Realtime publishers

The Shortcut Guide™ To

PCI Compliance and How SSL Certificates Fit

sponsored by

thawte™

Dan Sullivan

Copyright Statement

Chapter 3: What Is Required by PCI Data Security Standards?

The PCI Data Security Standards Council publishes a number of documents for businesses, IT professionals, software developers, and others who participate in implementing the PCI Data Security Standard (PCI DSS). One of these, the Requirements and Security Assessment Procedures (version 2.0), describes a set of requirements for businesses working with payment card data. The document describes a set of high-level requirements organized into six functional tasks:

- Build and maintain a secure network
- Protect cardholder data
- Maintain a vulnerability management program
- Implement strong access control measures
- Regularly monitor and test networks
- Maintain an information security policy

This chapter will describe these requirements in a slightly different structure, organized more around clusters of requirements that would be addressed by different groups within an IT department, for example, developers and systems administrators. These are not hard and fast divisions. Some of the requirements necessitate collaboration between developers, systems administrators, application architects, and application managers. Keeping in mind the need for multiple skill sets, we will discuss the requirements organized around:

- Data collection and storage practices
- Infrastructure security
- Vulnerability assessment
- Monitoring and reporting

We begin with the most basic of tasks: collecting data.

Data Collection and Storage Practices

The PCI DSS includes a substantial number of requirements about storing payment card data. There is no way to sufficiently summarize all these requirements in a brief statement but a crude approximation of the spirit of the requirements is:

> ***Don't store payment card data if you don't have to, but if you do store payment card data, lock it down and don't keep it any longer than necessary.***

Locking down data in this case calls for strong encryption and comprehensive access controls. As we shall see later in the chapter, keeping partial payment account information is another method for protecting PCI DSS-relevant information.

Data Storage and Retention Regulations

The amount of payment card data stored in your applications and databases should be minimized. The PCI DSS recognizes there may be business, legal, and regulatory reasons that payment card data has to be stored. In such cases, the reasons for storing the data should be defined as part of a policy, and that policy should include a justification for storing the data and a specific data retention period. The PCI DSS also requires that you have a procedure in place to actually delete data when the retention period passes. That process should be executed quarterly.

Although there are reasons to store some payment card data, sensitive data should not be stored after the authorization process is complete. The kinds of data that may be retained include:

- Name of customer on payment card account
- The account number
- Expiration date on the card
- Service code

The PCI DSS requirements specify that the full contents of any of the tracks on the payment card should not be stored. Magnetic stripes used in payment cards contain multiple tracks with different information stored on each track. Not storing the full contents of payment card tracks helps reduce the risk that someone could create fraudulent duplicate cards. There are at most three tracks on a payment card. Track 2 was created by the banking industry and includes the following data elements:

- Start sentinel, which is a single character
- Primary account number
- One character separator
- Expiration date
- Service code, which indicates interchange rules, authorization processing, and range of services

- Discretionary data, which may include a card verification value (CVV)
- End sentinel, which is a single character
- Longitudinal redundancy check, which is used by the card reader to verify data was read correctly

The CVV code, also known as the CVC code, is one piece of data that should never be stored. This code is used to validate a transaction when the card is not present at the point of sale; for example, if someone is entering a card number online or providing it over the phone to a sales person. This information is only used for the card authorization process, so there should be no reason to store it after that process completes.

The primary account number (which may or may not be the same as the card number) is protected in a number of ways by the PCI DSS. Unless someone has a need to see the full primary account number, it should be masked when displayed. For example, a receipt might display just the last four digits such as "**** **** **** 1234." This is enough to help those of us that have to track receipts from multiple cards without disclosing full account numbers.

Encrypting Stored Data

When the primary account number is stored in persistent digital form, it must be protected by some form of strong encryption, hash function, or by truncating the primary account number. With strong encryption, you could retrieve the primary account number if you have the decryption key, so obviously those must be carefully protected as well. In the case of a hash function, there is no way to retrieve the original account number from the hash value, but given a primary account number, you could apply the hash to determine whether it is the same value as one previously calculated.

Payment card data may be stored on devices that use full disk encryption. In such cases, the PCI DSS requires that the access control system is separate from the operating system (OS) access controls. Cryptographic keys used for full disk encryption must be secured properly. This entails common sense procedures, such as limiting the number of people who have access to the keys and the number of locations where the keys are stored. Of course, you will need to balance this with the need to ensure availability of the keys in the event of a failure on a device where the keys are stored. Some redundancy is allowed, but the requirements specify that "keys are stored in the fewest possible locations and forms" (PCI DSS Requirement 3.5.2).

Clearly, cryptographic keys are an important part of PCI DSS-compliant processes. Managing them can be challenging. The PCI DSS includes a number of requirements related to managing cryptographic keys:

- Documenting procedures for generating keys
- Securing cryptographic key distribution
- Securing storage mechanisms for cryptographic keys
- Setting valid periods for cryptographic keys and updating keys when necessary
- Retiring keys when the integrity of the key has been compromised
- Not using retired or compromised keys for encryption operations

In cases where keys are manually controlled, the PCI DSS requires that the process depend upon multiple individuals. For example, two persons may have to enter their individual codes to reconstruct a key. This separation of knowledge helps to mitigate the risk of a single employee compromising the key management system. Collusion between employees is still possible, but that can be mitigated in other ways, such as rotating the pairs of employees or requiring three people to reconstruct a key.

Help with Implementing Key Management

The National Institute of Standards and Technology (NIST) has many resources for security professionals, including frameworks for managing cryptographic keys. See the Cryptographic Key Management Project for resources on cryptographic key management.

Encrypting Transmitted Data

When payment card data is transmitted over a public network, such as the Internet, or over a wireless network, it must be encrypted using strong encryption. Strong encryption can be used in protocols such as IPSec and SSL/TLS. Neither protocol *guarantees* strong encryption, though. Clients and servers negotiate the type of encryption and key length they will use for a session. It is important that you ensure strong encryption is used when encrypting transmitted payment card data.

When payment card data is collected from a Web site, the Web application should use SSL. The use of SSL is indicated by the presence of https:// in the URL as well as through indicators such as lock icons.

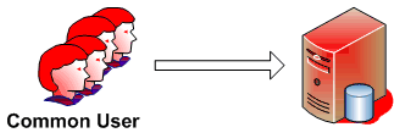
Implementing Logical Access Controls

Access to payment card data should be restricted to those who have a need to know such information to perform their job responsibilities. It follows from this idea that the access controls should enforce:

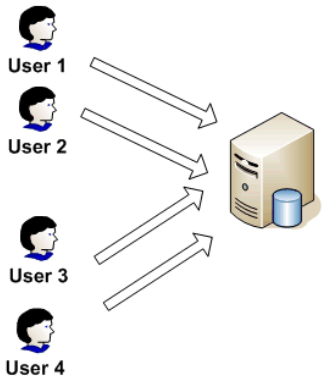
- Only the least privileges required to perform the job
- Privileges are determined by a person's role and function in the organization
- The need for proper approval before privileges are granted
- The privileges that are granted are enforced by an automated access control system

Access controls should be applied to all system components. For example, application functions, directory access, and so on should be limited by access control privileges tied to a user's function in the organization.

It is not enough to restrict access to a functional role. For example, you should not create a single application administrator user such as APPADMIN that is used by multiple application administrators. This practice makes it difficult to track which of the application administrators performs a particular operation. The PCI DSS specifically states that all users must have unique IDs before being granted access to payment card data or any component of the cardholder data environment, regardless of whether that person has actual access to cardholder data.



(a) Multiple users with a single account



(b) Multiple users with multiple accounts

Figure 3.1: Using a single account for multiple users, as in (a), is not allowed by PCI DSS; instead, each user must have individual accounts, as shown in (b).

In addition to having a unique identifier, users are required to authenticate using a password, a smartcard, or a biometric device. (When passwords are stored or transmitted, they must be strongly encrypted.) In cases where users access payment card data remotely, the PCI DSS requires that the user present two forms of authentication, such as a password and token. The two factors must be different types, that is, prompting the user for a password and a security question is not considered a valid combination for two-factor authentication. A better option is to use a combination of something you know (for example, a password), something you have (for example, a token), or something you are (for example, a fingerprint).

Procedures must be established to manage users who are granted access to payment card data. This includes sound procedures for examining a person's credentials and identifying material before granting access or resetting a user's password. Terminated employees must have their access privileges revoked immediately. In cases where user accounts are inactive for 90 days, the accounts should be disabled.

When a user is first granted access, their password should be set to prompt the user for a new password during the initial use of the account. When temporary access is granted to a vendor for remote access, those accounts should be monitored during use. They should also be removed or disabled after the time period they are needed.

Users should be made aware of policies and procedures regarding access controls:

- Limits on their access privileges
- Password policies, such as password strength and the lifetime of a password (90 days at most)
- Not to share their username and password, or other authentication device, with anyone else

Passwords on accounts with access to payment card data should be at least seven characters long and include both alphabetic and numeric characters. When passwords are reset, the new password cannot be the same as any of the previous four passwords on the account.

Login attempts should be limited to at most six failed attempts. After six failed attempts, the account should be locked out for at least 30 minutes or until an administrator resets access to the account.

Once a user has logged into a system, the user should remain logged in only as long as needed to perform required tasks. If a user is logged in but idle for more than 15 minutes, the user should be logged off the system or prompted to re-authenticate.

For most users, access to information in databases should be restricted to access through applications. Only database administrators should have direct access to database queries. Databases and applications should be configured to enforce this policy. In an extension to the one user-one account principle, applications should have unique identifiers for accessing databases as well. Individual user IDs should not have application access to databases; these should be separate application user accounts.

Implementing Physical Access Controls

Someone with physical access to systems can present a security risk even when logical access controls are in place. Someone without logical access to an application may still be able to steal removable media or pick up hard copies of cardholder data. The PCI DSS specifies several physical access control requirements to address these risks.

The physical access controls begin with access to buildings or rooms with information systems housing payment card data. The PCI DSS specifies that sensitive areas should implement a controlled access mechanism, such as magnetic badges, and monitor those areas with video surveillance. Sensitive areas are defined to include:

- Server rooms
- Data centers
- Rooms with equipment that store or process cardholder data
- Rooms with equipment that transmit cardholder data (with the exception of point of sales systems)

Access to networks or network equipment outside of sensitive areas should be protected as well. For example, access network jacks should be restricted to prevent unauthorized access to networks that might transmit payment card data. Similarly, physical access to wireless access points should be restricted to mitigate the risk of tampering.

Of course, there will be times when visitors will be present in sensitive areas or will have physical access to networking equipment outside of sensitive areas. A number of practices must be in place for such situations, according to the PCI DSS:

- Developing procedures to granting visitor badges and tracking their use
- Revoking badges
- Changing access privileges granted to visitors
- Ensuring the integrity of the badge system by limiting access to such systems to only those who need access
- Designing badges to allow visitors to be readily distinguished from regular employees

Visitors should be authenticated and granted badges before they are given physical access to sensitive areas. This should include recording information about the visitor in a visitor log. The log should include:

- Visitor's name
- The organization represented by the visitor
- The onsite personnel who is requesting the visitor be granted access

The PCI DSS requires visitor logs to be retained for at least 3 months.

Access Controls and Removable Media

Physical access controls are defined for removable media such as backup tapes. Backup media should be stored in a secure off-site location. When sensitive data is sent to an off-site location, it should be moved by a secure courier. Logs should be kept describing the movement of media with sensitive payment card data. The security procedures at the offsite location should be reviewed annually.

When storage media with sensitive information is no longer needed, it should be destroyed. In the case of paper copies of sensitive data, the paper should be shred, burned, or otherwise destroyed so that the data cannot be reconstructed. When the data is stored on digital media, the media can be rendered unreadable by eliminating magnetic fields (degaussing) or using a disk wiping procedure to overwrite data on the device.

> **Note**
>
> The name of the popular open source overwriting program, Darik's Boot and Nuke (DBAN), gives an indication of how thorough some overwriting programs can be.

Much of the PCI DSS deals with data collection and storage practices. For more on these topics, see the primary source for this chapter, the Payment Card Industry Data Security Standard: Requirements and Security Assessment Procedures, version 2 and in particular, Requirements 3, 4, 7, 8, and 9. Another area that receives significant attention in the PCI DSS is infrastructure security.

Infrastructure Security and the PCI DSS

The PCI DSS requirements with regards to infrastructure security address the need to maintain a secure network, deploy antivirus programs, and maintain policies that address information security issues.

Maintaining a Secure Network

With headline news stories about retailers and online service providers compromising millions of cardholder accounts in data breaches related to network intrusions, it is no surprise that significant requirements address network security issues. The PCI DSS starts with requirements specifying how firewalls and routers should be configured in networks that transmit payment card data. These requirements include developing formal procedures for configuring and changing firewalls and routers. Proposed changes should be tested and documented.

Firewalls must be deployed between an internal network and a DMZ as well as between a DMZ and the Internet.

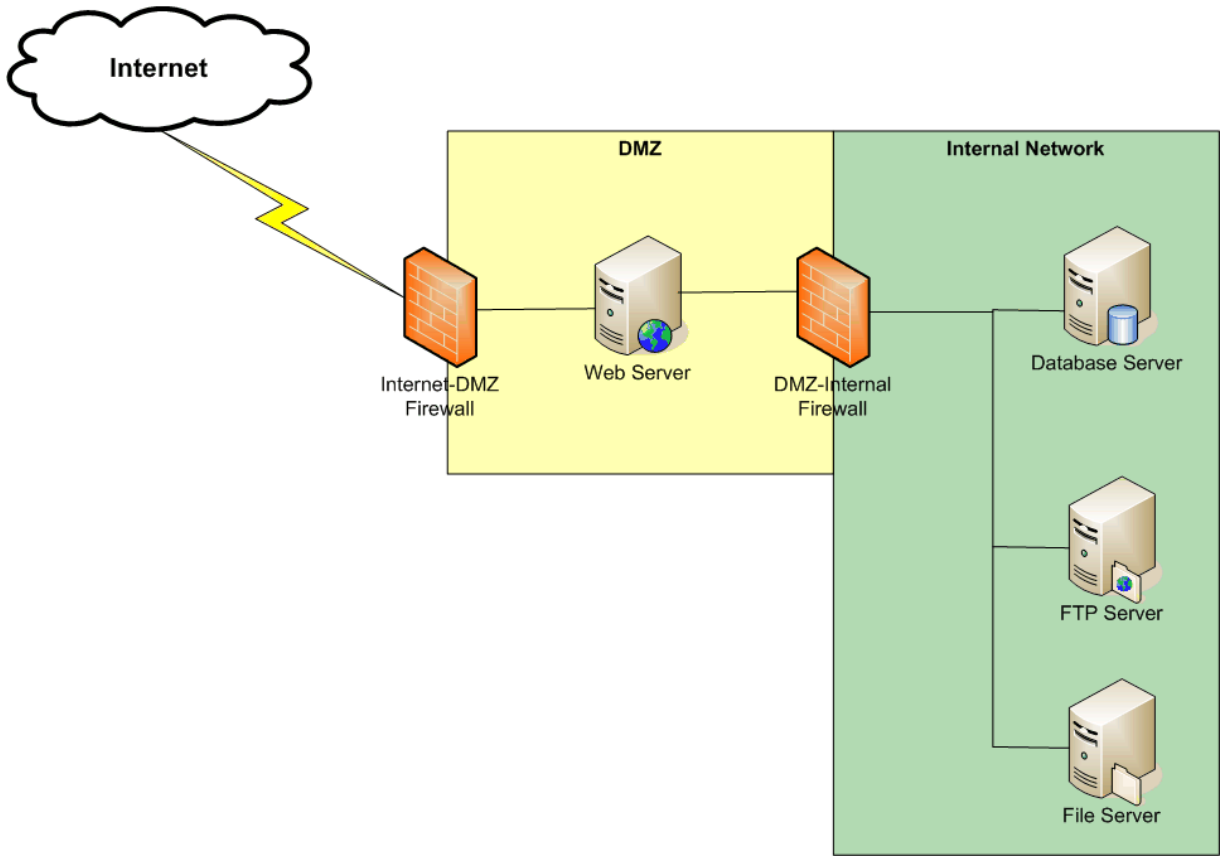


Figure 3.2: Firewalls are required to partition the DMZ from the trusted internal network.

DMZs should be in place to prevent direct access between the Internet and networks processing cardholder data. Traffic from the Internet should be limited to network addresses in the DMZ. Traffic from the trusted internal network to the Internet should be limited to authorized traffic that is needed for well-defined business requirements. Databases and other devices that store payment card data should only be located in a trusted internal network and not in the DMZ. IP addresses of devices on the internal trusted network should not be disclosed.

Documentation about firewalls and routers should specify the business justification for the particular configuration used (for example, why particular ports are open). It should also include mitigation strategies for insecure protocols, such as FTP and Telnet. Firewall and router configuration should be reviewed at least once every 6 months.

In general, firewalls should be configured to prevent inbound traffic from external, untrusted networks, such as the Internet, to networks housing systems working with cardholder data, unless that traffic is required. Firewalls should be placed between wireless networks and networks supporting card data processing as well. Network devices should be configured with denial rules for all unnecessary inbound and outbound traffic. Configuration files for firewalls and routers should be consistent across those devices.

Consumer devices owned by employees, such as mobile devices and laptops, that have access to the organization's network should run personal firewalls configured to meet the organization's requirements. In addition to protecting your network with these and other measures, it is important to secure servers and other devices on the organization's network.

Server Hardening to Meet PCI DSS Regulations

Server hardening is a multi-step process that includes:

- Changing default access control settings provided by a vendor
- Minimizing functionality on servers
- Encrypting administrative operations performed on non-console devices

Applications and devices sometimes arrive preconfigured with default passwords. Wireless routers, for example, may broadcast the default name of the wireless network as the vendor name. (How many times have you listed available wireless networks in your area only to find a neighbor who never bothered to change their wireless access point configuration?) Installing wireless and other devices with default vendor configurations is rarely a good idea, even in a home environment. It definitely should not be done in a professional environment and certainly not one in which payment card data is processed.

Servers and other devices on a network processing payment card data should minimize their attack surfaces. This means that unnecessary programs, daemons, and services are shut down and their application code removed. Servers should be deployed to serve a single function, such as a database server or a file transfer server but not both. Software components not needed for that purpose should be removed. Be sure to minimize applications installed with the OS. For example, only development servers should have compilers installed.

Virtual Servers vs. Physical Servers

When virtualization is used to run multiple virtual servers on a single physical host, each virtual server should perform a single function. Multiple virtual servers running different functions can be run on the same physical server. Be sure to harden the virtual servers as well as the host system providing the virtualization environment, for example, Windows Server running Hyper-V or VMware virtualization environments.

The steps required to harden a server will vary by OS and applications. Minimizing the attack surface on Linux will require different steps than doing so on a Windows server. Applications with a modular structure, such as a Web server, or a complex set of features, such as a database server, can also be hardened. Again, different types of applications will require different steps. Microsoft Internet Information Services (IIS) server and the Apache Web Server both support optional modules that should be reviewed and minimized. Databases, such as Microsoft SQL Server or the Oracle relational database, should have only the components required to meet business requirements. When servers are administered remotely, all communications should be over a secure protocol such as SSH.

Deploy Antivirus Applications

Malware such as viruses, Trojans, worms, keyloggers, and rootkits can find their way into a network through various entry points, including email, removable media, and compromised Web sites. It is important to deploy antivirus software capable of detecting, removing, and protecting against known malware threats. Since malware developers are frequently changing existing malware and developing new malicious code to exploit newly discovered vulnerabilities, it is important to keep antivirus programs up to date.

The PCI DSS requires verification that antivirus programs are constantly running, so be sure to generate logs from these applications. The software should be configured to perform periodic scans as well.

Develop and Enforce Security Policies

Security practices like those described here to protect information infrastructure have to be organized and coordinated, otherwise we risk missing important elements or losing track of what is done in practice. The PCI DSS defines an extensive set of policies that should be in place to protect payment card data. These include policies that:

- Address PCI DSS requirements
- Ensure formal risk assessments are performed
- Define operational security procedures
- Define acceptable use policies for technologies that are used to process or could compromise payment card data, such as wireless communications, consumer devices, and email
- Define authentication, authorization, and related policies for these technologies
- Assign information security responsibilities to a person or team

- Define security responsibilities for individuals in the organization
- Develop security awareness training
- Screen personnel to mitigate the risk of an insider-based attack or data breach
- Employ adequate security procedures with business partners working with payment card data
- Develop incident response procedures to deal with data breaches or other compromised security incidents

The combination of maintaining a secure network, deploying antivirus applications, and developing a comprehensive set of security policies and procedures constitute the group of requirements related to infrastructure security. For more on these topics, see the primary source for this chapter, the Payment Card Industry Data Security Standard: Requirements and Security Assessment Procedures, version 2 and in particular, Requirements 1, 2, 5, and 12.

Vulnerability Assessment and Management

The wide array of applications that run in today's business environments present ample opportunities for attackers. Every program is a potential point of entry for an attack, if the application has a vulnerability that can be exploited to provide unauthorized access to data or functions, allows the attacker to elevate his or her privileges in the environment, allows the attacker to install malicious software, or allows him or her to collect additional intelligence about the network and its applications. This is quite a list and that is why the PCI DSS puts so much emphasis on developing and maintaining secure applications.

The PCI DSS requires that businesses apply vendor-provided patches and that security patches be applied within a month of their release. IT professionals should also monitor newly-discovered vulnerabilities and assess their importance. Public vulnerability databases often specify a severity level when vulnerabilities are discovered, so you can identify top-priority vulnerabilities using these assessments.

Software should be developed following industry best practices. These practices include:

- Removing custom accounts, passwords, and related information and access points before releasing code for use
- Performing code reviews on custom code to mitigate the risk of backdoor code or other malicious code in the application
- Keeping development, test, and production environments separate
- Following a policy of separation of duties with developers and application administrators
- Using realistic but not real data for testing

- Formalizing change control procedures to manage security patches to custom software
- Following coding guidelines to avoid common coding patterns that lead to vulnerabilities, such as injection attacks, buffer overflows, cross-site scripting, improper access controls, and improper error handling
- Using vulnerability scanning tools, especially on public Web sites, to test for emerging threats

Insecure custom applications are not uncommon. There are many ways to compromise an application. Fortunately, there are guides documenting best practices for developing secure Web applications. The Open Web Application Security Project has developed security guidelines for developers and architects. The latest version is under development; it includes a wide range of advice covering topics such as:

- Security architecture
- Authentication
- Session management
- Access control
- Input validation
- Output encoding
- Cryptography
- Error handling
- Data protection
- Communication security
- HTTP security
- Security configuration
- Malicious code search
- Internal security

These guidelines can help new and ongoing development efforts to reduce the risk of developing vulnerable code. For existing applications, especially those running in production, use vulnerability scanners to probe for known vulnerabilities in your applications. Some commonly-used vulnerability scanners now include support for PCI DSS-oriented vulnerability scanning and reporting.

For more on these topics, see the primary source for this chapter, the Payment Card Industry Data Security Standard: Requirements and Security Assessment Procedures, version 2 and in particular, Requirement 6.

PCI DSS Monitoring and Auditing

A key element of the monitoring process for the PCI DSS is the ability to track users' activity, including logging in and detecting unusual events. Logs should provide sufficient data to link activities in systems to specific users. This is especially important when monitoring administers with elevated privileges.

The PCI DSS specifically requires logging and monitoring that includes the ability to reconstruct sequences of significant events, such as:

- Access to payment card data
- Operations performed with administrator or root privileges
- Access operations to audit trail files
- Attempts to access resources in violation of logical access controls
- Login attempts and other uses of the authentication system
- Initialization of audit logs
- Creation and deletion of system-level objects

When capturing these events in audit logs, the entries in the logs should include sufficient information about the event to allow a reviewer to develop a reasonably comprehensive understanding of who performed the action, when it occurred, and what was done. The attributes that should be logged include:

- User name or other identification
- The type of event that occurred, for example, attempt to access a file
- Date and time of the event
- An indication of the success or failure of the event

As many applications are distributed over several machines, it is important to keep the system times synchronized across servers so that timestamps in different logs can be compared.

Obviously, audit trails must be safe from tampering to be of use. Audit trail files should have limited access and should be protected against tampering. Audit trail files should be backed up and file-level message digest should be calculated to help identify tampering should it occur.

Log file should be review daily. Audit files should be kept for a year.

The PCI DSS also requires regular testing in the production environment to mitigate the risk of a security breach. Specifically, these requirements include:

- Testing to detect unauthorized wireless access points on the network
- Running vulnerability scanners at least once every 3 months; this includes internal scans as well as scans by an approved, external scanning vendor
- Running vulnerability scanners after making substantial changes to the environment
- Performing application- and network-level penetration testing at least once per year or after making substantial changes to the environment
- Monitoring for intrusions with the use of intrusion detection systems
- Using file integrity monitoring applications to detect file tampering

For more on these topics, see the primary source for this chapter, the Payment Card Industry Data Security Standard: Requirements and Security Assessment Procedures, version 2 and in particular, Requirements 10 and 11.

Summary

The PCI DSS requires a broad array of controls, procedures, and policies to ensure that payment card data is protected. The requirements include specifications on how to manage data collection and storage practices, how to secure infrastructure, ways to manage vulnerabilities, and how to monitor, report, and audit the state of systems processing payment card data.