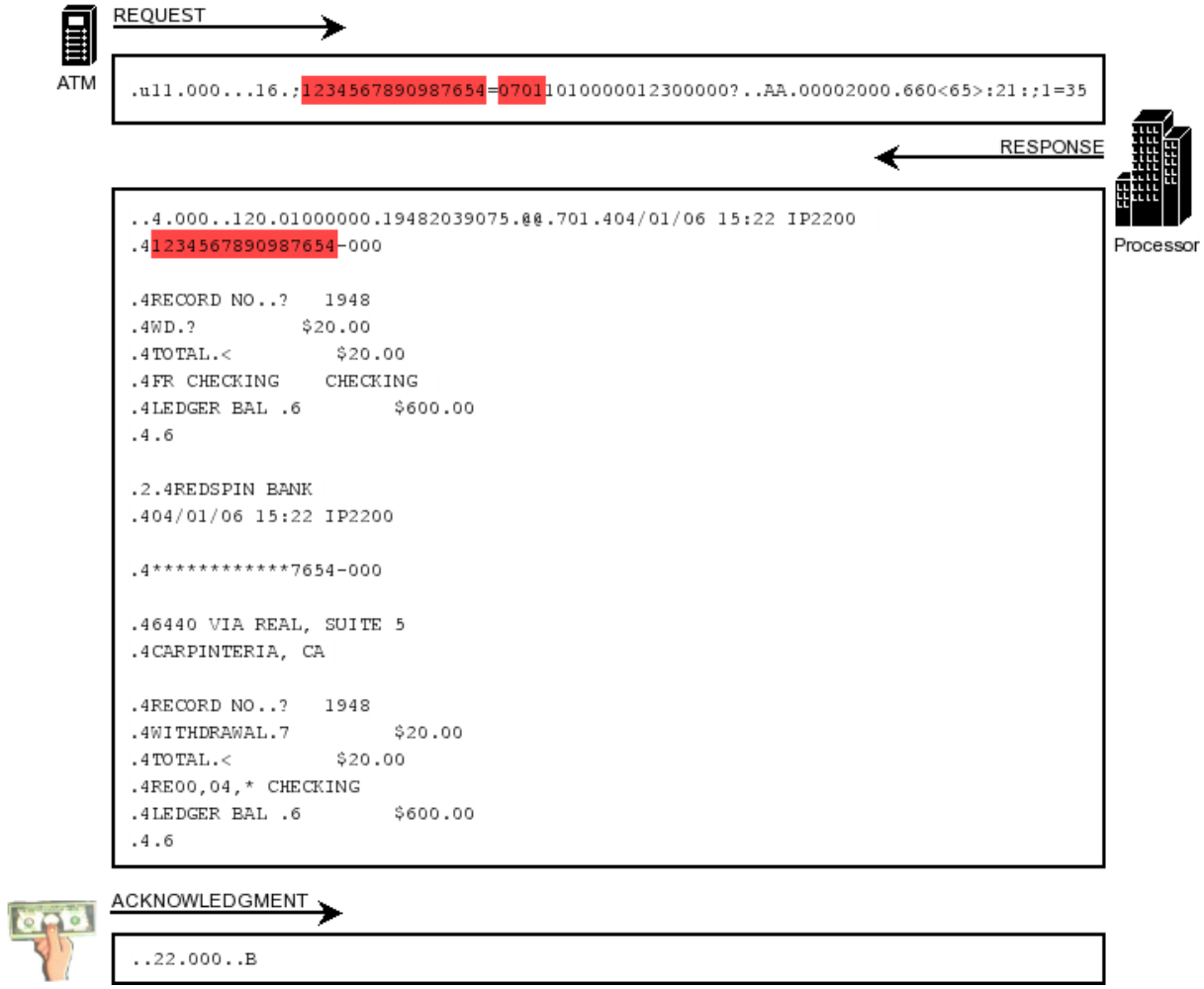

Figure 3. ATM transaction message flow in ASCII

with some fields encoded in a 4-bit scheme (similar to debit and credit cards [3]) and before transfer, 0x30 is added to each byte of those fields, conveniently converting the data to American Standard Code for Information Interchange (ASCII). In the following examples, characters in double quotes are hexadecimal, while single quotes represent ASCII. Also note that while the discussion below does not reveal the entire protocol, the aim of this study is to identify some but not necessarily all potential vulnerabilities.

Request
Although not all fields in the request message format are known, the card account number, card expiration date, dollar amount requested, and the encrypted PIN are easily identified (Figure 4).

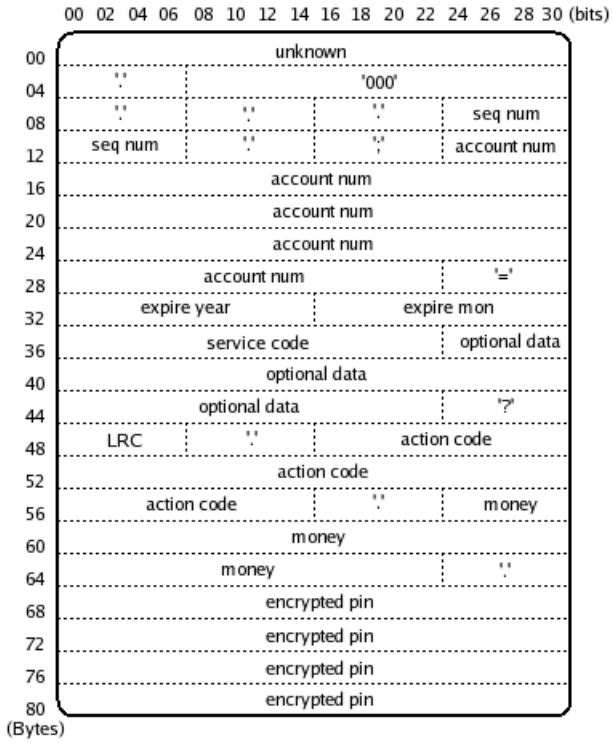

Figure 4. Request message format

The hex value “1c” represents message field separators. Starting at the top, bytes 0 through 7 are not known, but appear fixed and thus may indicate an issuer identifier. Bytes 11 and 12 are sequence numbers that increment by 1 with every transaction. Bytes 14 through 48 are copied directly off Track 2 of the debit card, which includes the account number (15-30), expiration date (32-35), type of card (36-38) such as ‘101’ for Visa, and optional data (39-48). Bytes 50 through 57 specify the request of the client. For example, value ‘AA’ represents a withdrawal and ‘CA’ indicates a balance transfer. In the case of ‘AA’, bytes 59 through 66 represent the amount of the withdrawal. For example, ‘00002000’ represents $20; if $500 were requested then bytes 59 through 66 would show ‘00050000.’ Lastly, bytes 68 through 83 show the encrypted PIN.

Response
The response message reveals the transaction number, the action code and associated data, and often the entire message to be printed on the customer receipt that includes the account balance (Figure 5).

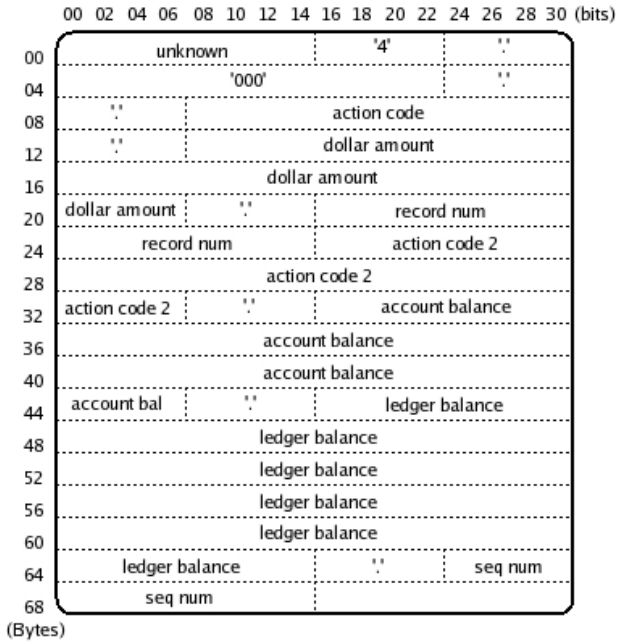

Figure 5. Response message format

Similar to the request, value “1c” represents a message field separator. Bytes 0 through 6 are still unknown but are similar to bytes 0 through 7 of the request message. Bytes 9 through 11 are the action code. Common action codes are shown in Table 1. Given a specific action code, the rest of the packet differs. For example, the withdrawal code of ‘120’ will be followed in bytes 13 through 20 with the dollar amount authorized. A $20 withdrawal authorization is represented as ‘01000000’ and a $500 authorization as ‘25000000’. The response simply specifies the number of $20 bills to disperse. A list of common dollar amounts is shown in Table 2. Bytes 22 through 25 show the record number, followed by another unique number associated with the action code in bytes 26 through 32. Bytes 34 through 44 show a value in ASCII for a balance inquiry or surcharge and ‘@@’ for all other actions. Bytes 46 through 65 are normally null unless there is a balance inquiry, in which case the ledger amount is shown. Bytes 67

Table 1. Codes corresponding to ATM requests.

Code	*Action*
074	Balance Inquiry
118	Surcharge
120	Withdrawal
134	Invalid PIN
154	Declined – Daily withdrawal limit reached

Table 2. Message codes for withdrawal amounts.

Amount	*Representation*
20	01000000
40	02000000
60	03000000
100	05000000
500	25000000

and 68 show the sequence number and following this is the text to be printed on the receipt.

Acknowledgment
All acknowledgment packets are consistent (Figure 6).

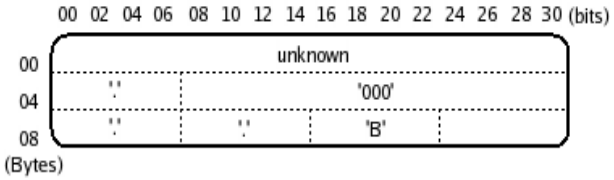

Figure 6. Acknowledgment message format

Bytes 0 through 7 replicate the request message, followed by a “B” in byte 10.

5. Vulnerabilities

Network-based ATM fraud is not a new concept, but rather is becoming easier to execute given the recent move of ATMs to Internet-connected networks:

> There are many types of fraud that can be perpetrated by tapping data communication lines, and using protocol analyzers or computers to intercept or introduce data. These types of fraud are not widespread, mainly because of the need for physical access and because sophisticated computer techniques are required [12].

While tapping data communication lines is still feasible, capturing data on an issuer’s local area network (LAN) is much easier and inconspicuous [2]. Once an attacker is on the LAN, a number of security concerns should be addressed.

Confidentiality
Given the current application protocol, confidentiality of user account information is clearly a significant issue. While the PIN is encrypted, the card number, expiration date and current balance are not. How valuable is that information? It can be used to create a duplicate physical card to be used for signature-based transactions or on-line purchases. It can also lead to identity theft. An attacker can simply passively sniff this information off the wire without setting off any alarms.

The PIN itself if decrypted could be used with the information in clear text to access a person’s account from any ATM. Fortunately, decryption is not trivial. Rather than encrypt a user’s PIN, which could be cracked in time, a combination of their account number and PIN is encrypted and transmitted over the wire thereby preventing most brute force attempts.

Integrity
There does not appear to be any protection in the application protocol to prevent an attacker from injecting or altering data in transit. The TCP keep-alive session in and of itself is not a significant control. There are two problems with the current architecture that benefit an attacker: 1) the data stream between the ATM and processor gateway is assumed to be secure, and 2) the source of the data stream is trusted by both the processor and the ATM [5]. Based on these trust relationships, a number of attacks are possible, one of which will be discussed below.

Perhaps the easiest attack would use a "man-in-the-middle" technique to gain access to the data stream, allowing an attacker to spoof either end of the connection. By spoofing the processor, it may be possible to direct the ATM to dispense money without the processor ever knowing a transaction occurred. Indeed, without the processor enforcing a daily maximum withdrawal limit, it may be possible to empty the entire ATM.

By spoofing the ATM, an attacker could infiltrate existing accounts, the credentials of which were captured on the network. Ironically, the encrypted PIN is more useful on the network than the actual PIN. Once a user's account number and encrypted PIN are captured, their account is compromised until the ATM encryption key is changed – often months away or never. With these credentials, a number of attacks are possible, such as an account balance modification or transfer.

Availability
There may also be availability concerns given the connection of ATMs on TCP/IP networks running the Microsoft operating system, effectively turning them into internal servers that must be maintained. Worms will be continually exploiting new vulnerabilities and denial of service (DoS) attacks will always be an issue [6].

6. Mitigation Strategies

Solutions for the vulnerabilities identified are known and have been recommended for years, and still not always followed [7]. The problem is that implementing many of these controls can be difficult or next to impossible given the deployment of ATM's worldwide.

> There are message authentication, encryption, and key management techniques that are available to combat this type of fraud, but currently these techniques are far more costly than the minimal fraud they could prevent. About the only such security technique that is in widespread use is encryption of PINs [12].

The short-term strategy is to implement controls at the issuer level and ensure attackers coming from the Internet or internal malicious employees have no more access to ATMs than they had in the past. This includes segmenting ATM traffic from the rest of the network either by implementing strict firewall rule sets or physically dividing the networks altogether. Also consider implementing network level encryption between the

routers that ATM traffic traverses. In the long term, the following recommendations will solve many of the vulnerabilities already discussed.

Confidentiality
Protecting confidentiality requires encrypting the current application protocol from ATM to processor or changing the protocol so that sensitive information is no longer transferred on the network. The problem with encryption is that much of the information transferred in clear text has other uses. For example, the first few digits of the card number are used to route the packet to the correct issuer and other fields such as expiration date are used to verify the authenticity of the transaction request. "ANSI Standard X4.13 defines the format of card systems and for Visa. Digits 2-6 are the bank number, digits 7-12 or 7-15 are the account number, and digit 13 or 16 is a check digit." [12] Additionally, a number of services rely on customer information sent in clear text, such as video surveillance systems used to monitor ATM machines, which should be an additional concern allowing third-party vendors access to this sensitive information [4].

Integrity
Ensuring data integrity requires better authentication between the ATM and processor and the elimination of any kind of trust relationship between end nodes. Random sequence numbers are also necessary to prevent replay and insertion attacks. All of these controls would require changing the application protocol.

Availability
Preventing worms and denial of service attacks require better access control and procedures in place to ensure security patches are applied promptly. Utilizing a standalone network for all ATMs is optimal, but if a shared network is necessary, segmentation with firewalls is essential.

7. Conclusion

The recent news concerning a compromise of hundreds of thousands of PINs and customer account numbers [14,15] should alert issuers and processors, indeed the entire banking industry, that network-based attacks are growing in severity. ATMs are not immune from this trend because they share the very same network and similar protocols as POS devices. While this paper has focused on the smaller to mid-sized bank architectures, even larger banks with TCP/IP ATMs are vulnerable without the proper controls. All of these devices use the same application layer protocol with sensitive customer information transmitted in the clear.

This paper has identified the risks associated with the current application protocol, including disclosure of sensitive information transmitted in cleartext, vulnerabilities with current ATM architectures and the use of the encrypted PIN to modify customer accounts. Practical short-term mitigation strategies for issuers to protect customer information have been offered, such as greater network segmentation or encryption.

Works Cited

1. Johnson, JoAnn. "ATMs: Triple DES Encryption." *National Credit Union Administration*, July 2004
2. Fenner, Robert M. "Enhancing Data Security: The Regulators' Perspective." *General Counsel National Credit Union Administration*, May 2005.
3. Visdómine, Luis Padilla. "Magnetic stripe examples: standard cards." 29 March 2004. 7 April 2006 <www.gae.ucm.es/~padilla/extrawork/magexam1.html>.
4. *GE Interlogix*. "Capture ATM Transaction Data and Provide Video Surveillance." 22 November 2002. 7 April 2006 <www.ge-consultantlink.com/docs/apnote_atm_ data_capture.pdf>.
5. Capehart, George. "A Tale of Two Systems.**"** *Open Web Application Security Project.* 27 October 2003. 7 April 2006 <www.owasp.org/columns/george/twosystems.html>.
6. Loney, Matt. "SQL Slammer worm wreaks havoc on Internet." *ZDNet UK*. 26 January 2003. 7 April 2006 <news.zdnet.co.uk/internet/security/0,39020375,2129330,00.htm>.
7. "Payment Card Industry Data Security Standard." *Visa*, December 2004.
8. Robert, Paul. "Windows ATMs raise security concerns." *IDG News Service*. 2 December 2003. 7 April 2006 <www.infoworld.com/article/03/12/02/HNwinatm_1.html>.
9. "Retail Payment Systems." IT Examination Handbook. *Federal Financial Institutions Examination Council*, March 2004.
10. Anderson, Ross. "Why Cryptosystems Fail." 1st Conf.- Computer and Comm. Security '93 – 11/93 – VA, USA c 1993 ACM.
11. Sandoval, Greg. "Web of intrigue widens in debit-card theft case." *CNET News.com*. 11 February 2006. 7 April 2006 <news.com.com/Web+of+intrigue+widens+in+debit-card+theft+case/2100-1029_3-6038405.html>.
12. Ziegler, Joe. "Everything you ever wanted to know about Credit Cards." *Eastland Data Systems*. 7 April 2006 <www.eastland.com/everythingCC.html>.
13. "How Credit Cards Work." *Howstuffworks*. 7 April 2006 <money.howstuffworks.com/credit-card4.htm>.
14. Sandoval, Greg. "Your secret PIN may not be so secret." *CNET News.com*. 16 March 2006. 7 April 2006 <news.com.com/Your+secret+PIN+may+not+be+so+secret/2100-1029_36050259.html>.
15. Gilbert, Alorie. "Retailers feel security heat." *CNET News.com*. 22 April 2005. 7 April 2006 <news.com.com/Retailers+feel+security+heat/2100-7349_3-5680788.html>.
16. McDonald, Ian. "Diebold Could Be a Cash Machine For Investors Willing to Wait It Out." *The Wall Street Journal*. 31 January 2006. 7 April 2006 <webreprints.djreprints.com/1400220913538.html>.