Realtime publishers

The Shortcut Guide™ To

PCI Compliance and How SSL Certificates Fit

sponsored by

thawte™

Dan Sullivan

Copyright Statement

Chapter 4: PCI DSS Compliance Checklist

The Payment Card Industry Data Security Standard (PCI DSS) contains more than just recommended best practices—they are required policies, procedures, and technical requirements for businesses that use payment cards. The PCI Security Standards Council provides a number of documents that describe requirements in detail along with FAQs and guidelines; these are available in the council's documents library. This chapter highlights key compliance areas with a focus on the use of SSL certificates, including:

- Self-assessment activities
- Security of servers and data transmission with SSL
- Essential security policies related to SSL certificates
- SSL certificate life cycle management

Before delving into the details of how to use SSL certificates to meet PCI DSS requirements, we will discuss the self-assessment process.

Self-Assessment Activities

For all but the smallest merchants, self-assessments are required annually, and they are highly structured and must include the detailed testing procedures prescribed in the PCI documentation. In this chapter, we will consider elements of self-assessments. This is not a complete discussion but considers five broad areas:

- Data collection and storage practices
- Network security practices
- Server and application security practices
- Monitoring practices
- Policies and procedures

The goal of the following discussion is to consider the main requirements in each area. These checklists are not a substitute for an audit; instead they are intended as a guide to important topics that will be tested during an audit.

Data Collection and Storage Practices

Data collection and storage practices cover how you handle payment card information, including account numbers and authentication data. The general rule is that you should not store data unless it is needed for a well-defined business reason and when data is stored, it should be encrypted.

As part of the self-assessment, you should verify the following requirements for storing payment card data:

- Determine if any authentication data is stored for extended periods of time. If authentication data is stored for extended periods, storage must only be done by payment card issuers or companies that support issuers. Merchants may never store authentication information after authorization.
- Determine whether any authentication data is stored temporarily and is then deleted. If so, verify that data is unrecoverable. This task can include verifying that data blocks that contain temporary data are overwritten sufficiently to prevent recovery. If relational database management systems are used to temporarily store authentication data, verify the data is unrecoverable from rollback segments, transaction logs, and other data structures used to implement data integrity and availability services of the database.
- Verify that the full contents of the magnetic strip of a payment card are not stored. All output from applications that process payment card data should be checked.
- Verify that the card verification code (CVV2, CVC2, CID, or CAV2) for payment cards is not stored. All output from applications that process payment card data should be checked.
- Verify that the personal identification numbers (PINs) are not stored, either in native form or as encrypted PIN blocks. All output from applications that process payment card data should be checked.
- Limit the use of sensitive payment card data in debugging and troubleshooting.
- Define retention periods for sensitive data that is stored for established business purposes.
- Verify that data retention policies are enforced and execute according to schedule.

When sensitive data is stored, it should be protected with encryption, and when it is displayed, subsets of the data should be masked in order to prevent inadvertent disclosure. As part of the self-assessment, you should verify the following requirements for encrypting and masking sensitive information:

- Verify that the payment account number (PAN) is partially masked unless there is a business need to view the full PAN.
- Verify that PANs are stored using one of the approved PCI DSS methods, which include one-way hashes and encryption with strong cryptography.
- Verify PANs are unreadable in application output, such as trace files, and exported copies of data, such as backup files.
- Verify cryptographic keys are securely stored. Acceptable methods include storing cryptographic keys on removable media protected with access controls. In cases where full-disk encryption is used, the encryption mechanism must be independent of the operating system (OS).
- Verify access to keys is restricted to the minimum number of people necessary, and that the keys are stored in the minimum number of locations practical.
- Verify key management procedures are in place to generate, store, and distribute keys.
- Verify procedures are in place to generate new keys to replace existing keys at the end of their useful life or in the case of suspected compromise.

Resource

For details on data storage requirements, see PCI Payment Application Data Security Standard: Requirements and Security Assessment Procedures v2, Requirements 3.

Network Security Practices

Networks supporting payment card applications must be secured using a number of security controls. The following checklist highlights major requirements:

- Verify that default configurations are not used on wireless access devices. This task includes changing encryption keys, Simple Network Management Protocol (SNMP) community strings, and passwords.
- Verify that firewalls are in place between wireless access devices and payment card applications. Network traffic between wireless access devices and the payment card application should be blocked unless the data exchange is required by the payment card application.
- Verify that payment card data is never stored in the DMZ.
- Limit remote access to payment card applications for application vendors only during times remote access is required by the vendor; for example, to update the application.

- If users have remote access to the payment card application, verify that secure controls are in place, such as access over a virtual private network (VPN), restricted access available only to known MAC addresses, and other approved methods.
- Verify that network traffic is encrypted using strong encryption, especially if data is sent over public networks.
- Verify that wireless access devices are updated to support strong encryption.
- Verify that non-console administrative access to payment card applications is encrypted.

It should be noted that many network security practices, such as encryption between clients and servers, utilize SSL certificates. The broad use of SSL certificates offers an example of a common occurrence in IT: A technology or service critical to the implementation of a higher-level function is often out of sight and, potentially, out of mind. It is important to maintain an awareness of the underlying technologies that enable security, such as SSL certificates, so that we plan for and manage their use properly.

Server and Application Security Practices

Payment card applications run on servers that must be secured. This security entails a combination of application configuration, OS configurations, and security services, such as antivirus and firewalls. As part of the process of ensuring payment card applications and their servers are secure, consider the following checklist items:

- Test payment card applications for vulnerabilities before deploying and while in production. Vulnerabilities that can compromise an application include susceptibilities to SQL injection attacks, buffer overflows, improper error handling, cross-site scripting attacks, and insecure cryptographic storage.
- Minimize the number of OS services and daemons running on payment card applications and other servers.
- Verify that OS services and daemons running on payment card and other application servers are securely configured.
- Establish procedures to identify new vulnerabilities and to acquire and install security patches as necessary. Critical security patches must be applied within 1 month of release.
- Ensure that default administrative accounts are not used by the payment card application.
- Ensure unique user accounts are assigned to all payment card application users. Shared accounts must not be used.
- Ensure that users are authenticated to the payment card application using something they know (for example, a password or passphrase), something they have (for example, a smartcard), or something they are (for example, a biometric measure such as a fingerprint).

- Verify passwords are changed every 90 days, ensure they conform to minimal requirements regarding length and use of alphanumerics, and verify that passwords are different from four previous passwords.
- Verify user accounts are locked after at most six failed login attempts, and ensure that accounts remain locked until unlocked by an administrator or for a period of at least 30 minutes.
- Verify anti-malware applications are properly configured on all servers and client devices that are used to access or support payment card applications.

Even with established procedures to protect data, secure networks and servers, and applications, it is important to monitor the function of these procedures.

Monitoring Practices

Monitoring a payment card application largely depends on the ability to collect information about significant events in application logs. The following checklist highlights steps you can take to ensure compliance with logging requirements:

- Verify payment card applications are enabled to log events.
- Ensure that details about individual access to cardholders' data are recorded in the application log.
- Ensure that activities performed by users with elevated privileges are recorded in the application log.
- Verify that the payment application logs events related to access to the audit trail.
- Verify that all use of the application's authentication and authorization systems is logged.
- Ensure that log entries include at least the following: user identification, the type of event that occurred, the date and time of the event, the point of origin of the activity, an indication of the data or system component affected by the activity, and an indication of whether the attempted activity succeeded.
- Use centralized logging to consolidate logs from multiple servers.

In addition to these checklist items, you should make sure that log files are secured and the risk of tampering is mitigated through the use of strong access controls and the transmission of log entries to a central server as soon as possible.

Policies and Procedures

Policies specify what a business does as a matter of practice, and in the case of security, what is required to meet business objectives regarding protecting data and IT resources. The PCI DSS has a number of requirements with regards to policies. The following checklist highlights important topics addressed:

- Verify that a policy exists that addresses the requirements listed in the PCI DSS documentation.
- Verify that a policy has been defined that documents the need for annual risk assessments to identify threats and vulnerabilities.
- Ensure the company's security policy is reviewed and revised as need at least once per year.
- Define operational security procedures in line with security policies.
- Define usage policies that address who is allowed to use payment card applications and what authentication steps are required.
- Ensure policies are in place that specify how devices are labeled and correlated to their owners.
- Ensure acceptable use policies are in place.
- Verify policies address issues such inactivity timeouts and remote access restrictions.
- Verify a policy exists that defines approved devices and approved locations for those devices.
- Ensure a policy exists that prohibits copying sensitive cardholder data to remote drives or removable media.
- Define policies that specify security personnel's roles and responsibilities. These should include details about who has responsibilities for authentication and authorization, incident response, and log review.
- Verify that a security awareness training policy is in place and complies with PCI DSS requirements.

The details of most of these policies are beyond the scope of this guide, but several are related to the use of SSL certificates and will be considered in more detail in a later section. The next section will delve deeper into checklist items specifically related to the use of SSL certificates.

Securing Servers and Data Transmission with SSL

The PCI DSS standards documentation mentions the use of strong encryption in several places. What does this mean from an operational perspective? In part it means:

- Using encryption algorithms and key lengths that are considered strong by the cryptographic community
- Using strong protocols
- Configuring SSL software to avoid vulnerabilities and weaknesses in protocols

Let's look at each of these in more detail.

Use Strong Encryption

Strong encryption is a somewhat imprecise term but it is generally used to describe an algorithm that is highly resistant to deciphering without an encryption key. In the simplest sense, an encryption algorithm is strong if the number of possible keys is so large there is no practical way to use a brute-force search of all possible keys to find the deciphering key. Strong cryptography is not as vulnerable to analytic techniques either.

Cryptanalysis can be done in many ways. For example, an early form of cryptanalysis looked at the frequency with which characters appeared in the encrypted text in order to correlate those with the frequency of characters one would find in plain text. Using a more advanced technique known as differential cryptanalysis, someone could apply an encryption algorithm to a large number of input texts and study the resulting output in an effort to detect how differences in input affect the resulting output. Strong encryption algorithms are considered less likely than weak encryption algorithms to be compromised by such techniques.

> **Note**
>
> The strength of an encryption algorithm is a product of the key length that is used. When considering the strength of an encryption algorithm, be sure to include the length of the encryption key.

The PCI DSS Glossary includes the following examples of strong cryptography:

- AES with at least 128-bit keys
- Triple DES (TDES) with minimum double key length
- RSA with 1024-bit keys
- ECC with 160-bit keys
- ElGamal with 1024-bit keys

Advances in cryptanalysis may one day eliminate one or more of these algorithm-key length pairs from the list of accepted strong cryptography. The Digital Encryption Standard (DES), for example, was once a standard encryption algorithm in widespread use starting in the mid-1970s. However, by 1999, a group of cryptography researchers demonstrated DES encryption could be broken in less than 24 hours using a brute-force search on a custom-built device dubbed "Deep Cracker." It is also possible that cryptanalysis will discover weaknesses in any of these algorithms that may be compensated for with the use of longer keys. In such cases, the algorithms may still be used in applications requiring strong cryptography as long as keys are sufficiently long.

Using strong encryption requires the use of sound key management practices. Key management is a complex process that includes:

- Distinguishing different types of keys according to their use, such as public/private keys, symmetric authentication keys, symmetric key wrapping keys, and so on
- Defining valid time periods for cryptographic keys, also known as cryptoperiods
- Considering private key possession
- Addressing compromised keys

Resource

For guidance on key management, see the National Institute of Standards and Technology's (NIST) Recommendations for Key Management – Part 1: General (Publication 800-57).

Use Strong Protocols

Protocols, in this discussion, are agreed upon procedures for exchanging data. Many of the widely used protocols are defined by the Internet Engineering Task Force (IETF—see the list of Official Internet Protocol Standards examples). Some of the protocols are designed with less concern for security than others. In some cases, this is not an issue because the protocols are low-level network protocols designed for minimum overhead; the User Datagram Protocol (UDP) is a case in point. In other cases, less secure protocols are coupled with more secure versions, such as the insecure FTP and the more secure versions FTPS or the alternative protocol SSH File Transfer Protocol. When assessing network communications for use with payment card applications, strong protocols are preferable to weak protocols.

Characteristics of Weak Protocols

Weak protocols do not employ security measures found in strong protocols. Strong measures include:

- Encrypting contents
- Enabling authentication
- Providing evidence for message integrity

The problem with weak protocols can be seen with basic HTML form authentication in which passwords are passed in plain text. An application user navigates to the authentication page of an application and enters a username and password. The username and password may be passed to the application server in clear text or encoded using a simple scheme such as base64 encoding. There is no encryption, so the username and password are available for capture by someone sniffing network traffic. There is no way to authenticate where the username and password originated, so the username and password may be used by someone other than the legitimate user. In addition, as there is no way to verify the integrity of the data that is passed, the data may be changed by someone using a man-in-the-middle attack.

Another example of a weak protocol is authenticating based on the contents of an HTTP referrer, which is metadata about an HTTP request that identifies the requestor. Like the example provided earlier, there is no encryption, reliable authentication, or ability to detect tampering with the contents of the HTTP referrer data.

Strong protocols should be used when transmitting payment card data.

Characteristics of Strong Protocols

Not surprisingly, strong protocols avoid the vulnerabilities of weak protocols using cryptographic techniques, including:

- Strong encryption algorithms and key lengths
- Support for authentication, typically using SSL certificates
- Support for message integrity using message digests

Often when SSL certificates are used for authentication, it is only servers that are authenticated. This setup is understandable when the objective is to assure users that they are communicating with legitimate servers belonging to the business the users are attempting to work with. In the case of payment card data processing, it is important to authenticate any device that receives payment card data. Even when you use strong protocols, you should keep in mind that some configurations, early versions of the protocols, or implementations may harbor weaknesses.

Beware of Weaknesses in Protocols

One of the targets of cryptanalysis is weaknesses in cryptographic algorithms, but those algorithms are only part of the process of establishing secure communications. There are other steps, such as starting a session, deciding which algorithms to use for encryption and message digests, acknowledging responses to requests, and so on. These steps are all defined in protocols, such as the SSL/TLS protocols, and the sequences of steps defined in the protocols may be vulnerable to attack.

An example of a protocol weakness is the SSL renegotiation vulnerability, which creates a vulnerability that could be exploited for a man-in-the-middle attack. Renegotiation is the process of carrying out the SSL handshake over an established SSL connection. Prior to 2010 when the vulnerability became widely known, commonly used SSL implementations allowed for unsecure renegotiations by default. It is now considered a safe practice to use secure renegotiation by default. SSL protocols should be configured by default to use secure renegotiation.

> **Resource**
>
> See "Transport Layer Security (TLS) Renegotiation Indication Extension" for more details on secure renegotiation.

More recently, security researchers have demonstrated how to exploit a vulnerability in the TLS 1.0 protocol. The attack, known as Browser Exploit Against SSL/TLS (BEAST) exploits a weakness in the way a cipher is implemented. Version 1.1 of TLS introduced more randomization into the way CBCs are used and is not vulnerable to the BEAST attack.

The key point to remember here is that even strong encryption algorithms can be used in vulnerable protocols. In order to protect sensitive and confidential information, it is important to be aware of vulnerabilities in protocols and in particular implementations of those protocols. The BEAST attack took advantage of a weakness in the TLS protocol that could be exploited in any implementation of the TLS 1.0 protocol. In addition to vulnerabilities in protocols, which affect all implementations, there may be vulnerabilities introduced by the way a protocol is implemented. For example, there may be a bug in a program that implements part of the SSL/TLS protocol in a browser. This requires a patch to the browser but does not warrant changing the protocol. From an operations perspective, it is important to be aware of both protocol and implementation vulnerabilities.

> **Tracking Vulnerabilities**
>
> A number of resources on the Web are available to help you track vulnerabilities, including the National Vulnerability Database and the Common Vulnerabilities and Exposures web site.

Essential Security Policies Related to SSL Certificates

Security policies define rules that must be followed with regards to implementing information security. The range of security policies includes acceptable use, physical security, wireless security and desktop security. Policies that influence the use of SSL certificates, and PCI DSS compliance, include:

- Encryption policy
- Access control policy
- Data classification policy
- Monitoring and auditing policy

Policy documents typically include brief definitions of the scope, intended audience, and purpose of the policy. The core of any policy document is the policy statement that defines the rules to be followed with regard to the subject of the policy.

Encryption Policy

Encryption policies should define several aspects of encryption use within a business setting, including:

- Use of encryption for data at rest
- Use of encryption for transmitted data
- Types of encryption algorithms allowed and key length
- Key management

Within each of these sections, the policy should define high-level requirements, perhaps referencing government or industry standards. Policy documents tend to be brief because they only specify what needs to be done, they do not describe how to implement these requirements or elaborate on why a particular choice was made about each rule. However, policy writers sometimes include a brief description of the motivation for the need for a particular set of rules.

Using Encryption for Data at Rest

Data at rest is data that is stored on fixed or mobile storage devices. The data at rest section of the policy should specify under what conditions data needs to be encrypted and can include:

- Encrypting by data classification (see later discussion for more on data classification schemes)
- Encrypting data by application type, such as payment card processing applications
- Encrypting data by business function; for example, human resources and legal departments may encrypt all data by default

The policy may specify if full-disk encryption is required and if so how authentication should be implemented. This specification is particularly important with respect to the PCI DSS because it explicitly states that the authentication system for full-disk encryption should be separate from the OS's authentication mechanism.

The policy may also include rules for the use of encryption on mobile devices, particularly how access controls are implemented on mobile storage devices. For example, a policy may specify that a passphrase must meet several criteria (for example, length, combination of character types, and so on) for acceptable use. It may also include rules about how long confidential data can remain on a mobile device, who may physically possess mobile media with confidential information, limits on where the mobile device may be taken, etc.

Using Encryption for Transmitted Data

Encrypting data during transmission is clearly an important aspect of the PCI DSS regulations. Policies on this area should meet the minimal standards specified in the PCI DSS. In addition to stating the need for strong encryption anytime payment card data is transmitted, the policy may explicitly state the need for additional measures to ensure the standards are met. For example, the policy may specify that SSL software be configured to negotiate the use of only strong cipher suites. This is implied in the PCI DSS requirement and boarders on an implementation issue, but it is an important point and explicitly stating it in the policy can help educate the audience about a potential weakness in an SSL configuration.

Types of Algorithms and Key Lengths

The PCI DSS documentation provides examples of strong cryptography algorithms and key lengths. They can be included in the encryption policy. The list may change over time, so part of the regular reviews of security policies should include a review of the list of algorithms and key lengths considered acceptable by the PCI Security Standards Council.

Key Management

The key management section of the encryption policy should include rules about who is allowed to generate new keys, separation of duty and job rotation requirements for personnel with key management responsibilities, and the need for separate keys for separate functions. Key management is such a complex task that this subject is a good candidate for referencing external guidelines, such as the NIST guidelines on key management.

Access Control Policy

Businesses need to control who has access to their data and applications, and access control policies specify the rules for this. In practice, you may find the need for multiple access control policies, such as remote access control, application access control, and physical access control policies. In general, these policies define the actions that people in particular roles can perform on the network, with data, or with regard to physical access to IT infrastructure.

Some of the topics that may be included are:

- Description of business roles
- Specification of who is allowed to make decisions about granting and revoking access
- Minimal requirements for access controls, such as logical access controls for data or the use of magnetic cards for physical access to data centers
- Logging requirements for access events

As with other policies, there may be a discussion of separation of duties and job rotation requirements.

Data Classification Policy

The PCI DSS regulations make reference to payment card data and cardholder data when specifying the types of controls that must in place to protect this information. This kind of data warrants a particular set of controls. Even within the PCI DSS standard, authentication data, such as card not present verification data, is subject to even stricter limits on storage than other data, such as the payment account number. These rules highlight an implicit data classification scheme within the PCI DSS regulations, but the data classification is generally applicable to all business data.

A data classification policy defines the different types of data and the types of security controls that should be in place to protect the different types. A basic data classification scheme might include four categories:

- Private
- Confidential
- Sensitive
- Public

Private data may be data about customers, employees, or others. The business maintains this data as part of normal business operations, but it is not considered to be data about the business. Private data can include name, address, health records, credit scores, etc. The data classification policy may specify that private data must not be released to unauthorized users; it also defines rules for determining what constitutes an authorized user.

Confidential data is similar to private data in that it is not to be disclosed unless necessary. Unlike private data, confidential data is about business operations and includes topics such as strategic plans, non-public financial statements, legal opinions, etc. The data classification should include rules for protecting this type of data, such as the need for using encryption to transmit data and only transmitting data between devices that are mutually authenticated using SSL certificates.

Sensitive data is business data that should not be disclosed but if it were it would not cause substantial harm to the business. This category might include mid-level manager's schedules and HR policy documents. The data classification policy might indicate that such documents do not need to be encrypted when at rest but should be encrypted when transmitted over public networks.

Public information is any information that can be disclosed outside the company without harm. Press releases, product catalogs, and marketing material are examples of public data. The data classification policy may not require any security controls on this data.

Monitoring and Auditing Policy

Policies governing monitoring and auditing describe the scope of logging activity required, including:

- Types of applications that should generate logs
- Types of information captured in the logs
- Retention periods for logs
- Requirements for centralized log management
- Reporting of events captured in logs

The PCI DSS documentation includes requirements for logging activity that should be addressed in the monitoring and auditing policy.

Another aspect of PCI DSS compliance is ensuring that security controls built on SSL certificates are maintained by procedures, including SSL certificate life cycle management.

SSL Certificate Life Cycle Management

SSL certificates are assets that must be managed like any other. The basic operations in SSL certificate life cycle management are:

- Choosing the right SSL certificate
- Renewing SSL certificates
- Tracking SSL certificate inventory

The first step in the life cycle management process is selecting the right types of SSL certificates for your requirements.

Choosing the Right SSL Certificate

SSL certificate vendors offer a number of types of SSL certificates for different needs. Some SSL certificates are designed for a single server or other device, some are designed to be used with multiple devices on the same domain, and still others can be used on multiple servers with different names. (The latter are known as Subject Alternate Name—SAN—SSL certificates). Determining which option is best for you will depend on a few factors:

- Is the number of servers that requires SSL certificates small enough that you can manage separate certificates for all of them?
- Do you frequently start new virtual servers that are part of one domain and would benefit from a domain-level certificate?
- Do you have multiple servers that have different domain names but are managed as a single logical unit?

Your answers to these questions can help guide the selection of the type of SSL certificate you should use.

In addition to the different ways SSL certificates can be associated with servers, there are two levels of verification and assurance that SSL certificate vendors provide. A conventional SSL certificate is granted to a business or other entity that demonstrates it is actually that entity according to the criteria established by the vendor. These criteria could be as simple as having an email address at the domain specified in the SSL certificate request or it could be much more rigorous. The SSL certificate industry established standards for an alternative verification regime and created the Extended Validation (EV) SSL certificate. The industry established common standards for verifying the identity of an EV SSL certificate requestor and ensuring it is a legitimate business. EV SSL certificates are designed to give customers, business partners, and others additional assurance that they are dealing with a legitimate and authenticated business.

Renewing SSL Certificates

SSL certificates are valid for a predefined period of time and at the end of that period, the SSL certificate must be renewed. Tracking SSL expiration dates can be done either on a server-by-server basis using OS tools, such as Microsoft Windows Management Console (Figure 4.1) or with SSL certificate management consoles provided by SSL certificate vendors (Figure 4.2).

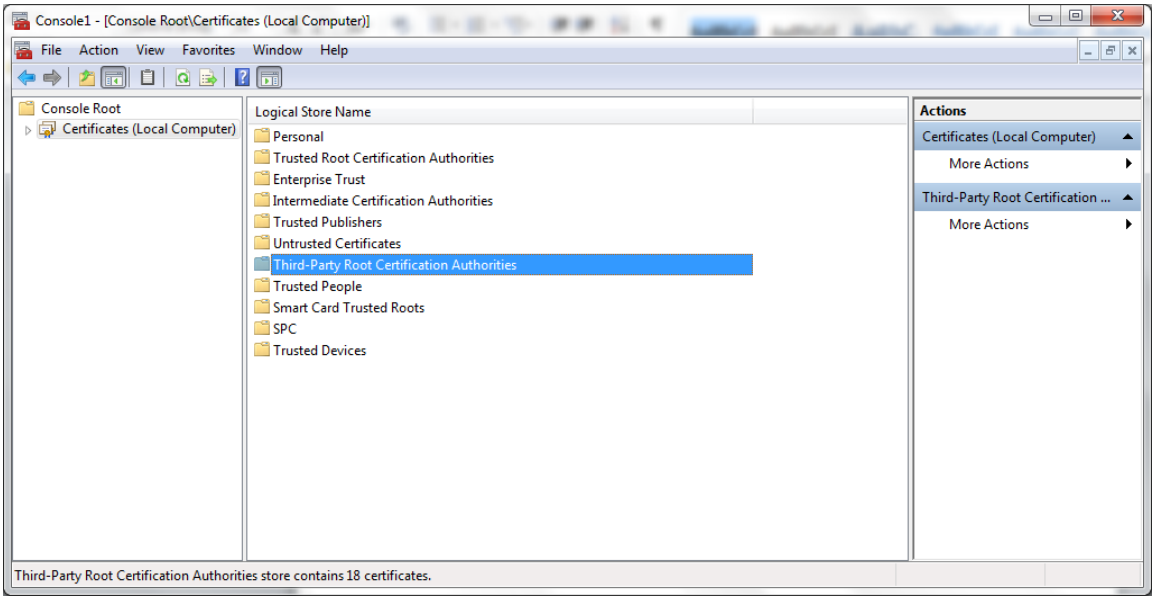

Figure 4.1: Microsoft Windows OS provides a certificate management snap-in for the Microsoft Management Console (MMC).

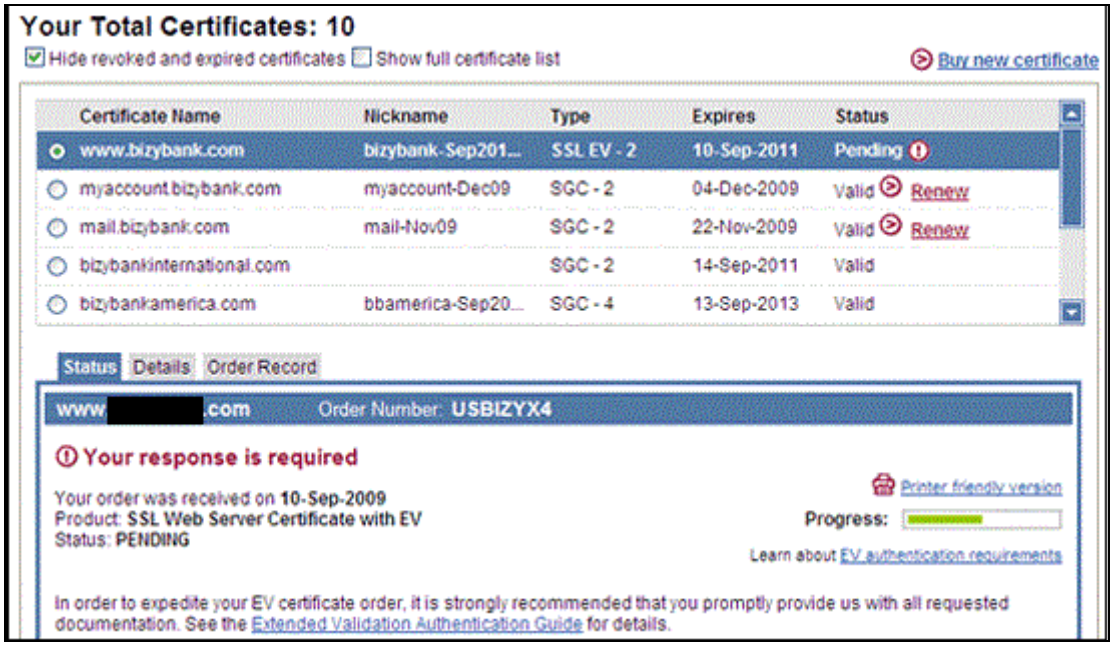

Figure 4.2: Example console for managing multiple SSL certificates across an enterprise.

Tracking SSL Certificate Inventory

Another general task in the SSL certificate life cycle management process is tracking inventory of SSL certificates. Enterprise management consoles are helpful for this task, especially in businesses with Windows, Linux, Unix, and other OSs each of which have its own methods and procedures for managing SSL certificates. By tracking SSL certificates, you might be able to better understand your needs. This understanding can lead to more efficient use of domain certificates, SAN certificates, and other options for reducing management overhead and reducing the cost of purchasing large numbers of SSL certificates.

Summary

The PCI DSS defines a range of security requirements for protecting payment card and cardholder data. SSL certificates play important roles in a number of PCI DSS requirements, including authentication and encryption services. This chapter has provided a high-level checklist for assessing compliance with key requirements. It also provided a more detailed look at policies and SSL management practices that support your efforts to meet PCI DSS requirements.