Part I

Anatomy of Payment Application Vulnerabilities

Science in the service of humanity is technology, but lack of wisdom may make the service harmful.

—Isaac Asimov

In This Part

Chapter 1: Processing Payment Transactions
Chapter 2: Payment Application Architecture
Chapter 3: PCI

CHAPTER

1

Processing Payment Transactions

Because people have no thoughts to deal in, they deal cards, and try and win one another's money. Idiots!

—Arthur Schopenhauer

In order to understand the vulnerability points of point-of-sale and payment applications, it is necessary to know the basics—how, when, and why sensitive cardholder data moves between different peers during the payment transaction cycle:

- Why (the reason): Is it really necessary to hold, store, and transmit this data throughout the entire process?
- How (the location and the routes): What are the areas with a concentration of sensitive records?
- When (the timing): How long is this information available in those areas?

Payment Cards

The use of payment cards is obviously one of the main subjects of this book. There are several main types of payment cards commonly used for payments:

The credit card was the first payment card and it is still very common. By paying with a credit card, customers use their available credit and pay the bill afterwards. Credit cards are not usually protected by a Personal Identification Number (PIN), which allows them to be used for online purchases.

The debit (ATM, Cash) card is a relatively new method of payment. It is different from a credit card because the debit cardholder pays with the money available in their bank account, which is debited immediately in real time. A debit card seems to be more dangerous compared to a credit card because the debit card is directly linked to the bank checking account and usually allows ATM cash withdrawals. On the other hand, it is more protected by the required two-factor authentication (PIN number plus card itself). The real dangerous element of many branded debit cards is that they can be processed as credit cards, without entering the PIN.

The gift card is similar to a debit card but usually does not have the protection provided by a PIN. The gift card is not linked to a bank account and normally "contains" fixed amounts of funds. The card itself does not hold any financial information—the point-of-sale (POS) terminal communicates with the gift card provider during payment transactions in order to get authorization. Gift cards are less dangerous than credit and debit cards because only fixed, often very limited, amounts of money can be stolen.

The fleet (or proprietary) card is similar to a credit card but can be used only at particular locations (usually gas stations and convenience stores) and for purchasing only limited types of merchandise (such as fuel and other automobile items). Fleet cards, even though often issued by major card brands, are less interesting to "bad guys" because they cannot be used for ATM withdrawal, online shopping, or purchases in department or grocery stores.

Table 1-1 shows a list of major payment card types and their main features.

Table 1-1: Payment Card Types

CARD TYPE	ISSUED	PURCHASE POWER, $$	ACCEPTANCE	PROTECTED ACCORDING TO PCI DATA SECURITY STANDARDS?
Credit	By banks under payment brands (such as Visa) or directly by payment brands (such as American Express)	Several thousand	Virtually any brick-and-mortar or online merchant.	Yes
Debit	By banks with or without payment brands	Several thousand	Virtually any brick-and-mortar or online merchant; bank ATM .	Only if issued under payment brand

Continues

Table 1-1 *(continued)*

CARD TYPE	ISSUED	PURCHASE POWER, $$	ACCEPTANCE	PROTECTED ACCORDING TO PCI DATA SECURITY STANDARDS?
Gift	By payment brands or proprietary providers	Several hundred	If branded, virtually any brick-and-mortar or online merchant. If proprietary, only particular merchants.	Only if issued under payment brand
Fleet	By banks, payment brands, or proprietary providers	Several hundred	Particular merchants (usually gas stations and c-stores) and limited merchandise types (usually fuel).	Only if issued under payment brand

PCI: Payment Card Industry

Card Entry Methods

There are two main methods used to enter the card data into the POS in order to start a payment transaction: *swipe* and *manual entry*.

MSR

The first method uses a *Magnetic Stripe Reader*, or MSR, which is a device that reads the magnetic stripe on payment cards. Modern MSR devices have encryption capabilities and can be used in point-to-point encryption (P2PE) solutions (see Chapter 8 for more details). The easiest way to enter the card data into the POS is to just swipe the card in the MSR so it can read the magnetic stripe and automatically enter all the necessary information. However, if the magnetic stripe is damaged, the customer or cashier can manually enter the account number and expiration date embossed on the front of the card.

Some MSR devices emulate keyboard input, so swiping the card is equivalent to simply typing numbers and letters on the computer keyboard. Stealing the track data in this case is as simple as sniffing the MSR input by installing a *keystroke logger*.[1]

Pinpad

The second method uses a pinpad. A pinpad, or Point of Interaction (POI) with a built-in MSR, is a more sophisticated device because it has firmware which can be customized for various functions including protection of the card's sensitive data. Most pinpads also have hardware encryption capabilities implemented as TRSM (Tamper-Resistant Security Module). In addition to MSR, POI also includes other peripherals, such as a customer display and keyboard (in addition to the pinpad), for better direct interaction with the customer throughout the payment process.

Key Players

According to Visa, there are five key players in the card payment processing game: *Consumers*, *Merchants*, *Acquirers*, *Issuers*, and *Card Brands*.[2] However, in practice, there are usually more participants. In addition to Consumers, Merchants, Acquirers, Issuers, and Card Brands, there are also *Gateways*, *Processors*, *Software Vendors*, and *Hardware Manufacturers* who facilitate the payment transaction processing.

Before diving into the details of these players, I would like to remind you that the scope of this book is security of POS and associated payment applications which are located in brick-and-mortar stores. Despite the fact that merchants account for a relatively small percentage of the overall payment processing life cycle, their portion of responsibility and risk is incomparably larger than anyone else's share. There are several reasons for this:

1. First, merchants have a very distributed structure compared to others—a typical retail chain may consist of dozens to thousands of stores. Compare this to a processor who may have a few enterprise-scale data centers where it is much easier to organize the security measures.
2. Second, retail stores are public places with all the ensuing consequences for security.
3. Third, most merchants rely on hardware and software vendors as their technology providers (including security) and simply are not ready to accept the fact that they have a technology which is vulnerable by design. When the PC and Internet revolution in the late 1990s started replacing the old cash registers and standalone credit terminals with complex POS systems with integrated payment applications, it also began bringing countless system and network security flaws and eventually made them an inescapable day-to-day nightmare reality for millions of retailers around the world.

Consumer (Cardholder)

It's us. We go to stores, swipe the cards, and pay the bills.

Ideally, consumers are not supposed to care about security beyond keeping their PIN a secret. If the card is lost or stolen, the consumer just wants to call the bank and get a new one. When our card is swiped, our private information is shared with the merchant, whose POS system is supposed to protect our information throughout the process. We rely on modern high-end technologies to protect our plastic money.

In practice, unfortunately, it's not happening. Not all the cards are protected by a PIN. So if the card is lost or stolen, and this fact went unnoticed, the consumer's money can be easily stolen. And when the card is swiped at the POS, the data is not being kept confidential 100 percent of the time, so the bill arriving at the end of month might contain surprising charges.

Merchant

Merchants, such as supermarkets, convenience stores, restaurants, or hotels, are central figures in the process. They make a lot of decisions, both business and technical: what types of payments to accept—credit, debit, or both; what brands to accept; what bank to open a merchant account with; what kind of POS and payment terminal hardware and software to purchase (or lease); and, finally, how to protect the cardholder data. This last decision might sound different and irrelevant compared to others, but this is the reality—merchants must take care of payment data security because other players often fail to do so.

Nevertheless, merchants still take card payments because they want to sell their goods and services. Their POS hardware and software accepts and processes the card information, sends it to their payment processor for authorization and settlement, and eventually receives money on their merchant account.

Acquirer

Acquirers, or Acquiring Banks, authorize the payment transactions and settle them with the card issuers. Payment processors route transactions, based on transaction and card type, to a corresponding acquirer for authorization and settlement. Acquirers regulate the basic merchant discount rates (the fees that a merchant pays for each processed payment transaction).

Issuer

Issuers, or Issuing Banks, maintain the customer accounts and issue the cards to customers. They bill the customers for their transactions and send money back to acquirers so they can pay the merchants. Issuers manufacture the cards and so are responsible for physical security features.

Card Brands

Card Brands, or Card Networks, facilitate the entire process of payment authorization and settlement. Networks, such as VisaNet, maintain connections between acquirers and issuers. Some card brands, such as Visa and MasterCard, do not participate directly in acquiring and issuing, but rather delegate those functions to third-party independent organizations.[3] Other brands, such as American Express, issue cards and acquire payment transactions themselves.

Card brands regulate the payment processing but do not intervene directly in most cases, including security of sensitive cardholder data in the stores. The various card brands founded the PCI Security Standards Council (PCI SSC) which creates and maintains security standards in order to make merchants responsible for payment data protection.

More Players

In addition to the main players in the payment processing game, there are "man-in-the-middle" participants who provide a lot of "extras" to merchants. Theoretically, a merchant might be able to accept electronic payments without these additional organizations by communicating directly with acquirers. In practice, however, given the complexity of the payment processing schemes and the enormous amount of different payment cards and methods, such implementation would be almost impossible without involvement of payment processors and gateways.

Payment Processor

Payment Processors handle the payment transactions between the merchant and multiple acquirers. They also maintain *merchant accounts* where merchants actually receive their money paid by the cardholders for their goods or services.

Processors route payment transactions to the appropriate acquirer based on the payment type and card brand, such as credit, debit, gift, or fleet cards issued by Visa, MasterCard, American Express, or others. Payment processors create financial (transaction) reports for merchants. There are many other helpful functions that payment processors provide. In many cases, however, they cannot provide payment data security to merchants simply because they have no presence at their stores.

Processors may offer extra functions such as tokenization and even point-to-point encryption. However, this often does not resolve the security problems entirely as many merchants use third-party hardware and software to support more than one payment processor. Moreover, tokenization features provided by many processors do not resolve the card data security problem.

CROSS-REFERENCE **See Chapter 3 for more details on PCI.**

In the example shown in Figure 1-1, the merchant may process all credit transactions with processor B, but send gift card transactions to processor C.

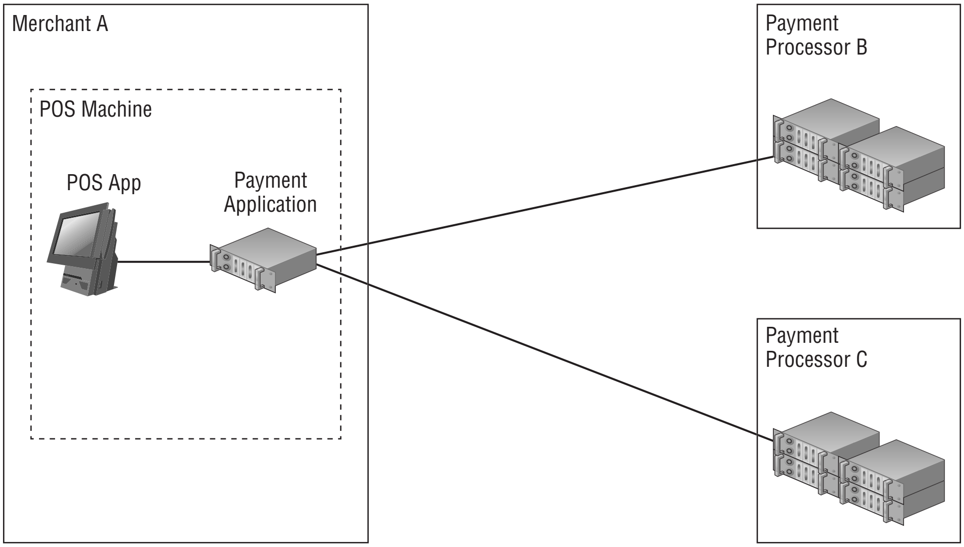

Figure 1-1: Merchant connected to payment processor

Unlike acquirers, payment processors support various payment card types and methods, such as gift cards, fleet cards, Electronic Benefit Transfers (EBTs), and more. They are not limited to credit and debit cards only.

Payment Gateway

In many cases, a merchant's POS payment system talks directly to the payment processor. Sometimes, however, in-between the merchant and payment processor there is another "man in the middle" called Payment Gateway (or Payment Switch). Its primary function is providing *gateway*, or *routing*, services to merchants.

Imagine the situation when merchant A has a service agreement with payment processor B which takes $0.30 plus a 2 percent fee for each processed transaction. Everything is great until the merchant sees a commercial for payment processor C which promises to charge $0.29 plus a 1.9 percent fee per transaction. This apparently small difference could save a lot of money for merchant A who makes thousands of transactions every day. However, in order to switch from payment processor B to C, merchant A must pay the POS software vendor $200,000 for making changes in its payment application so it would be able to communicate with processor C; because originally it was designed to only work with processor B. Figure 1-2 shows that if merchant A were using a payment gateway, such a change would be transparent for its POS software because the routing would be done at the payment gateway's switch that runs in a data center.

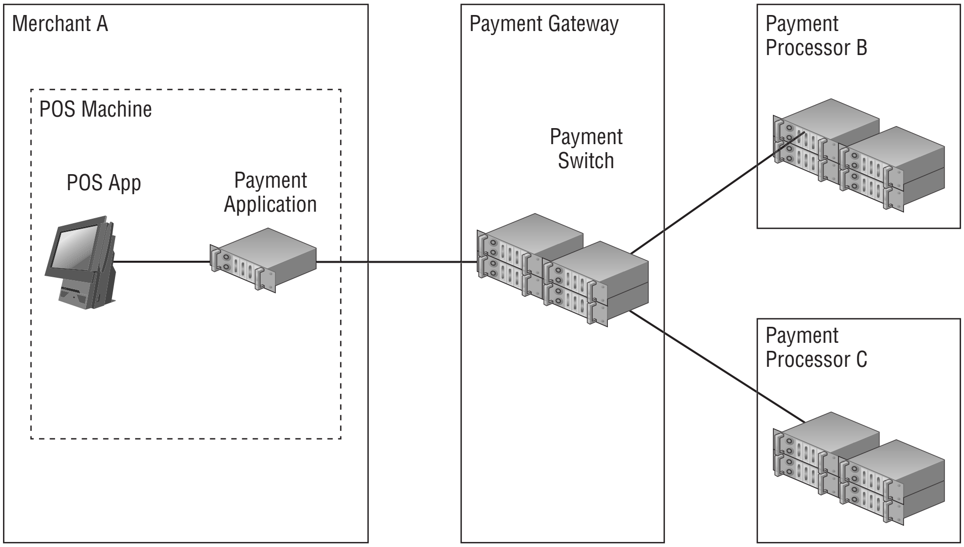

Figure 1-2: Merchant connected to payment gateway

Payment gateways might provide additional convenient services, such as point-to-point encryption, centralized reporting, POI device management, tokenization, and more.

The main difference between a payment processor and a gateway is that the processors, in addition to the switch functions provided by gateways, also maintain merchant accounts and facilitate settlement processes.

Another important role that a payment gateway might play is providing a piece of software that runs at the store and talks with the POS/payment application on one end, and the gateway's switch (server running in the data center) on the other end. In this way, the payment gateway might influence the security of the merchant's payment infrastructure. It can either improve it (for example, by providing P2PE functionality) or damage it (by implanting its insecure component into a previously secure POS system). In most cases, the merchant is still responsible for the security of the gateway's client application running at the store.

Even More Players

Even though the key players such as issuers, acquirers, and card brands are necessary elements in payment processing, the reality is that they have minimal to no influence on security of the payment data in the merchant stores because their part of the process happens “up in the clouds,” far away from the dangerous surface of the retail environment. Processors and gateways are a bit closer to reality because they communicate directly with the stores and sometimes even provide their piece of software running at the POS. However, they still have no control over the situation because their interfaces are just a single fragment of a complex integrated payment environment, and there are other players in the payment processing game who are located right at the front line: payment software vendors and hardware manufacturers.

Payment Software Vendors

Third-party software vendors develop POS and payment applications for merchants. These applications handle (process, transmit, and store) the sensitive data during the entire payment cycle in the retail store, from the moment of card swipe to the settlement. If you take a look at the information brochures provided by payment brands, you will not find software vendors in the list of payment-process players. This is wrong. Software vendors create applications which are installed at the stores and, among other things, are supposed to protect the cardholder data from being stolen. Unlike merchants, application developers are in a good position to invent and implement complex security technologies. If a POS system, which is usually created by a third-party software vendor and not by the merchant itself, fails to protect the cardholder data, the entire payment security fails because retail stores are the main target of hacker attacks.

Payment application vendors are obligated to obey the PCI PA-DSS (Payment Application Data Security Standard), which is not strong and effective enough to protect sensitive data (see Chapter 3 for more about PA-DSS).

Hardware Manufacturers

Hardware manufacturers are another example of misrepresentation of the payment-processing cycle. They create the peripheral devices necessary for transaction processing, such as MSR and pinpads (see more information about these devices later in this chapter). These devices are located at the front line of the payment data security because they accept, process, and transmit the sensitive authorization data, in the form of magnetic stripe swipe or manual entry, at the very first phase of the payment cycle. Pinpad devices must be PCI PTS compliant in order to be allowed to process debit transactions. Both MSR and pinpad devices must be PCI PIN Transaction Security (PTS) compliant in order to be included in P2PE solutions. For all that, their manufacturers are often still not mentioned while describing the payment-processing flow.

Payment Stages

Card payment transactions go through the stages starting with the initial swipe at the POS and ending by the bill being sent to the cardholder. From the entire payment system perspective, there are two main stages of payment processing: *authorization* and *settlement*. From the POS and payment application viewpoints, however, there is more granularity behind these two phases.

Authorization

The very first step in a card payment process is called *authorization*. It is necessary in order to check the cardholder's credit or, in the case of a debit card, check whether the cardholder's bank account has enough funds to process the payment. The authorization flow is shown in Figure 1-3.

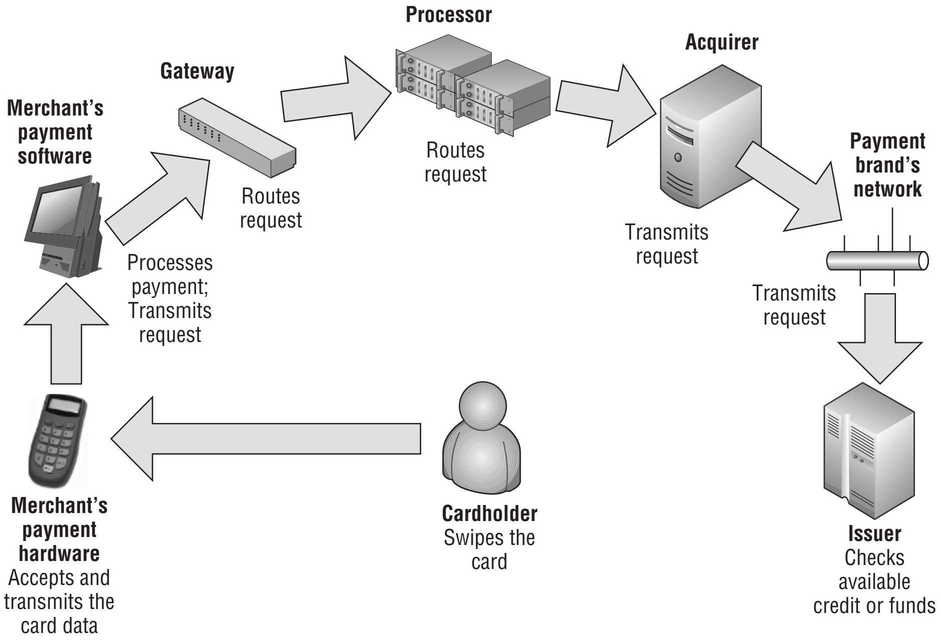
Processor
Acquirer
Gateway
Merchant's payment software
Payment brand's network
Routes request
Routes request
Transmits request
Processes payment; Transmits request
Transmits request
Merchant's payment hardware
Accepts and transmits the card data
Cardholder
Swipes the card
Issuer
Checks available credit or funds

Figure 1-3: Authorization flow

In a retail store, for example, as soon as the cashier finishes scanning the items selected by the customer, he hits the subtotal button which usually switches the POS to payment mode. The customer is then prompted to select the method of payment and, if it is a credit, debit, gift, or EBT card, swipes the card at the card reader, which can be either a simple MSR or a sophisticated POI device. The payment application analyzes the card data and initiates the payment transaction according to card type and Bank Identification Number (*BIN*) *range* (more details about BIN range are discussed in Chapter 4). The next "station" for the payment transaction is either payment gateway or payment processor. The transaction data is routed to the appropriate acquirer which then communicates with the issuer in order to get an *approval*. The issuer is the one who maintains the database with the information about the cardholder's account and checks its status in real time.

If a credit transaction is being performed, the issuer will check the amount of credit and compare it with the transaction amount. If the customer has enough credit to cover the full payment transaction amount, the issuer will return an approval response to the acquirer which returns it to the payment processor and so on back to the POS. In the case of a debit card, the issuer will check the cardholder's bank checking account to make sure the account has enough funds. A similar checkup is performed for gift cards with the only difference that a gift card has no bank account associated with it. Instead, each gift card is linked to a special database record maintained by the gift card provider. In any case, if the customer does not have appropriate credit or enough funds in their bank account or gift card record, the transaction will be *declined* by the issuer, and the decline response will be returned all the way back to the POS, which will display an appropriate message prompting the customer to use a different method of payment.

The authorization stage is most important from a data security viewpoint because it requires sending all available sensitive authentication data (Track 1 or Track 2 or both) from the POS throughout the entire system to the acquirer. Most of the attacks on card data occur at this stage.

Settlement

Once authorization is obtained and the transaction is finalized by the POS system, the payment must be *settled*, which means that the payment is *reconciled* between the merchant, its acquirer, and the issuer. During the settlement, the merchant (or more precisely, its payment system) sends the transaction data to the processor which forwards it to the acquirer. The acquirer or the processor credits the merchant's account, and sends the data to the issuer who posts the transaction to the cardholder's account. Figure 1-4 shows the settlement flow.

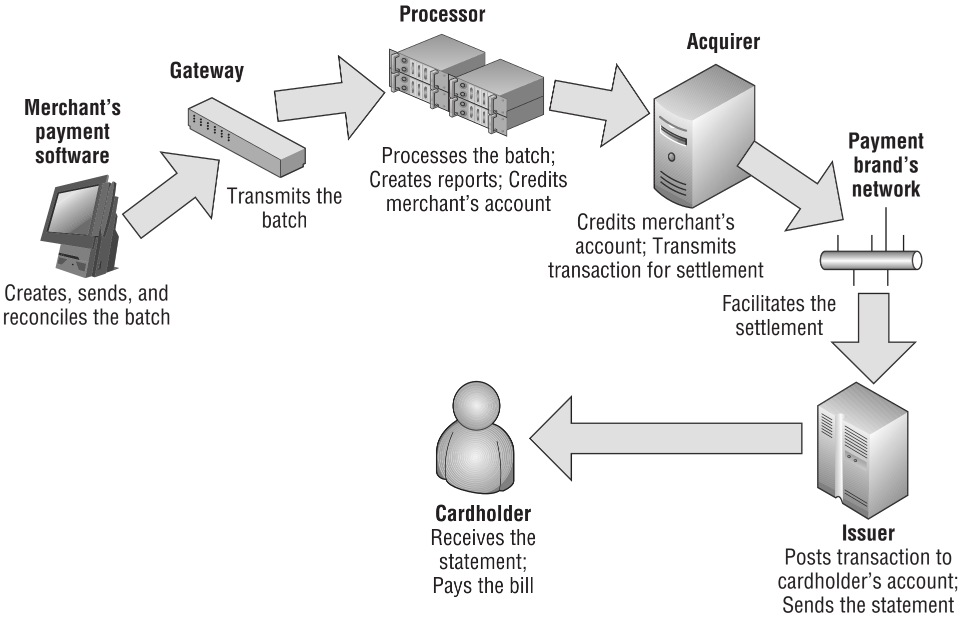
Processor
Acquirer
Gateway
Merchant's payment software
Payment brand's network
Transmits the batch
Processes the batch; Creates reports; Credits merchant's account
Credits merchant's account; Transmits transaction for settlement
Creates, sends, and reconciles the batch
Facilitates the settlement
Cardholder
Receives the statement; Pays the bill
Issuer
Posts transaction to cardholder's account; Sends the statement

Figure 1-4: Settlement flow

From a security viewpoint, settlement is less dangerous than the authorization stage because it usually does not operate with full sensitive authentication data (Tracks 1 and 2). On the other hand, the settlement process requires accumulation of multiple transactions in batches as well as storage of the Primary Account Number (PAN), while authorization data is normally destroyed as soon as approval is received by the POS (Table 1-2). Therefore, in the case of a security breach associated with data storage, information about multiple transactions stored in the batches and awaiting settlement can be "sucked out" in a short period of time. Stealing an equivalent amount of card data during the authorization stage would require long term "listening" to the system.

Table 1-2: Participation in Authorization and Settlement Processing

INVOLVED PARTY	AUTHORIZATION	SETTLEMENT
Cardholder	Swipes the card or keys the account number at POS.	Pays the bill!
Merchant (personnel, network and server infrastructure)	Accepts, processes, and transmits the sensitive authentication data.	Stores and transmits the sensitive cardholder data.
Merchant's hardware (MSR, pinpad devices)	Accepts, processes, and transmits the sensitive authentication data to the payment application.	N/A
Merchant's software (POS, payment application)	Processes the payment transaction; Routes transaction to appropriate payment gateway or processor.	Stores transaction batches; Initiates and processes the settlement through the payment gateway or processor.
Payment gateway	Routes the request to appropriate payment processor or acquirer.	Stores transaction batches; Initiates and processes the settlement through appropriate processor or acquirer.
Payment processor	Sends the request to appropriate acquirer.	Stores transaction batches; Initiates and processes the settlement; Credits the merchant's account
Acquirer	Sends the request to the payment brand's network or issuer.	Credits the merchant's account.
Payment brand	Passes the request from acquirer to issuer.	Facilitates settlement.
Issuer	Checks the cardholder's available credit (credit card) or funds (debit, gift cards); Provides online response.	Posts transaction on cardholder's account; Sends the bill to the cardholder.

Payment Transactions

Each payment transaction has two parameters: *authorization amount* and *transaction amount*. At the authorization stage, the acquirer authorizes the merchant to charge the cardholder for the amount of money which is less or equal to the *authorization amount*. Once the payment is finalized, the merchant sends the transaction to the acquirer for the settlement with the *transaction amount* which cannot exceed the *authorization amount*.

Sale vs. PreAuth/Completion

Depending on transaction and merchant type, authorization can be requested for a particular payment amount or for an abstract *limit amount*. For example, if you purchase groceries in the local supermarket, the POS will likely obtain authorization for the exact amount of your purchase. This "plain" transaction is called *Sale*.

However, if you pay for fuel at the gas station, the payment application first obtains pre-authorization *(PreAuth)* for some predefined "limit" amount which can be either dictated by the card brand or set by the merchant. This is because the POS system does not know exactly how much fuel will be pumped into the tank. Once the fueling is finished, the POS at the fuel pump will calculate and send the exact amount of the payment. Such an additional step in payment processing is called *completion*.

An important difference from a security viewpoint between Sale, PreAuth, and Completion messages is that both Sale and PreAuth contain sensitive authentication information (full track data or PAN), while a Completion message usually contains only the PAN or no card data at all because the transaction was already initiated so Completion can be linked with the original PreAuth using other forms of identification, such as transaction number or token.

Void and Return

It's fair to say that *Void* and *Return* are just the opposite of Sale or PreAuth/Completion. If the payment is done by mistake, or the customer wants to return the merchandise and get their money back, the cashier can initiate a Void or Return payment transaction. There is a difference between Void and Return though. Void is usually triggered when the customer or merchant wants to cancel the entire transaction which might include several items and payment methods. Return, or *Refund*, is normally used when the customer returns a single item and the merchant needs to return only a partial amount rather than the entire payment.

Another important difference (from a security viewpoint) between Void and Return is that Void cannot be performed without a link to the original Sale transaction, while Return can be initiated any time. Void is just a cancellation of a previously existing payment, while Return is placing the money into the cardholder's account without any connection to previous activity. In other words, it is much easier to use Return to steal money from a merchant's account and put it into the bad guy's account. Also, Void transactions (if implemented correctly by payment application vendor and processor) do not necessarily contain sensitive information because the original transaction record already contains the card data.

Fallback Processing

Fallback processing (also referred as *Stand-In*, or *Store & Forward*, or *Offline Authorization*) is a very important function for a merchant's business continuity. It provides the ability to accept card payments "offline" when the network connectivity or payment-processing hosts are down for any reason. If a payment application cannot obtain online authorization from the acquirer, under some circumstances (depending on card type, transaction amount, and other parameters) it is allowed to generate internal approval and store transactions for further processing. Fallback authorization can be almost transparent for untutored cashiers and customers. In many cases, however, some evidence of offline authorization is present and can be recognized:

- Transaction processing time for offline approval can be noticeably longer compared to online authorization because a payment application can be programmed to wait for *response timeout* before it is allowed to approve the payment internally. Response timeout values are defined by payment processors as part of the *message protocol* and may vary significantly depending on connectivity and communication type. However, in most cases the value can be set to several seconds, which is distinctly longer than the several milliseconds required for online processing on fast networks.

  Failover processing (such as *dial-up fallback*) can be another reason for significant delay in obtaining offline approval. Some processors require the payment application to switch to a backup host or different communication line in order to attempt online authorization if the main connection or host is down.

- When a payment application receives online approval for a payment transaction, the host response contains an authorization code generated by the acquirer's software. This code is often printed on the *payment slip* of the transaction receipt. If the POS goes to the offline mode, the payment application generates its own authorization code while approving the transaction offline. Such a code can be generated using a different algorithm (sometimes simply running a counter or current timestamp) and, therefore, it can be easily distinguished by the cashier or customer from the code created by the host. For example:

  Authorization code returned by the host: FVIKP0.

  Offline authorization code generated by payment application: LA1234.

A very important feature of fallback processing is the fact that the POS must store sensitive cardholder data on disk for the entire period of network outage, which may vary from a few seconds to several days. Such a need to accumulate sensitive information in potentially very large amounts opens an obvious opportunity for the bad guys. Normally, if the system is properly designed, the cardholder data is no longer stored at the POS machine after the authorization phase is done. However, in the case of Store & Forward (S&F), the authorization cannot be technically done because there is no communication with the authorization host.

Timeout Reversals

Timeout Reversal, or TOR, is a mechanism that prevents duplicate charges, which is described in more detail in Chapter 2. TOR is another example (after S&F) of a situation when the POS must store locally (even if only temporarily and in encrypted form) the sensitive authentication data which could then be retrieved by an attacker.

Special Transaction Types

There are less common POS transaction types (Table 1-3) that are mostly used either in exceptional situations or when handling special card types. Examples of such transactions are gift card balance inquiry and recharge. *Balance inquiry* is used to check the remaining balance of the gift card, and the resulting transaction data may contain full track data. *Recharge* is used to add funds to a gift card using another method of payment (such as cash or credit card), thus containing sensitive card data information.

Table 1-3: Payment Transaction Types

TRANSACTION TYPE	SYNONYM	FUNCTION	SECURITY CONCERNS
Sale	Purchase	Regular payment transaction (mostly used)	Contains full sensitive authentication data (magnetic Tracks 1 and 2)
PreAuth	Authorization	Checks available balance and obtains authorization	Contains full sensitive authentication data (magnetic Tracks 1 and 2)
Complete	Completion	Finalizes the payment initiated by PreAuth	May contain PAN
Void	Post Void	Cancels previously processed payment	Requires a link to the original transaction; May contain full sensitive authentication data
Return	Refund	Credits cardholder's account (opposite to Sale)	Contains full sensitive authentication data (magnetic Tracks 1 and 2); Can be used to move money out of merchant's account
TOR	Reversal	Attempts to cancel transaction of any type when no response was received from the host	Contains full sensitive authentication data (magnetic Tracks 1 and 2)
Balance Inquiry	Check Balance	Checks available balance on gift card account	Contains full sensitive authentication data (magnetic Tracks 1 and 2)
Recharge	Reload	Adds funds to gift card account	Contains full sensitive authentication data (magnetic Tracks 1 and 2)

Key Areas of Payment Application Vulnerabilities

There are several ways to attack a POS system and its associated payment application in order to steal sensitive card data. Such methods are often called *Attack Vectors* in information security theory. An attack vector usually includes a scenario of the attack—a description about the performed steps and tools used. If you know that a particular invasion is possible, at least theoretically, the particular scenario (for example, penetration methods, specific instructions, and tools used during the attack) is not so important when discussing the application security controls (protection measures). Therefore, instead of attack vectors, there will be a focus on *Vulnerability Areas* of the attacks throughout this book.

Assuming that in the context of this research the target (object) of the attack is always sensitive payment data (or cardholder information), and the environment (subject) is a brick-and-mortar merchant POS and payment application, vulnerability area usually describes the location (both physical and logical) of the data in the application at the moment of the attack. There are three such locations, or states of the data, in any software program including a payment application:

1. **Data in Memory**—When the payment application processes an authorization or settlement, it performs various manipulations with the payment card data in the memory of the hosting computer (usually the RAM of the POS machine).
2. **Data at Rest**—The payment application stores data, either temporarily or for long term, on the hard drive.
3. **Data in Transit**—The payment application sends and receives data to and from other applications and devices.

With the exception of data in memory, other data states have sub-areas determined by a difference in technology around them. For instance, data at rest can be stored in database or log files, and data in transit can be sent via a LAN or serial connection.

Another key vulnerability area is payment *Application Code* itself and its *Configuration* (config). The code or config do not contain any cardholder information by themselves, but can be *tampered* with (modified) by an attacker or malicious software in order to gain unauthorized access to the data in other key vulnerability areas.

With that said, there are four key vulnerability areas of payment applications which are shown in Figure 1-5:

1. Data in memory
2. Data at rest
3. Data in transit
4. Application code and configuration

Table 1-4 lists the key vulnerability areas with all sub-areas. These terms will be used in the discussions about payment application threats and mitigations throughout this book. More details about vulnerability areas and their examples can be found in Chapter 2.

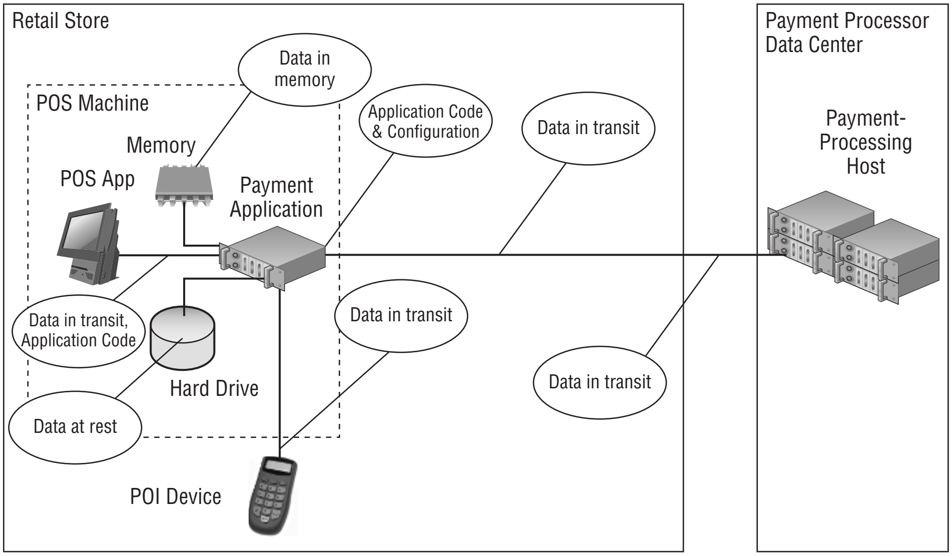

Figure 1-5: Key vulnerability areas

Table 1-4: Vulnerability Areas of Payment Applications

KEY AREA	SUB-AREA	EXAMPLES	TYPICAL DATA	USUALLY PROTECTED?
Data in memory			Full	No
Data at rest	Temporary storage	S&F, TOR, active transaction databases	Full	Yes
	Long-term storage	Batch, settlement, archive records	PAN	Yes
	Log files		Random	
Data in transit	Local communication	LAN between application modules	Full	No
	Communication between POI device and POS		Full	No
	Communication to processors	Host links	Full	No

Continues

Table 1-4 *(continued)*

KEY AREA	SUB-AREA	EXAMPLES	TYPICAL DATA	USUALLY PROTECTED?
Application code and configuration	Application code		N/A	No
	Application configuration		N/A	No

Summary

There are several main types of payment cards: credit, debit, gift, and fleet. Credit and debit cards are the most vulnerable because they are widely accepted and carry significant amounts of money.

There are several participants or "players" in electronic payment processing: cardholder, merchant, software vendor, hardware manufacturer, gateway, processor, acquirer, card brand, and issuer. The merchant is the most vulnerable element in this chain because it faces the public directly, and its interaction with the customers has a significant surface: multiple stores and POS.

The process of payment by plastic card consists of two main stages: authorization and settlement. The authorization phase is more dangerous because it requires transmission of sensitive authentication data, which is often unencrypted, throughout multiple systems. Such data can be intercepted by an attacker and used to produce counterfeit cards.

There are several key vulnerability areas of a POS system and its associated payment application:

- Data in memory
- Data at rest
- Data in transit
- Application code and configuration

Each of these vulnerability areas has its specifics and can be attacked using different methods at different times throughout the payment processing cycle.

Notes

1. USB Keystroke Loggers, Amazon.com, `http://www.amazon.com/s/ref=nb_sb_noss?url=search-alias%3Daps&field-keywords=USB%20keystroke%20Logger`
2. Card Acceptance Guidelines for Visa Merchants, Visa, `http://usa.visa.com/download/merchants/card-acceptance-guidelines-for-visa-merchants.pdf`
3. Merchant Acquirer List, Visa, `http://usa.visa.com/merchants/new-acceptance/merchant-acquirer-list.html`