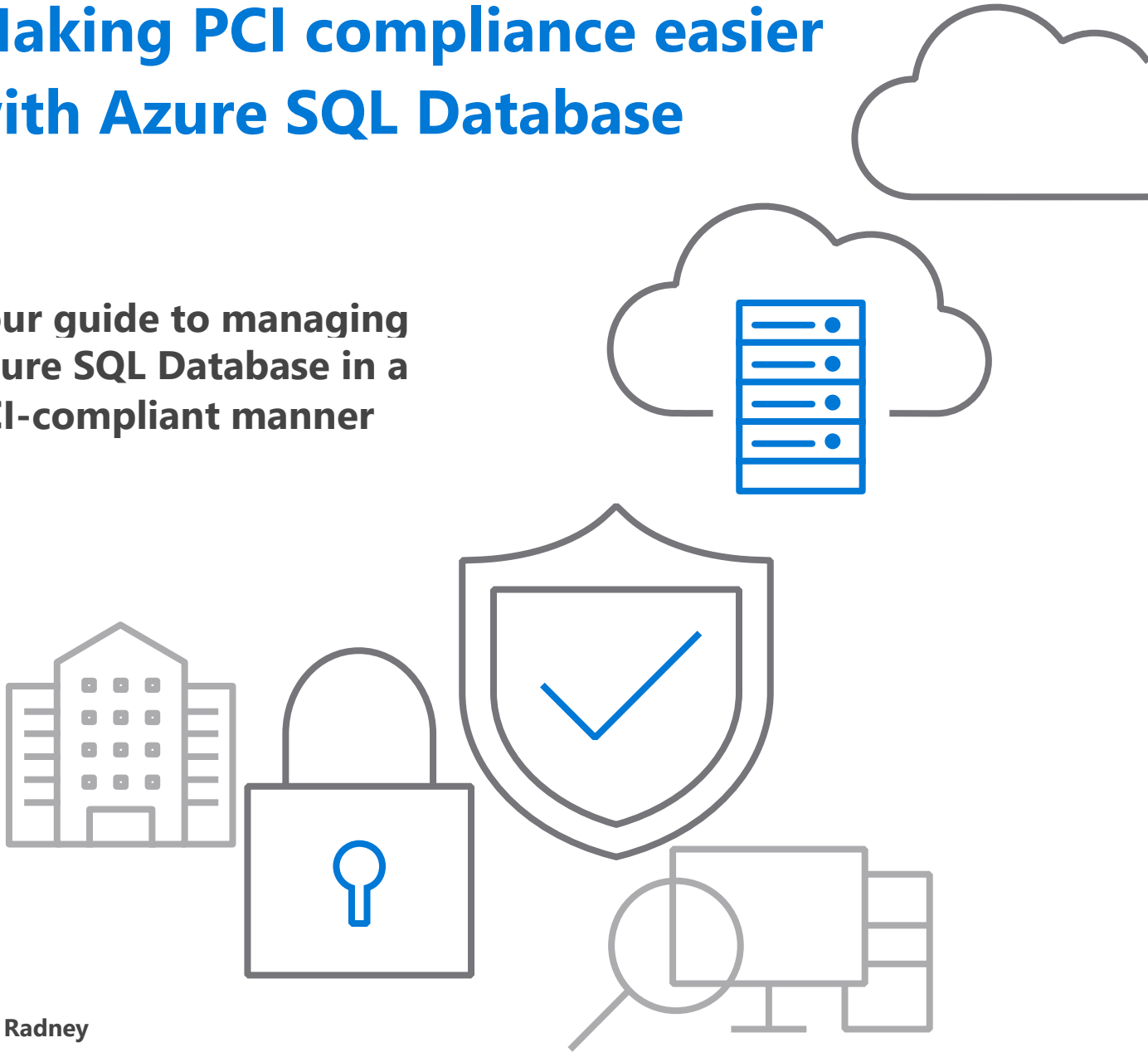
Microsoft

Making PCI compliance easier with Azure SQL Database

Your guide to managing Azure SQL Database in a PCI-compliant manner



Tim Radney
SQLskills

Contents

Introduction 1
Configuring Azure SQL Database and Azure SQL Managed Instance for PCI DSS Compliance 6
Requirement 1 – Implement Firewalls 6
Requirement 2 – Don't Use Vendor-Supplied Defaults for System Passwords and more 11
Requirement 3 – Protect Stored Cardholder Data 14
Requirement 4 – Encrypt Transmission of Cardholder Data 19
Requirement 5 – Protect All Systems Against Malware 20
Requirement 6 – Develop and Maintain Secure Systems and Applications 21
Requirement 7 – Install and Maintain a Firewall Configuration to Protect Cardholder Data 26
Requirement 8 – Identify and Authenticate Access to System Components 28
Requirement 9 – Restrict Physical Access to Cardholder Data 32
Requirement 10 – Track and Monitor All Access to Network Resources and Cardholder Data 36
Requirement 11 – Regularly Test Security Systems and Processes 38
Requirement 12 – Maintain a Policy that Addresses Information Security for all Personnel 41
Conclusion 44
Appendix: PCI DSS Compliance Guidelines 46
Requirement 1 46
Requirement 2 48
Requirement 3 49
Requirement 4 52
Requirement 5 53
Requirement 6 54
Requirement 7 59
Requirement 8 60
Requirement 9 63
Requirement 10 66
Requirement 11 69
Requirement 12 71

Introduction

The Payment Card Industry Data Security Standard (PCI DSS or PCI Standard), defined and controlled by https://www.pcisecuritystandards.org/, is a standard designed to prevent fraud using credit card information through increased controls around credit card data. Obtaining PCI DSS compliance is a requirement for all organizations that:

- Accept credit card payments
- Process credit card transactions
- Transmit or store credit card data

For organizations that have their own data centers, it can be a time consuming and costly process to become PCI compliant and remain so. Those companies must stay current with patching operating systems and applications as well as applying any security updates that may be released by software vendors. They must collect and review daily log files, perform periodic vulnerability test, and may also require penetration tests.

Many organizations that require PCI compliance are not able to take on the responsibility of building a PCI-compliant environment and handle the daily responsibilities that come with being PCI compliant. In that case, they must look elsewhere for solutions. Migrating to or building those environments in Microsoft Azure can drastically reduce the PCI-compliance responsibility that an organization must bear.

The intent of this paper is to showcase how much easier PCI DSS compliance is to achieve when customers take advantage of Azure SQL database and Azure SQL managed instance. Microsoft has invested heavily in making sure that the Platform-as-a-Service (PaaS) environment meets the PCI Standard. This paper will highlight specifics that Microsoft fully owns, those that are a shared responsibility where Microsoft and the customer collectively have requirements to meet, and those that are the customers responsibility.

Throughout the total Microsoft PaaS control coverage, Microsoft is responsible for 61.45% of the PCI DSS compliance requirements, which is a huge burden lifted off customers that must be PCI-compliant. The requirements that are the customer's responsibility are mostly related to customer or organization having to document various processes or standards relating to protecting card holder data. Many of those scenarios if the customer has deployed their solution in PaaS, is simply documenting how Microsoft controls that standard. As you read through this document, you'll see just how much of the burden of meeting PCI DSS compliance has been removed from the customer and absorb by Microsoft.

PCI DSS compliance is made up of 6 primary goals that represent common-sense steps that mirror security best practices. The 6 primary goals are refined by 12 specific requirements which

give organizations further guidance on how to achieve each goal. These requirements are as follow:

1. Build and maintain a secure network and systems
   1.1. Install and maintain a firewall configuration to protect cardholder data
   1.2. Do not use vendor-supplied defaults for system passwords and other security parameter
2. Protect cardholder data
   2.1. Protect stored cardholder data
   2.2. Encrypt transmission of cardholder data across open, public networks
3. Maintain a vulnerability management program
   3.1. Protect all systems against malware and regularly update antivirus software or programs
   3.2. Develop and maintain secure systems and applications
4. Implement strong access control measures
   4.1. Restrict access to cardholder data by business need to know
   4.2. Identify and authenticate access to system components
   4.3. Restrict physical access to cardholder data
5. Regularly monitor and test networks
   5.1. Track and monitor all access to network resources and cardholder data
   5.2. Regularly test security systems and processes
6. Maintain an information security policy
   6.1. Maintain a policy that addresses information security for all personnel

To further explain the objectives of each requirement and to make them testable for compliance assessors, they are broken down further into multiple objectives, and these are listed and addressed in the rest of this whitepaper.

Additionally, there are 4 merchant levels of PCI compliance that are based upon the number of payment transactions that are performed. These levels are used to determine the depth of assessment and security validation that is required for the merchant to meet and pass the PCI DSS compliance assessment.

- PCI DSS Compliance Level 1: Merchants with over 6 million transactions a year, across all channels or any merchant that has previously had a data breach
- PCI DSS Compliance Level 2: Merchants with between 1 million and 6 million transactions a year, across all channels
- PCI DSS Compliance Level 3: Merchants with between 20,000 and 1 million online transactions a year
- PCI DSS Compliance Level 4: Merchants with less than 20,000 online transactions a year, or any merchant processing up to 1 million regular transactions a year

Compliance for Level 1 merchants is the most demanding and requires an independent annual Report on Compliance (ROC) by a certified Qualified Security Assessor (QSA). This is also known

as a Level 1 onsite assessment. The passing of the audit and receiving the AoC from the assessor starts the effective period which is good for one year from the date the AoC is signed.

Levels 2-4 complete an annual Self-Assessment Questionnaire (SAQ), a quarterly network scan by an Approved Scanning Vendor (ASV), and an Attestation of Compliance form. Level 4 merchants may not be subject to all those requirements.

For merchants that maintain their cardholder data in-house, it's their responsibility to maintain PCI DSS compliance for their level and pass their assessment. Depending on their level, this can be costly and require multiple full-time staff as well as expensive software solutions for monitoring, scanning, and processing logs.

For merchants who are looking at Microsoft Azure for solutions, many aspects of the PCI DSS compliance have already been met. Microsoft Azure is certified as compliant under PCI DSS compliance 3.2.1 as a Level 1 service provider. The auditor successfully reviewed the Azure environment including validating operations, management, infrastructure, development, support, and any in-scope services. For more details please look at the Service Trust Portal, which is a free resource but will require you to login to view, here: https://servicetrust.microsoft.com/ViewPage/MSComplianceGuideV3?command=Download&downloadType=Document&downloadId=425af30f-1236-41bc-b45c-98a52ee84c28&tab=7027ead0-3d6b-11e9-b9e1-290b1eb4cdeb&docTab=7027ead0-3d6b-11e9-b9e1-290b1eb4cdeb_PCI_DSS.

Microsoft provides an Azure Security and Compliance PCI DSS Blueprint that enables customers to build and launch cloud solutions that can handle sensitive payment and cardholder data, including the card number, expiration date, and verification data. The blueprint provides guidance and common reference architectures designed to help you protect your applications and data, provide cost-effective security for organizations of all sizes, and support your compliance efforts. You can find the blueprints at https://servicetrust.microsoft.com/ViewPage/PCIBlueprint.

Although Microsoft Azure is PCI DSS 3.2.1 compliant, any merchant, processor, acquirer, issuer, service provider, or anyone storing cardholder data, still has a customer responsibility for PCI DSS compliance. Understanding the shared responsibility for implementing security controls in a cloud environment is critical for any customer building systems on Azure as implementing a PCI security control may be Microsoft's responsibility, the customer's responsibility, or a shared responsibility. Different cloud services may affect how responsibilities are shared. For example, implementing a data solution that utilizes both Azure SQL Database and Microsoft SQL Server on an Azure VM would have a different set of guidelines than building a data solution on just Azure SQL Database.

Microsoft's PaaS data offerings for SQL Server give customers a choice between Azure SQL Database and Azure SQL Database Managed Instance. Although these are both in the PaaS

environment, they are quite different by design. Azure SQL Database is built on a programming model that is database scoped and offers database isolation. Azure SQL Database Managed Instance is built on an instance-scoped programming model making it much more compatible with the full Microsoft SQL Server product.

For instance, as Azure SQL Database is database scoped, it does not support cross-database queries. This is a critical benefit for customers that need database isolation, such as Software-as-a-Service (SaaS) companies that have a database per customer. They can easily create a database when a new customer signs up for their service, grant access to the customer, and not have to worry about the customer being able to elevate privileges or see data from any other customer with a database on the system.

Another benefit of Azure SQL Database is that its surface area configuration is much smaller than for Microsoft SQL Server and Azure SQL Database Managed Instance. Azure SQL Database does not have SQL Server Agent built in or support Database Mail, Common Language Runtime (CLR), linked servers, and other features. This restrictive footprint makes some aspects of the PCI DSS compliance easier to attain. With Azure SQL Database Managed Instance being instance-scoped and a goal of 100% surface area compatibility with Microsoft SQL Server, there are more components to secure to attain compliance, as Azure SQL Database Managed Instance does support cross database queries and all the features mentioned above that Azure SQL Database does not support.

The Azure PaaS platform also offers many features and tools that can be leveraged to better support and protect the data services offerings, and many of those are focused on security. Examples of these features include Vulnerability Assessments, Data Classification and Discovery (in preview at time of writing), auditing, Azure Active Directory (AAD), and Threat Protection. For data protection, there are many features including Always Encrypted for encrypting column level data within the database itself, Transparent Data Encryption (TDE) for encrypting the data and log file at rest, Dynamic Data Masking (DDM) for masking how sensitive data appears in query results, data redundancy, and managed encrypted backups.

You can securely and reliably deploy resources within Azure using the Azure Portal, an API, PowerShell, or the ARM. Azure networking resources include virtual networks, virtual private networks, network security groups, public and private endpoints, firewalls, ExpressRoute, VPN Gateway, and more. The Azure Application Gateway allows for web traffic load balancing that enables you to manage traffic to your web applications. For encryption key management there is the Azure Key Vault.

All these tools and features help both customers and Microsoft meet the requirements for PCI DSS compliance when using Microsoft Azure.

When reading further, please note that the PCI DSS Requirement controls have a unique naming convention. Some requirements start with whole numbers, others start with decimals. Some controls have many subparts. Please reference the appendix for control number definitions.

Configuring Azure SQL Database and Azure SQL Managed Instance for PCI DSS Compliance

This section of the whitepaper explains how Microsoft Azure helps with the various PCI DSS requirements (the actual requirements are listed in the Appendix). An introduction to the overall requirement is given, then some notes on who owns each control and what is required, and finally a summary of the overall requirement.

Requirement 1 – Implement Firewalls

Requirement 1 states that customers must install and maintain a firewall configuration to protect cardholder data. This requirement is primarily the customer's responsibility since they are responsible for configuring network controls between on-premises and Azure resources. The firewall controls are within the customer's responsibility for protecting and limiting access to cardholder data.

Microsoft Azure provides several resources to help customers with Requirement 1 and its individual controls and can provide 85.7% reference architecture requirement coverage. The Azure Resource Manager (ARM) provides a consistent management layer that enables customers to create, update, and delete resources in their Azure subscription. With the ARM, you can use its access control, auditing, and tagging features to secure and organize your resources after a deployment. Azure Networking provides virtual private network (VPN) gateway, ExpressRoute, application gateways, network security groups, Azure App Service Environment (ASE), AAD, and more.

A VPN gateway is a specific type of virtual network gateway that customers can use to send encrypted traffic between an Azure virtual network and an on-premises location over the public internet or to send encrypted traffic between Azure virtual networks over the Microsoft network. Each virtual network is limited to one VPN gateway, however multiple connections can be created to the same VPN gateway. Customers who require private connections to Azure can utilize an ExpressRoute facilitated by a connectivity provider. Customers can setup connectivity from an any-to-any (IP VPN) network, a point-to-point Ethernet network, or a virtual cross-connection through a connectivity provider at a co-location facility. ExpressRoute connections do not go over the public internet allowing more reliability, faster speeds, consistent latency, and higher security than typical connections over the Internet. ExpressRoute and VPN gateways help customers met the requirements of isolating and protecting data on the network.

An Azure Application Gateway is a web traffic load balancer that enables customers to manage traffic of their web applications. With an Application Gateway, customers can make routing decisions based on additional attributes of an HTTP request, such as URI path or host headers in addition to how traditional load balancers operate at the transport layer (OSI layer 4 – TCP and UDP) and route traffic based on source IP address and port, to a destination IP address and port. Azure Application Gateways support the following features:

- Autoscaling
- Secure Sockets Layer (SSL/TLS) termination
- Zone redundancy
- Static VIP
- Web application firewall
- URL-based routing
- Multiple-site hosting
- Redirection
- Session affinity
- Websocket and HTTP/2 traffic
- Azure Kubernetes Service (AKS) Ingress controller preview
- Connection draining
- Custom error pages
- Rewrite HTTP headers
- Sizing

A Network Security Group (NSG) contains security rules that allow or deny inbound network traffic to, or outbound network traffic from, different types of Azure resources. An NSG can contain zero, or as many rules as desired within the Azure subscription limits. By default, Azure creates rules in each NSG that you create. The default rules consist of AllowVNetInBound, AllowAzureLoadBalancerInBound, DenyAllinbound, AllowVNetOutBound, AllowInternetOutBound, and DenyAllOutbound. Each rule applies to all source and destination ports and all protocols.

The ASE is a feature that provides a fully isolated and dedicated environment for securely running App Service apps as high scale. ASEs are appropriate for application workloads that require high memory utilization, very high scale, and isolation and secure network access. ASEs are isolated to only running a single customer's applications and are always deployed into a virtual network. Customers have control over inbound and outbound application network traffic to establish high-speed secure connections over VPNs to on-premises corporate networks.

For identity management, customers can leverage AAD natively or by synchronizing on-premises Active Directory to AAD. AAD is Microsoft's cloud-based identity and access management service

This requirement is further broken down into the 22 controls listed below.

1.1.1 This is a customer-owned control.

The customer is responsible for establishing a formal process that tests and approves all changes made to firewall and router configurations. The ARM can assist customers with this requirement by leveraging the APIs to create, update, and delete resources within their Azure

subscription as well as control access, audit, and tag features to secure and organize resources after a deployment.

1.1.2 & 1.1.3 These are customer-owned controls.

The customer is responsible for documenting all connections between the cardholder data environment and other networks and all cardholder data flows across systems and networks. This documentation should include a network diagram. Customers and leverage the ARM capabilities by reviewing audit logs and tags from properly tagged features and resources and use those in the documentation process. Customers can also use the reference architecture network diagram and threat model found at https://servicetrust.microsoft.com/ViewPage/PCIBlueprint.

1.1.4 This is a shared control.

Microsoft Azure employs boundary protection devices such as gateways, network ACLs and application firewalls to control communications at external and internal boundaries at the platform level. The customer then configures these to their specifications and requirements and Microsoft Azure filters communication coming into the platform.

The customer is responsible for establishing firewall requirements at each internet connection and between any demilitarized zone (DMZ) and internal networks. Customers can use the ASE which provides resources complete tenant isolation and supports address restrictions and manage service identities. Customers can use Azure Networking which supports NSGs, an Application Gateway that enables customers to manage traffic, an ExpressRoute which is a private connection to Azure and removes the need for a DMZ, and a VPN Gateway to encrypt traffic.

1.1.5 This is a customer-owned control.

The customer is responsible for establishing descriptions of groups, roles, and responsibilities for the management of network components. AAD can be used to enforce role-based access control (RBAC) for all Azure services and network components can be managed through the systems engineering role.

1.1.6 This is a customer-owned control.

The customer is responsible for documenting the business justification for all services, protocols, and ports allowed. This documentation should include what security features are implemented for any protocols considered to be insecure.

1.1.7 This is a customer-owned control.

The customer is responsible for reviewing firewall and router rule sets at least every six months.

1.2 This is a customer-owned control.

The customer is responsible for ensuring firewall and router configurations restrict untrusted networks and system components in the cardholder data environment. See customer options in 1.1.4.

1.2.1 This is a customer-owned control.

The customer is responsible for restricting inbound and outbound traffic to that which is necessary for the cardholder data environment. All other traffic must be denied. See customer options in 1.1.4.

1.2.2 This is a customer-owned control.

The customer is responsible for securing router configuration files and ensuring the most current configuration is used at start-up. See customer options in 1.1.4.

1.2.3 Microsoft Azure covers this control.

This control is not applicable to the Microsoft Azure environment because wireless access is not permitted within the Microsoft Azure network environment.

1.3 This is a customer-owned control.

The customer is responsible for ensuring no direct access exists between the public Internet and system components. See customer options in 1.1.4.

1.3.1 This is a customer-owned control.

The customer is responsible for implementing a DMZ that limits inbound traffic. The DMZ must be limited to system components that provide authorized, publicly accessible services, protocols, and ports. See customer options in 1.1.4.

1.3.2 This is a customer-owned control.

The customer is responsible for ensuring that all inbound Internet traffic is limited to IP addresses within the DMZ. See customer options in 1.1.4.

1.3.3 This is a shared control.
Microsoft Azure implements network filtering to prevent spoofed traffic and restrict incoming and outgoing traffic to trusted platform components.

The customer is responsible for implementing anti-spoofing measures. See customer options in 1.1.4.

1.3.4 This is a customer-owned control.

The customer is responsible for restricting unauthorized outbound traffic from the cardholder data environment. The ASE, Azure Networking, Application Gateway, ExpressRoute, and VPN Gateway can be used.

1.3.5 This is a shared control.

Microsoft Azure implements network filtering to prevent spoofed traffic and restrict incoming and outgoing traffic to trusted platform components. The Microsoft Azure network is segregated to separate customer traffic from management traffic.

The customer is responsible for ensuring that only established connections into the cardholder data environment are permitted. See customer options in 1.1.4.

1.3.6 This is a shared control.

Microsoft Azure uses SQL DB firewall settings to whitelist ASE web application connections only.

The customer is responsible for segregating cardholder data from the DMZ and/or untrusted networks. Azure Networking and Azure SQL Database firewall settings can be used.

1.3.7 This is a shared control.

Microsoft Azure uses NAT and network segregation to separate customer traffic from management traffic. Azure devices are uniquely identified by their UUID and are authenticated using Kerberos. Azure managed network devices are identified by RFC 1918 IP addresses.

The customer is responsible for ensuring private IP address and routing information is not disclosed to unauthorized parties.

1.4 Microsoft Azure covers this control.

This control is not applicable to the Microsoft Azure environment because Microsoft Azure does not issue or allow connectivity via mobile devices to Microsoft Azure.

1.5 This is a customer-owned control.

The customer is responsible for documenting security policies and operational procedures of firewall management. These policies and procedures should be distributed to all associated parties.

Summary: This requirement requires the customer to implement numerous controls and policies that are largely covered by Azure services and features. When it comes to deploying Azure SQL Database and Azure SQL Managed Instance in Microsoft Azure, the use of firewalls,

network security groups, AAD, subnets, VNETs, and endpoints, give you the security and control you need to help meet PCI DSS compliance requirement 1 standards. Microsoft Azure also performs quarterly external scans for vulnerabilities.

Requirement 2 – Don't Use Vendor-Supplied Defaults for System Passwords and more

Requirement 2 states that customers are not to use vendor-supplied defaults for system passwords and other security parameters. These other security parameters pertain to handling things like wireless router password rotations, developing configuration standards to address known security vulnerabilities, and limiting responsibility of systems. This requirement also focuses on cleaning up and locking down systems so that they are only running what is required for the system, that access is using cryptography, and that the organization is maintaining a proper inventory of components that are in scope. The organization must have policies and procedures in place for managing any vendor defaults and other security parameters.

This requirement is mostly a shared responsibility; however, Microsoft Azure can provide 83.3% reference architecture requirement coverage. Key Azure Services included in the reference architecture consist of Azure PaaS resources such as virtual machines, ARM for implementing resources, Azure networking, ASEs, AAD, Azure SQL Database, and the Azure Portal.

This requirement is further broken down into the 12 controls listed below.

2.1 This is a customer-owned control.

The customer is responsible for changing all default passwords and for removing or disabling any unnecessary accounts before installing a system on the network. Any default passwords, regardless of the software or device should be updated. Customers can leverage AAD to manage users, passwords, and password management. Customers can also use access tokens which enable clients to securely call APIs protected by Azure. Access tokens provide helpful information for use in authentication and authorization, such as the user, client, issuer, permissions, and more. It essentially provides AAD auth without using passwords.

2.1.1 Microsoft Azure covers this control.

This control does not apply to the Microsoft Azure environment due to wireless access not being permitted. All wireless protection controls are implemented and managed by Microsoft. This control is inherited from Microsoft Azure

2.2 This is a shared control.

From a PaaS responsibility, the OSSC Technical Security Services team develops security configuration standards for systems in the Microsoft Azure environment that follow and are consistent with industry-accepted hardening standards. The configurations are documented in system baselines and any relevant configurations changes are communicated to impacted

teams. Procedures are in place to monitor for compliance against the security configuration standards. These security standards are reviewed at least annually.

The customer is responsible for developing security-configuration standards for all customer-deployed system components and for making sure these configurations address all known vulnerabilities and adhere to industry-accepted hardening standards. Customers can utilize the ARM to deploy new Azure resources consistently and securely and reuse existing approved standard deployments. Customers can also leverage Azure PaaS resources such as Azure SQL database and Azure SQL managed instance which are a handled by Microsoft.

2.2.1 This is a shared control.

Microsoft Azure implements this control for PaaS operating systems on behalf of its customers.

The customer is responsible for ensuring that each server it controls has one primary function. If the customer is using Azure SQL database or Azure SQL managed instance, this is handled by Microsoft, however any deployments using on-premises servers would need to be addressed as well as any Azure virtual machines. Deploying resources using the ARM using approved deployment scripts can help ensure that systems are limited to a single primary function.

2.2.2 This is a shared control.

Microsoft configures information systems to provide only essential capabilities and specifically prohibits unnecessary functions, ports, protocols, and/or services through a secure development lifecycle and bootstrap process. This is done for all operating systems supporting the PaaS resources.

The customer is responsible for the ensuring that only necessary services, protocols, and/or services are enabled for customer-deployed resources. If the customer is using all PaaS resources, this is covered. If the customer is using Azure virtual machines or on-premises resources, they will need to make sure that any unneeded services, protocols, daemons, etc are disabled by default. Azure VMs can be deployed using the ARM based on an approved template. Azure Networking can also block any unsupported or approved protocols.

2.2.3 This is a shared control.

Microsoft configures information systems to provide only essential capabilities and specifically prohibits unnecessary functions, ports, protocols, and/or services through a secure development lifecycle and bootstrap process. This is done for all operating systems supporting the PaaS resources. All connections to the Azure environment are made through an established VPN Gateway or ExpressRoute connection which are configured to accept FIPS 140-2 validated encryption. Communication between the Azure service offerings and the Microsoft Azure Management Portal App Service Environment supports the modification of virtual networks, including the implementation of load balancing and TLS certificate requirements. Azure deploys

resources as necessary for specific system functions. All other services, protocols, daemons, etc. are disabled by default.

The customer is responsible for the ensuring that only industry-accepted security features are implemented for all insecure services, daemons, protocols, etc. for all customer-deployed resources outside of the PaaS environment. Any Azure VMs or on-premises systems should be locked down to only the services, protocols, or daemons that are needed for the function of the system. Azure VMs can be deployed using the ARM based on an approved template. Azure Networking can also block any unsupported or approved protocols.

2.2.4 This is a shared control.

Microsoft Azure performs this control for operating systems supporting the PaaS resources. The Azure team leverages industry-standard security benchmarks to assist in the creation of initial baselines. Product and technical subject matter experts review the benchmarks and, where needed, they customize the base settings. This customization may include the addition or removal of rules or rule updates. When the process is completed, the initial baseline is presented for approval. Once approved, the initial baseline is established.

The customer is responsible for making sure system configurations are established to prevent any misuse of all customer-deployed resources. Customers can leverage AAD to manage all user accounts and permissions. Any Azure VMs deployed can be configured to only be accessed with proper AAD credentials. All connections to the Azure environment should be established through a VPN Gateway or ExpressRoute connection with proper encryption. Any data stored within Azure SQL database is encrypted, if the customer is using Azure SQL managed instance, they should enable TDE. Client traffic should be using TLS.

2.2.5 This is a shared control.

Microsoft Azure performs this control for operating systems supporting the PaaS resources. Microsoft configures information systems to provide only the essential capabilities and specifically prohibits unnecessary functions, ports, protocols, and/or services through secure development and bootstrap processes.

The customer is responsible for this control for any customer-deployed resources, such as scripts, drivers, web servers, or other services. This follows the guidance for 2.2.2 and 2.2.3, systems should have any unnecessary items disabled or removed. This would include any scripts, drivers, features, subsystems, file systems, and more. This would apply to any Azure VMs or on-premises systems.

2.3 This is a customer-owned control.

The customer is responsible for making sure that industry-accepted cryptography is applied to all non-console administrative access. Customers should make sure all connections to the Azure

environment are made through an established VPN Gateway or ExpressRoute connection with valid encryption. Client connections should be utilizing TLS encryption.

2.4 This is a customer-owned control.

The customer is responsible for maintaining an accurate inventory of PCI-relevant system components. The Azure Portal can be utilized to show all Azure resources that are deployed. Resources and be grouped and tagged for easy identification for their role or purpose.

2.5 This is a customer-owned control.

The customer is responsible for documenting their security policies and operational procedures for managing all vendor default security parameters. These policies should then be distributed to all associated parties.

2.6 This is a shared control.

Microsoft Azure protects all customer data, including cardholder data, and adheres to all requirements outlined in Appendix A of the PCI DSS. More information regarding Microsoft's compliance with PCI DSS can be found in Microsoft's Trust Center located here: https://www.microsoft.com/en-us/trustcenter/compliance/pci.

Any customer that are shared hosting providers are also responsible for protecting each hosted entity's cardholder data.

Summary: Microsoft Azure has taken responsibility for all controls that fall within the PaaS infrastructure. Customers are responsible for ensuring any controls are met for customer-deployed resources as well as documenting various security policies and inventorying system components in scope for PCI DSS compliance. Customers also must make sure any non-console access is encrypted. Utilizing AAD properly helps meet the requirement for password management, whereas Azure networking and other controls help limit access to sensitive data. The deployment methods in Azure are configured to not install unnecessary services, protocols, or daemons; anything not deemed necessary is disabled by default.

Requirement 3 – Protect Stored Cardholder Data

Requirement 3 states that customers must protect stored cardholder data. This requirement is a near even split between customer-owned and Microsoft Azure owned, however Microsoft Azure can provide 68.2% reference architecture requirement coverage. Azure resources to help customers with this requirement include Azure Storage, Azure SQL Database, AAD, Azure Key Vault, ARM, and Azure PaaS resources to include TDE, managed keys, DDM, Always Encrypted, and Data Classification and Discovery.

TDE provides encryption at rest and is enabled by default for Azure SQL databases created after May 2017. TDE must be enabled for any Azure SQL managed instance databases. TDE protects against the threat of malicious offline activity by encrypting the data at rest. It performs real-time encryption and decryption of the database, any associated backups, and the transaction log files at rest without requiring any application code changes. Encryption of the entire database is done by using a symmetric key called the database encryption key and is protected by the transparent data encryption protector. This protector is either a service-managed certificate or an asymmetric key stored in the Azure Key Vault (BYOK - Bring Your Own Key). The Azure Key Vault is a cloud service that safeguards encryption keys and secrets like certifications, connection strings, and passwords. It is a key management system that allows customers to protect sensitive and business critical data by allowing only authorized applications and users access.

DDM is built-in SQL security feature that allows customers to limit sensitive data exposure by masking it to non-privileged users. This feature can greatly simplify the design and coding of security within your application. DDM allows customers to specify how much sensitive data is revealed to the end users with minimal impact to the application layer. Customers configure which database fields to hide sensitive data in the result sets of queries by configuring masking rules. DDM does not require changing any data within the database which makes it easy to use with third party databases. DDM features full masking and partial masking functions, and a random mask for numeric data. A central data masking policy acts directly on sensitive fields in the database. Customers can designate privileged users or roles that do have access to the sensitive data. Simple Transact-SQL commands define and manage masks. The purpose of DDM is to limit exposure of sensitive data, preventing users who should not have access to the data from viewing it while allowing those with a business need to have access to unmasked data. DDM does not aim to prevent database users from connecting directly to the database and running exhaustive queries that expose pieces of the sensitive data. Four types of masks are available, default which uses X's for string data types, zero for numeric data types, '01.01.1900 00:00:00.0000000' for date and time data types, and for binary data types use a single byte of ASCII value 0. Email mask with the first letter of an email address and the constant suffix '.com'. Random masking mask any numeric type to mask the original value with a random value within a specified range. Custom String gives customers the ability to expose the first and last letters and adds a custom padding string in the middle. If the original value is too short to complete the entire mask, part of the prefix or suffix will not be exposed.

Always Encrypted helps protect sensitive data at rest on the server, during movement between the client and server, and while the data is in use. It does this by ensuring that sensitive data never appears as plaintext inside the database system. Always Encrypted allows clients to encrypt sensitive data inside client applications and never reveal the encryption keys to the Database Engine. Always Encrypted is simple to use, transparent, and ready to protect your data. Client drivers have been enhanced to work in conjunction with the database engine to decrypt and encrypt data at the point of use, requiring only minimal modifications to your applications,

typically a connection string modification. Encryption keys are managed outside of the database for maximum safety and separation of duties. Only authorized users with access to the encryption keys can see unencrypted data while using your applications. The encryption keys used with Always Encrypted can be stored in the Azure Key Vault. Ensuring the encrypted data and its corresponding keys are not ever revealed in plaintext to the database system, Always Encrypted allows you to store your sensitive data in Azure SQL Database and Azure SQL Database managed instance with confidence. Always Encrypted can also restrict high privilege users in your organization, such as database administrators.

Always Encrypted with secure enclaves will be introduced in SQL Server 2019 and PaaS at some point in the future. Always Encrypted with secure enclaves addresses certain limitations by allowing computations on plaintext data inside a secure enclave on the server side. A secure enclave is a protected region of memory within the SQL Server process, and acts as a trusted execution environment for processing sensitive data inside the SQL Server engine. A secure enclave appears as a black box to the rest of the SQL Server and other processes on the hosting machine. There is no way to view any data or code inside the enclave from the outside, even with a debugger.

Data discovery and classification (currently in preview) provides customers the ability for discovering, classifying, labeling, and protecting sensitive data within their databases. This gives organizations the ability to classify their most sensitive data which helps improves their information protection stature. By discovering, classifying, and labeling sensitive data, organizations can take the next step in securing that data with stronger access controls, masking, encrypting, and auditing that data. Data discovery and classification can help organizations meet data privacy standards and regulatory requirements.

The classification engine scans the database and identifies columns containing potentially sensitive data and provides an easy way to review and apply the appropriate classification recommendations via the Azure portal. Sensitivity classification labels can be persistently tagged on columns using new classification metadata attributes that have been introduced in the SQL Engine. This metadata can then be utilized for more advanced auditing and protection scenarios. The current database classification state can be viewed in a detailed dashboard in the portal and can also be downloaded as a report to be used for compliance and auditing purposes.

This requirement is further broken down into the 22 controls listed below.

3.1 This is a shared control.

Microsoft Azure provides tools to securely delete data including immediately removing the index from the primary location and removal of any geo-replicated copy of the data asynchronously within Azure Storage. Microsoft Azure is responsible for ensuring that customer data designated for deletion is securely decommissioned using NIST 800-88 compliant protocols specified in its Secure Disposal policies.

The customer is responsible for defining and implementing data retention policies, procedures, and processes as required by this control. When using Azure SQL database, customers can leverage Azure Automation to schedule the deletion of data that has reached its retention period. Azure Automation is a cloud service that customers can use to create runbooks to perform task in Azure. Data deleted in Azure SQL database is deleted per Microsoft Azure's establish policies. Data classification and discovery can be utilized to scan for sensitive card holder data and mark its classification level for future reporting and auditing needs.

3.2 This is a customer-owned control.

The customer is responsible for preventing the storage of sensitive authentication data after authorization. Microsoft offers different solutions that can be used to store cardholder data; it will be the customer's responsibility to determine what cardholder data is stored within their Azure subscription. If the customer must store sensitive authentication data, the database should be encrypted using TDE and implement Always Encrypted for the appropriate columns.

3.2.1 This is a customer-owned control.

The customer is responsible for preventing the storage of full track data after authorization.

3.2.2 and 3.2.3 This is a customer-owned control.

The customer is responsible for preventing the storage of the card verification value after authorization.

3.3 This is a customer-owned control.

The customer is responsible for redacting PANs. Personnel with a legitimate business justification can view the full PAN however everyone else should see redacted data. DDM can be utilized to mask the full PAN. AAD should be utilized to manage which users or groups see masked or unmasked data.

3.4 This is a customer-owned control.

The customer is responsible for securely storing PAN data, rendering it unreadable. Customers can use Azure SQL Database which has TDE enabled by default. If the customer is using Azure SQL managed instance, TDE should be enabled.

3.4.1 This is a customer-owned control.

The customer is responsible for managing logical access, if disk encryption is used. The management of logical access should be independent from the native operating system (OS). Decryption keys cannot be associated with user accounts. Azure SQL Database and Azure SQL Managed Instance can utilize TDE and Always Encrypted with Azure Key Vault integration. OS-level keys are managed by Microsoft.

3.5 This is a shared control.

Microsoft has policies, procedures, and mechanisms established for effective key management to support encryption of data in storage and in transmission for the key components of the Azure service. Azure provides each customer subscription with an associated logical certificate store (Key Vault) that enables automatic deployment of service-specific certificates, and to which customers can upload their own. For internal corporate data and transmission encryption, Microsoft has established procedures to manage cryptographic keys throughout their lifecycle. Microsoft Azure uses Microsoft's corporate PKI infrastructure.

Certificates used in Azure are x.509 v3 certificates and can be signed by another trusted certificate or they can be self-signed. The certificate store is independent of any hosted service, so it can store certificates regardless of whether they are currently being used by any of those services. These certificates and other credentials uploaded to Azure are stored in an encrypted form.

The customer is responsible for implementing and documenting their procedures outlining the protection of keys used to secure stored cardholder data. Customers should the service managed keys with TDE or BYOK utilizing the Azure Key Vault.

3.5.1 This is a shared control.

This control for Microsoft Azure is covered under the same control for 3.5.

Customers that are service providers are responsible for documenting their cryptographic architecture. This documentation should enable any entities using the service to understand algorithms, protocols, and cryptographic keys used to protect cardholder data.

3.5.2 This is a customer-owned control.

The customer is responsible for limiting access to cryptographic keys to only necessary personnel. AAD and the Azure Key Vault can be utilized for this requirement.

3.5.3, 3.5.4, 3.6.1, 3.6.2, 3.6.3, 3.6.4, 3.6.5, and 3.6.7 are shared controls.

These controls for Microsoft Azure are covered under the same control for 3.5.

The customer can take advantage of AAD for provisioning access to keys and the Azure Key Vault for their key management for storing and protecting their cryptographic keys.

3.6.6 This is a customer-owned control.

The customer is responsible for documenting and implementing key-management processes and procedures for cryptographic keys. This should include split knowledge and dual control operations if manual cleartext key-management is used. The customer should take advantage of AAD for provisioning access to keys and the Azure Key Vault for their key management process.

3.6.8 This is a customer-owned control.

The customer is responsible for documenting and implementing key-management processes and procedures for cryptographic keys. This should include the acknowledgment of responsibilities from key custodians.

3.7 This is a customer-owned control.

The customer is responsible for documenting and implementing cardholder data protection policies and procedures.

Summary: Any storing of cardholder data must be protected. Any data within Microsoft Azure is covered with policies, procedures, and mechanisms for supporting the encryption of data in storage and in transmission. Processes also exist for the deletion of data. Any data stored or transmitted by the customer falls within the customer's responsibility to ensure proper controls are in place to protect said data. Numerous services exist within Azure to assist with meeting and mitigating those controls; AAD assists with any user and role provision, TDE for encryption at rest, Always Encrypted for column level encryption, Azure Key Vault for encryption key storage, DDM for masking sensitive data, data classification and discovery, and Azure SQL database or managed instance for database storage.

Requirement 4 – Encrypt Transmission of Cardholder Data

Requirement 4 states that customers are to encrypt transmission of cardholder data across open, public networks. This requirement covers using strong cryptography and security protocols, securing any wireless networks, not sending unprotected PANs across messaging applications, and having proper security policies in place.

Microsoft Azure can provide 75% reference architecture requirement coverage. To secure the network traffic, customers can leverage Azure networking, application gateways, and ASEs. With ASEs, customers are isolated to just running a single application in a virtual network. This provides a strong layer of security.

Microsoft Azure has already secured the backend infrastructure by defining policies, procedures, and mechanisms for effective key management to support encryption of data. Logical certificate stores are provided to each customer that enables automatic deployment of service-specific certificates and customers can use their own. For internal corporate traffic, Microsoft manages the cryptographic keys throughout their lifecycle. Certificates used in Azure are x.509 v3 certificates and can be signed by another trusted certificate or they can be self-signed.

This requirement is further defined by the 4 controls listed below.

4.1 This is a shared control.

This control for Microsoft Azure is covered under the same control policies as 3.5. The Azure infrastructure is secure by leveraging Azure networking, application gateways, and ASEs.

The customer is responsible for implementing strong cryptography and security protocols to safeguard sensitive cardholder data in-transit. Customers should be using TLS 1.2 encryption for all data in transit over public networks with Azure. Starting June 30th, 2018, all new apps in the ASE are created with TLS 1.2 by default. The Azure Key Vault can handle requesting and renewing TLS certificates. It provides a robust solution for certificate lifecycle management.

4.1.1 Microsoft Azure covers this control.

This control is not applicable to the Microsoft Azure environment because wireless access is not permitted within the Microsoft Azure network environment.

4.2 This is a customer-owned control.

There is no messaging solution within the scope of the systems deployed on Azure.

The customer is responsible for secure transmission of all primary account numbers (PAN). Any transmission of PAN data through a messaging system must be secure.

4.3 This is a customer-owned control.

The customer is responsible for documenting security policies and operational procedures for encrypting in-transit cardholder data. These policies and procedures should be distributed to all associated parties.

Summary: All cardholder data transmissions must be encrypted. Microsoft Azure covers any transmissions within the data center; however, the customer has responsibility for documenting the security and operational procedures for encrypting transmissions of cardholder data as well as making sure that unprotected transmission of PAN data never happens across end-user messaging systems. The customer is also responsible for encrypting all transmission of cardholder data to Azure. Leveraging TLS 1.2 or greater with Azure Key Vault integration for certificate management helps mitigate this requirement.

Requirement 5 – Protect All Systems Against Malware

Requirement 5 states that customers are to protect all systems against malware and regularly update antivirus software or programs. Microsoft Azure can provide 83.3% reference architecture requirement coverage leveraging Azure PaaS resources since all VMs are constantly scanned to validate that anti-malware clients are installed and current.

The Azure PaaS VMs are constantly scanned to validate that anti-malware clients are installed and current.

This requirement is mostly owned by Microsoft Azure and has the 6 controls listed below.

5.1 Microsoft Azure covers this control.

Hosts in the scope boundary are scanned to validate anti-virus clients are installed and current signature-definition files exist. Microsoft Online Services uses anti-virus tools to scan nodes. Weekly and real-time virus scans are configured, and alerts are generated to the administrators upon detection. Viruses are either blocked or quarantined depending on the type.

When providing the anti-malware solution for virtual machines, Microsoft Online Services is responsible for ensuring the service is highly available, definitions are updated regularly, that configuration through the management portal is effective, and that the software detects and protects against known types of malicious software. Azure-managed hosts in the scope boundary are scanned to validate anti-virus clients are installed and current signature-definition files exist.

5.1.1 – 5.3 Microsoft Azure covers these controls.

These controls for Microsoft Azure are covered under the same control policies as 5.1.

5.4 This is a customer-owned control.

The customer is responsible for documenting security policies and operational procedures for protecting systems against malware. These policies and procedures should be distributed to all associated parties.

Summary: Microsoft scans for viruses within their scope to protect all systems against malware and viruses. The customer must document their security policies and operational procedures for protecting against malware.

Requirement 6 – Develop and Maintain Secure Systems and Applications

Requirement 6 states that customers are to develop and maintain secure systems and applications. This requirement is mostly a shared responsibility between Microsoft Azure and the customer.

Microsoft provides many resources to help customers monitor their security state and securely communicate to Azure resources. The Azure Security Center is a unified infrastructure security management system that strengthens customers security posture of their data centers and provides advanced threat protection across hybrid workloads in the cloud, as well as on-premises. The Azure Security Center provides customers with tools to assesses their environment and understand the status of their resources. Customers can assess workloads and raise threat prevention recommendations and threat detection alerts. The Azure Security Center

is natively integrated making deployment fast and easy. The Azure Security Center continuously discovers new resources that are being deployed across your workloads and assesses whether they are configured based on security best practices, if not, they're flagged in the customers list of recommendations.

SQL Vulnerability Assessment is a service that can discover, track, and help customers remediate potential database vulnerabilities. This service can be used to help meet compliance requirements that require database scan reports, meet data privacy standards, and monitor a dynamic database environment where changes are difficult to track.

The Vulnerability Assessment is built into the Azure SQL database service and uses a knowledge base of rules that flag security vulnerabilities. The service highlights deviations from best practices, such as misconfigurations, excessive permissions, and unprotected sensitive data. These rules are based on Microsoft's best practices and focus on security issues that present the biggest risk to your database. The knowledge base rules cover both database-level issues as well as server-level security issues, such as server firewall settings and server-level permissions. Many requirements from various regulatory bodies for compliance standards are included in the rules.

The results of the scan include actionable steps to resolve each issue and provides customized scripts when applicable. Assessment reports can be customized by setting an acceptable baseline for your organization for feature configurations, permission configurations, and more. Scans can be performed on demand or customers can enable periodic recurring scans that automatically run a scan on the database once per week.

Advanced Threat Detection detects anomalous activities that indicate unusual and potentially harmful attempts to access or exploit databases and is part of the Azure Security Center. This service provides customers the ability to detect and respond to potential threats as they occur by providing security alerts on anomalous activities. Users can receive alerts on suspicious database activities, potential vulnerabilities, SQL injection attacks, in addition to anomalous database access and queries patterns. Advanced Threat Detection integrates alerts with the Azure Security Center which include details about the suspicious activity and recommended action on how to investigate and mitigate the threat. Advanced Threat Protection makes it simple and easy to address potential threats to the database without the need to be a security expert or manage advanced security monitoring systems.

Microsoft Azure can provide 11.5% reference architecture requirement coverage.

This requirement is broken down into the 26 controls listed below.

6.1 This is a shared control due to Microsoft using information from numerous outside sources to update security vulnerability procedures.

Microsoft Azure manages security vulnerabilities for the operating systems supporting PaaS resources. The Microsoft Security Response Center regularly monitors external security vulnerability awareness sites. As part of the routine vulnerability management process, Microsoft evaluates its exposure to these vulnerabilities and leads action across Microsoft to mitigate risks when necessary.

Procedures have been established and implemented to scan for vulnerabilities on hypervisor hosts in the scope boundary. Vulnerability scanning is performed on server operating systems, databases, and network devices with the appropriate vulnerability scanning tools. The vulnerability scans are performed on a quarterly basis at minimum. Microsoft Azure contracts with independent assessors to perform penetration testing of the Microsoft Azure boundary. Red-Team exercises are also routinely performed, and results used to make security improvements.

The customer is responsible for identifying security vulnerabilities using information sourced from a reputable third-party. This information should be used for assigning risk to new security vulnerabilities. Customers can utilize the Azure Security Center for monitoring security risk and the vulnerability assessments feature to perform weekly scans of their Azure resources.

6.2 Microsoft Azure covers this control.

Microsoft Azure is responsible for ensuring all network devices and hypervisor operating system software is protected from known vulnerabilities by installing applicable vendor-supplied security patches. A patch management process exists to ensure that operating system level vulnerabilities are prevented and remediated in a timely manner. Production servers are scanned to validate patch compliance on a monthly basis.

6.3 This is a shared control.

Microsoft Azure applications and endpoints are developed in accordance with the Microsoft Security Development Lifecycle (SDL) methodology which is in line with DSS requirements. Microsoft applies information system security engineering principles in the specification, design, development, implementation, and modification of the information system. The Microsoft SDL process is followed for all engineering and development projects. The Microsoft SDL process includes the following phases which implement standard security engineering principles across all Microsoft Online Services systems:

> Phase 1: Requirements - The Requirements phase includes the project inception, when the organization considers security and privacy at a foundational level, and also a cost analysis, when determining if development and support costs for improving security and privacy are consistent with business needs.
>
> Phase 2: Design - The Design phase is when the organization builds the plan for how to take the project through the rest of the SDL process. From implementation, to verification, to release. During the Design phase the organization establishes best

practices to follow for this phase by way of functional and design specifications, and by performing risk analysis to identify threats and vulnerabilities in the software.

Phase 3: Implementation - The Implementation phase is when the organization creates the documentation and tools the customer uses to make informed decisions about how to deploy the software securely. The Implementation phase is when the organization establishes development best practices to detect and remove security and privacy issues early in the development cycle.

Phase 4: Verification - During the Verification phase, the organization ensures that the code meets the security and privacy tenets established in the previous phases. This is accomplished through security and privacy testing, and a security push. This is a team-wide focus on threat model updates, code review, testing, and thorough documentation review and edit. A public release privacy review is also completed during the Verification phase.

Phase 5: Release - The Release phase is when the organization prepares the software for consumption and prepares for what happens once the software is released. One of the core concepts in the Release phase is response planning. Mapping out a plan of action, should any security or privacy vulnerabilities be discovered in the release which carries over to post-release, as well, in terms of response execution. To this end, a Final Security Review and privacy review is required prior to release. As established by the Microsoft Online Services Security Policy, application code changes must be reviewed and approved by the Microsoft Online Services Security team.

The customer is responsible for securely developing software applications in accordance with PCI DSS compliance, based on industry standards and/or best practices, and by incorporating information security throughout the software-development life cycle.

6.3.1 This is a customer-owned control.

A Final Security Review (FSR) is performed for major releases prior to production deployment by a designated Security Advisor outside of the Azure development team to ensure only applications ready for production are released. As part of this final review it is ensured that all test accounts and test data have been removed.

The customer is responsible for removing test accounts and test data for customer-deployed resources.

6.3.2 This is a shared control.

This control for Microsoft Azure is covered under the same control policies as 6.3.

The customer is responsible for code review of customer-deployed resources before release to production.

6.4.1 This is a shared control.

Microsoft follows NIST guidance regarding security considerations in software development in that information security must be integrated into the SDLC from system inception. Continual integration of security practices in the Microsoft SDL enables early identification and mitigation of security vulnerabilities and misconfigurations; awareness of potential software coding challenges caused by required security controls; identification of shared security services and reuse of security best practices tools which improves security posture through proven methods and techniques; and enforces Microsoft's already comprehensive risk management program.

Microsoft Azure has established change and release management processes to control implementation of major changes including:

- The identification and documentation of the planned change.
- Identification of business goals, priorities and scenarios during product planning.
- Specification of feature/component design.
- Operational readiness review based on a pre-defined criteria/check-list to assess overall risk and impact.
- Testing, authorization and change management based on entry/exit criteria for development, integration testing, Pre-production and production environments as appropriate.

The customer is responsible for implementing change control processes and procedures for customer-deployed resources. This should include the separation of development/test environments from production environments by enforceable access controls.

6.4.2 – 6.5.10, These are shared controls.

This control for Microsoft Azure is covered under the same control policies as 6.4.1.

The customer is responsible for implementing change control processes and procedures for customer-deployed resources. These procedures should include the following:

- The separation of development/test environments from production environments by enforceable access controls.
- Prohibit the use of production data in development/test environments.
- The removal of test accounts and data before release to production.
- Documentation detailing the impact of installing security patches and software modifications.
- Documented approval of security patch and software modification installation by authorized parties.
- Testing security patches and software modifications for adverse impact.
- Roll-back procedures.

- Ensure all PCI DSS requirements are implemented after a change has been made to the customer-deployed resource.
- Training developers in secure coding techniques to prevent injection vulnerabilities, buffer overflow vulnerabilities, insecure cryptographic storage, insecure communications, improper error handling, "high risk" vulnerabilities, cross-site scripting vulnerabilities, improper access control, cross-site request forgery vulnerabilities, and to prevent broken authentication and session management vulnerabilities.

6.6 This is a customer-owned control.

The customer is responsible for addressing new threats on customer-deployed web applications. The Azure Security Center can monitor for any threats and deviations from your security baseline. The Application Service Environment can make sure that resources are in complete tenant isolation and Application Gateways can making routing decisions to ensure only trusted addresses can reach resources.

6.7 This is a customer-owned control.

The customer is responsible for documenting security policies and operational procedures for developing and maintaining secure systems. These policies and procedures should be distributed to all associated parties.

Summary: The Azure Security Center can provide real-time monitoring and scanning of customer workloads to find security risk and anomalies with data access. Vulnerability scans should be performed regularly on all systems. Microsoft Azure handles the PaaS environment however customers should consider contracting with a qualified third-party to perform additional vulnerability scans against their environment. Code reviews and change control processes should be designed, implemented, and followed for all development. Processes should be in place to ensure that coding practices are following the current security best practices to prevent introducing vulnerabilities.

Requirement 7 – Install and Maintain a Firewall Configuration to Protect Cardholder Data

Requirement 7 states that customers are to restrict access to cardholder data by business "need to know". This requirement is on the customer to manage and control, however Microsoft provides many features and services to help meet this requirement. AAD manages users and groups to limit access by using roles and permissions. RBAC helps customers manage who has access to Azure resources so that they can only grant access to card holder systems by business need. Customers can use Row-level security (RLS) to further restrict access to certain rows within a table to only those who require access. DDM can also be used to mask sensitive data only revealing full column level data to those who require it.

RLS enables customers to use group memberships or execution context to limit access to rows in a database table. RLS simplifies the design and coding of security within your application by

applying restriction logic in the database tier rather than away from the data in another application tier. RLS implements restrictions on data row access and does not require any physical change to the data. The database system applies the access restrictions each time that data access is attempted from any tier making your security system more reliable and robust by reducing the surface area of your security system. RLS supports filter predicates that silently filter the rows available to read operations (SELECT, UPDATE, and DELETE) and block predicates that explicitly block write operations (AFTER INSERT, AFTER UPDATE, BEFORE UPDATE, and BEFORE DELETE) that violate the predicate. Access to row-level data in a table is restricted by a security predicate defined as an inline table-valued function. The function is then invoked and enforced by a security policy. For the filter predicates, the application is completely unaware of rows that are filtered from the result set. If all rows are filtered, then a null set will be returned. For block predicates, operations that violate the predicate will fail with an error. Filter predicates are applied while reading data in the base table and affect all get operations: SELECT, DELETE, and UPDATE. Users can't select, delete, or update rows that are filtered; however, it is possible to update rows in such a way that they'll be filtered afterward. Block predicates affect all write operations. RLS filter predicates are functionally equivalent to appending a WHERE clause. It's highly recommended that you create a separate schema for the RLS objects, predicate function, and security policy.

RBAC is access management for Azure. RBAC helps manage who has access to Azure resources and what users can do with those resources, and what areas they have access to. It is an authorization system built on ARM that provides customers with a fine-grained access management for Azure resources. This ultimately provides a separation of duties. Organizations can segregate duties within teams and grant only the amount of access to users that are required for them to fulfill their job. Organizations goal should be granting the least privilege required for users to get their work done.

Microsoft Azure can provide 55.6% reference architecture requirement coverage.

This requirement is further broken down into the 9 controls listed below.

7.1 This is a customer-owned control.

The customer is responsible for limiting access to system components and cardholder data to only those individuals whose job requires such access. This should include limiting and restricting access to the Azure Management Portal as well as specifying accounts or roles with permission to create, modify, or delete Azure services. Customers should be using AAD for user authentication and RBAC to restrict what users have access to within the Azure Portal.

7.1.1 This is a customer-owned control.

The customer is responsible for defining access requirements for system components. Least privilege should be taken into consideration when defining access requirements. Customers should use AAD for user authentication, RBAC for role assignments for access to Azure

resources, RLS for limiting access to sensitive data, and DDM for limiting access to full cardholder data.

7.1.2 and 7.1.3 These are customer-owned controls.

The customer is responsible for restricting privileged access and assigning access based on job classification. This can be accomplished using AAD and RBAC.

7.1.4 This is a customer-owned control.

The customer is responsible for documenting access requirements. These privileges must be documented and approved by authorized personnel.

7.2.1 This is a customer-owned control.

The customer is responsible for establishing an access control system. This should include the management of access to the Azure Portal as well as specifying accounts or roles with permission to create, modify, or delete Azure services. This can be accomplished using AAD and RBAC.

7.2.2 and 7.2.3 These are customer-owned controls.

The customer is responsible for establishing an access control system. This should include the ability to assign access based on job classification and to configure a "deny-all" setting in the access control system. This can be accomplished using AAD and RBAC.

7.3 This is a customer-owned control.
The customer is responsible for documenting security policies and operational procedures for access control. These policies and procedures should be distributed to all associated parties.

Summary: Least privileged should be the standard regarding provisioning access to any system that contains cardholder data. Microsoft Azure provides AAD for maintaining users, groups, and permissions, as well as data access controls using RLS and DDM. RBAC should be well planned out to provide least privileges to users within the Azure Portal for access to resources and who can create and delete resources. Requirement 7 is the customer's responsibility; however, Microsoft Azure provides the tools to meet this requirement.

Requirement 8 – Identify and Authenticate Access to System Components

Requirement 8 states that customers are to identify and authenticate access to system components. This requirement is more of the customer's responsibility, with some shared controls. Through using AAD, RBAC, and the Azure Key Vault, many of these controls are easily met. AAD supports Multi-Factor Authentication (MFA) which adds a significant challenge for attackers. With MFA, even if an attacker where to gain access to a valid username and password, it is useless without also having the additional authentication method. This could be something

you know such as a password, something you have such as a phone, or something biometric. MFA helps organizations safeguard access to data and applications while keeping things simple for users. It adds a strong layer of additional security by requiring a second form of authentication and offers a range of easy to use authentications methods. By default, SQL authentication is enabled for Azure SQL Database. SQL Server authentication can be beneficial in certain scenarios such as supporting legacy applications where AAD may not be supported. Using SQL Server authentication means managing strong credentials manually, protect the credentials in the connection string, and protect the credentials passed over the network from the web server to the database. Ideally, in a highly protected environment, AAD with MFA is used instead of SQL Authentication and the built in SQL Authentication accounts are secured and not used.

Microsoft Azure can provide 41.7% reference architecture requirement coverage.

This requirement is further broken down into the 24 controls listed below.

8.1.1 This is a customer-owned control.

The customer is responsible for defining and implementing policies and procedures outlining proper user identification of all non-consumer accounts for customer-deployed resources. This should include the assignment of unique user IDs. Customers should utilize AAD with MFA and not allow any sharing of credentials.

8.1.2 This is a customer-owned control.

The customer is responsible for defining and implementing policies and procedures outlining proper user identification of all non-consumer accounts for customer-deployed resources. This should include the modification of user IDs, credentials, and other identifier objects. Customers should use RBAC with least privilege to establish who can perform which task based on job function.

8.1.3 This is a customer-owned control.

The customer is responsible for defining and implementing policies and procedures outlining proper user identification of all non-consumer accounts for customer-deployed resources. This should include the immediate revocation of terminated accounts. Identity management policies should be tied to automatically disable accounts in AAD when users are terminated.

8.1.4 This is a customer-owned control.

The customer is responsible for defining and implementing policies and procedures outlining proper user identification of all non-consumer accounts for customer-deployed resources. This should include the removal or disabling of inactive user accounts. Password expiration policies are defined with AAD and for RBAC through the Azure Portal.

8.1.5 This is a shared control.

Microsoft Azure has adopted applicable corporate and organizational security policies, including an Information Security Policy. The policies have been approved, published, and communicated to Microsoft Azure. The Information Security Policy requires that access to Microsoft Azure assets be granted based on business justification, with the asset owner's authorization and limited based on "need-to-know" and "least-privilege" principles. In addition, the policy also addresses requirements for access management lifecycle including access provisioning, authentication, access authorization, removal of access rights, and periodic access reviews.

The customer is responsible for defining and implementing policies and procedures outlining proper user identification of all non-consumer accounts for customer-deployed resources. This should include the management of remote access of vendor accounts. Vendor access should be provisioned with AAD accounts with access policies defined.

8.1.6 This is a customer-owned control.

The customer is responsible for defining and implementing policies and procedures outlining proper user identification of all non-consumer accounts for customer-deployed resources. This should include a lockout period after not more than six failed sign-in attempts and can be configured in AAD.

8.1.7 This is a customer-owned control.

The customer is responsible for defining and implementing policies and procedures outlining proper user identification of all non-consumer accounts for customer-deployed resources. This should include a minimum lockout period of 30 minutes and can be configured in AAD.

8.1.8 This is a customer-owned control.

The customer is responsible for defining and implementing policies and procedures outlining proper user identification of all non-consumer accounts for customer-deployed resources. This should include the lockout of sessions idle for more than 15 minutes and can be configured in AAD.

8.2 This is a shared control.

Microsoft's Azure portal enforces user authentication over TLS. For users connecting to Azure hosted resources, AAD authentication is required.

The customer is responsible for the assignment of unique user IDs and enforcing proper user-authentication of non-consumer accounts. Organizations should be using AAD with MFA.

8.2.1 Microsoft Azure covers this control.

Microsoft's Azure portal enforces user authentication over TLS. For users connecting to Azure hosted resources, AAD authentication is required. For resources accessed through the Azure portal and AAD, Microsoft meets the requirements of this control.

8.2.2 Microsoft Azure covers this control.

Microsoft's Azure portal enforces user authentication over TLS. For users connecting to Azure hosted resources, AAD authentication is required. For resources accessed through the Azure portal and AAD, Microsoft meets the requirements of this control.

8.2.3, 8.2.4, 8.2.5, and 8.2.6 These are customer-owned controls.

The customer is responsible for the generation and assignment of unique passwords. This should include adherence to PCI DSS compliance password requirements and those should be set within AAD.

8.3, 8.3.1, and 8.3.2 These are customer-owned controls.

The customer is responsible for implementing multi-factor authentication controls for administrative access and remote access to the cardholder data environment. AAD with MFA should be utilized.

8.4 This is a customer-owned control.

The customer is responsible for documenting security policies and procedures for authentication. These policies and procedures should be distributed to all associated parties and include PCI DSS compliance requirements.

8.5 This is a customer-owned control.

The customer is responsible for adhering to PCI DSS authentication requirements. Utilizing AAD simplifies this control.

8.5.1 Not Applicable. This control does not apply to Microsoft Azure. Azure hosts multiple customer environments.

8.6 This is a shared control.

Microsoft Azure services and infrastructure must at a minimum meet Microsoft internal IT requirement, but an internal organization can increase the strength past this standard, on their own discretion and to meet their security needs.

Microsoft's Azure portal enforces user authentication over TLS. For users connecting to Azure hosted resources, AAD authentication is required.

The customer is responsible for adhering to PCI DSS authentication requirements. Where other authentication methods are used, they must be assigned as follows:

- Authentication mechanisms must be assigned to an individual account and not shared among multiple accounts.
- Physical and/or logical controls must be in place to ensure only the intended account can use that mechanism to gain access.

Microsoft's Azure portal enforces user authentication over TLS. For users connecting to Azure hosted resources, AAD authentication is required. Any certificates, tokens, or encryption keys can be stored in the Azure Key Vault.

8.7 This is a customer-owned control.

The customer is responsible for limiting access to any databases containing card holder data. All access to the data, including queries and user actions, should be through programmatic methods. Only database administrators should have the ability to directly access or query the databases. Application IDs for database applications can only be used by the application (not by individual users or other non-application processes). Leveraging AAD to enforce RBAC for all Azure services can help mitigate this control since access to cardholder data and other user actions in databases are controlled with AAD to ensure that only approved users are making restricted actions.

8.8 This is a customer-owned control.

The customer is responsible for documenting security policies and operational procedures for identification and authentication. These policies and procedures should be distributed to all associated parties.

Summary: Handling authentication, user lockout, access revocation, removal of inactive and terminated accounts, and password management is a critical component of maintaining PCI DSS compliance. While Microsoft handles a portion of the controls for requirement, it is up to the customer to make sure that policies and controls are in place for user-authentication of non-consumer accounts. Utilizing AAD Authentication with MFA, the Azure Key Vault, and the Azure Portal can help meet these requirements.

Requirement 9 – Restrict Physical Access to Cardholder Data

Requirement 9 states that customers are to restrict physical access to cardholder data. In Microsoft Azure, this requirement is mostly taken care of for the customer by Microsoft since they manage the data centers.

This requirement is further broken down into the 26 controls listed below.

Customers do not have physical access to any system resources in Azure datacenters; all physical and environmental protection controls are implemented and managed by Microsoft Azure. Microsoft Azure can provide 92.3% reference architecture requirement coverage.

9.1 Microsoft Azure covers this control.

Microsoft Azure is responsible for implementing, enforcing, and monitoring physical access security for data centers. Access to all Microsoft buildings is controlled, and access is restricted to those with card reader (swiping the card reader with an authorized ID badge) and biometrics for entry (datacenters). Front desk personnel are required to positively identify employees or authorized contractors without ID cards. All guests are required to wear guest badges and be escorted by authorized Microsoft personnel. Datacenter entrances are always guarded by security personnel and access is controlled through security personnel, authorized badges, biometrics, and locked doors and closed-circuit television monitoring.

Azure employees and contractors must have a business need to enter a Microsoft datacenter and have received prior approval by Microsoft personnel. Doors between areas of differing security require authorized badge access, are monitored through logs and cameras, and audited on a regular basis. Failure to abide by the Microsoft Datacenter Security Policies is grounds for instant dismissal of the employee. Front desk personnel are required to positively identify employees or authorized contractors without ID cards. Staff must always wear identity badges and are required to challenge or report individuals without badges. Guests must be escorted by authorized Microsoft personnel when within the datacenter.

9.1.1 – 9.4 Microsoft Azure covers these controls.

This control is covered by the policies outlined in control 9.1.

9.4.1 Microsoft Azure covers this control.

In addition to the physical entry controls operational procedures have been implemented to restrict physical access to authorized employees, contractors and visitors:

- Authorization to grant temporary or permanent access to Microsoft datacenters is limited to authorized staff. The requests and corresponding authorization decisions are tracked using a ticketing/access system.
- Badges are issued to personnel requiring access after verification of identification.
- Visitors are always required to be escorted. The escort's access within the datacenter is logged and if necessary, can be correlated to the visitor for future review.
- Datacenter Management performs a quarterly access list review and takes any follow up actions necessary.

9.4.2, 9.4.3, and 9.4.4 Microsoft Azure covers this control.

This control is covered by the policies outlined in control 9.4.1.

9.5 Microsoft Azure covers this control.

Azure equipment is placed in environments which have been engineered to be protected from theft and environmental risks such as fire, smoke, water, dust, vibration, earthquake, and electrical interference. Azure does not store any customer data on durable media. The use or

storage of Microsoft-managed information processing equipment and/or media containing sensitive data (as defined by Microsoft policy) outside a Microsoft-managed facility must be approved by the asset owner. Protection afforded to such equipment and storage media is commensurate with the protection it is afforded on-site.

9.5.1 Microsoft Azure covers this control.

This control is covered by the policies outlined in control 9.5.

9.6.1 This is a customer-owned control.

The customer is responsible for implementing media protection mechanisms. This should include the classification of media. Data Classification and Discovery can be used to classify data prior to its export.

9.6.2, 9.6.3, and 9.7 Microsoft Azure covers these controls.

Microsoft Azure implements this requirement on behalf of customers.

9.7.1 Microsoft Azure covers this control.

Microsoft develops and documents an inventory of information system components at a level of granularity deemed necessary for tracking and reporting. Microsoft Online Services has implemented a formal policy that requires major assets used to provide services to be accounted for and have a designated asset owner.

9.8 Microsoft Azure covers this control.

Microsoft uses best practice procedures and a NIST 800-88 compliant media wiping solution when disposing of media. For hard drives that can't be wiped, Microsoft uses a destruction process that destroys the media (e.g., disintegrates, pulverizes, or incinerates) and renders the recovery of information impossible. The appropriate means of disposal is determined by the asset type. Records of the destruction are retained.

9.8.1 Microsoft Azure covers this control.

This control is covered by the policies outlined in control 9.8.

9.8.2 Microsoft Azure covers this control.

Data destruction techniques vary depending on the type of data object being destroyed, whether it be subscriptions, storage, virtual machines, or databases. In the Microsoft Azure multi-tenant environment, careful attention is taken to ensure that one customer's data is not allowed to either "leak" into another customer's data, or when a customer deletes data, no other customer (including, in most cases, the customer who once owned the data) can gain access to that deleted data.

Microsoft Azure follows NIST 800-88 Guidelines on Media Sanitization, which address the principal concern of ensuring that data is not unintentionally released. These guidelines encompass both electronic and physical sanitization.

9.9, 9.9.1, 9.9.2, and 9.9.3 Microsoft Azure covers these controls.

Access to all Microsoft buildings is controlled, and access is restricted to those with card reader (swiping the card reader with an authorized ID badge) and biometrics for entry. Front desk personnel are required to positively identify employees or authorized contractors without ID cards. All guests are required to wear guest badges and be escorted by authorized Microsoft personnel.

Datacenter entrances are always guarded by security personnel and access is controlled through security personnel, authorized badges, biometrics, and locked doors and closed-circuit television monitoring.

Azure employees and contractors must have a business need to enter a Microsoft datacenter and have received prior approval by Microsoft personnel. Doors between areas of differing security require authorized badge access, are monitored through logs and cameras, and audited on a regular basis. Failure to abide by the Microsoft Datacenter Security Policies is grounds for instant dismissal of the employee.

Access to Microsoft buildings is controlled using smart cards for Microsoft offices and biometrics for entry into datacenters. Front desk personnel are required to positively identify employees or authorized contractors without ID cards. Staff must always wear identity badges and are required to challenge or report individuals without badges. Guests must be escorted by authorized Microsoft personnel when within the datacenter.

9.10 This is a customer-owned control.

The customer is responsible for documenting security policies and operational procedures for physical access to cardholder data. These policies and procedures should be distributed to all associated parties.

Summary: Microsoft Azure has this requirement mostly covered for the customer since this is directed at the physical access to controls within the Azure data centers. Customers just need to implement media protection mechanisms for any external distribution of media and making sure they have documented Microsoft's security policies and operational procedures for how they restrict physical access.

Requirement 10 – Track and Monitor All Access to Network Resources and Cardholder Data

Requirement 10 states that customers must track and monitor all access to network resources and cardholder data. This requirement is overwhelmingly a shared requirement where Microsoft and the customer have responsibility for tracking and monitoring access to network resources. Microsoft provides customers in the PaaS environment the tools to lock down and monitor their resources. Microsoft Azure can provide 75.0% reference architecture requirement coverage by leveraging AAD, Azure PaaS resources, Azure Portal, Log Analytics, and the Azure Security Center.

Log Analytics is the primary tool in the Azure portal for writing log queries and interactively analyzing their results. Queries can be used to search terms, identify trends, analyze patterns, and provide many other insights based on the data.

This requirement is further broken down into the 32 controls listed below.

10.1 This is a shared control.

Microsoft Azure conducts real-time analysis of events within its operational environment and evaluates the results from systems that generate near real-time alerts about events that could potentially compromise the system. Azure is designed to enforce client segmentation through logical controls. The Azure logging and monitoring infrastructure encompass the entire Azure platform and does not vary by tenant. Azure is responsible for maintaining, collecting, storing, restricting access, and backing up log and audit trails. Microsoft Azure records and maintains a log of all individual user access to Microsoft Azure system components in the platform environment. Microsoft Azure platform components (including OS, CloudNet, Fabric, etc.) are configured to log and collect security events. Administrator activity in the Microsoft Azure platform is logged.

The customer is responsible for auditing individual user access. AAD reporting provides insights into user sign-in activities and system activity information about user and group management. The ARM provides data to the Azure Activity Log that provides the who, what, and when about resources in your subscription. The Azure Security Center also tracks activity.

10.2.1 – 10.2.7 These are shared controls.

This control for Microsoft Azure is covered by the policies outlined in control 10.1.

This control for the customer is covered by the policies outlined in control 10.2.1.

10.3.1 – 10.3.6 These are shared controls.

Microsoft Online Services will record and maintain a list of all individual user access to system components in the platform environment. Microsoft Online Services has established monitoring systems to detect audit processing failures and report to appropriate personnel. Audit logs are stored for a minimum of 180 days.

The customer is responsible for recording audit trail entries. This should include user IDs, event types, timestamps, success or failure of an event, origination of an event, and the asset name. With the built-in capabilities of Azure, Windows Server, SQL Server, and SQL Database, the deployed solutions capture audit records with enough detail to satisfy the requires of these controls.

10.4 – 10.4.3 Microsoft Azure covers these controls.

Microsoft Azure has established procedures to synchronize servers and network devices in the Microsoft Azure environment with NTP Stratum 1-time servers synchronized to Global Positioning System (GPS) satellites. Synchronization is performed automatically every five minutes. Microsoft Azure is responsible for ensuring service hosts properly sync time.

10.5 – 10.6.3 These are shared controls.

Microsoft protects facilities and log information against tampering and unauthorized access. Audit records are continually analyzed for indications of inappropriate or unusual activity using a formal monitoring process. Findings are reported using the security incident response process. Microsoft Online Services have formal monitoring processes to include frequency of review for Standard Operating Procedures and review oversight processes and procedures. Microsoft's presumed breach stance involves auditing all operator/administrator access and actions.

Logging of service, user and security events (web server logs, FTP server logs, etc.) is enabled and retained centrally. Azure restricts access to audit logs to authorized personnel based on job responsibilities. Event logs are archived on the Azure secure archival infrastructure and are retained for 180 days.

The customer is responsible for ensuring the accuracy and integrity of audit trails and well as reviewing and identifying any anomalies or suspicious activity. Azure provides a wide array of configurable security auditing and logging options to help you identify gaps in your security policies and mechanisms. RBAC should be configured to limit who can see and configure the audit logs for your subscription. See https://docs.microsoft.com/en-us/azure/security/azure-log-audit for more details.

10.7 This is a shared control.

Microsoft Azure retains audit logs for one year, with the most recent 3 months immediately accessible through their internal portal.

The customer is responsible for retaining audit trails for at least one year. The most recent three months must be immediately available for analysis. Retention options are built-in to Azure.

10.8 and 10.8.1 These are shared controls.

Microsoft Azure uses Exchange Web Services (EWS) to support real-time analysis of events within its operational environment. MAs and AIMS generate near real-time alerts about events that could potentially compromise the system.

Customers that are service providers are responsible for the timely detection and reporting of failures to critical systems. Customers can use PowerShell to retrieve log information or use a log aggregation service like Azure Log Analytics to report and generate near real-time alerts about events.

10.9 This is a customer-owned control.

The customer is responsible for documenting security policies and operational procedures for monitoring access to network resources and cardholder data. These policies and procedures should be distributed to all associated parties.

Summary: Microsoft Azure and the customer share a responsibility for auditing and monitoring all access to network resources and cardholder data. Fortunately for customers using Microsoft Azure, auditing and logging tools exist with numerous categories. Reporting from these logs has been made easy using Azure Log Analytics. This data can also be ingested into numerous other log aggregation tools that customers may already be using.

Requirement 11 – Regularly Test Security Systems and Processes

Requirement 11 states that customers must regularly test security systems and processes. Microsoft Azure can provide 41.2% reference architecture requirement coverage leveraging the Azure portal, Azure Security Center, App Service Environment, Application Gateway and Azure Networking including ExpressRoute and VPN Gateways.

This requirement is further broken down into the 17 controls listed below.

11.1, 11.1.1, and 11.1.2 Microsoft Azure covers this control.

This control is not applicable to the Microsoft Azure environment because wireless access is not permitted within the Microsoft Azure network environment.

11.2 This is a shared control.

Vulnerability scans are performed on a quarterly basis. Microsoft Azure contracts with independent assessors to perform penetration testing of the Microsoft Azure boundary.

Software updates are released through the monthly OS release cycle using change and release management procedures. Emergency out-of-band security software updates (0-day & Software Security Incident Response Process - SSIRP updates) are deployed as quickly as possible. If customers use the default "Auto Upgrade" option, software updates will be applied to their VMs automatically. Otherwise, customers have the option to upgrade to the latest OS image through

the management portal. In the case of a VM role, customers are responsible for evaluating and updating their VMs.

New vulnerabilities (e.g., those from responsible disclosure programs) are communicated to Azure through the Microsoft Security Response Center. If a patch is developed for the vulnerability, each service team evaluates the relevance of the patch to its environment and applies the patch as applicable.

The customer is responsible for performing internal and external vulnerability scans on at least a quarterly basis. Scans should also be performed after a significant change to the network. Many resources can be scanned internally using the Azure Vulnerability Assessment, which can be configured to scan weekly.

11.2.1 and 11.2.2. Microsoft Azure covers this control.

This control for Microsoft Azure is covered by the policies outlined in control 11.2.

The customer is responsible for performing internal and external vulnerability scans on at least a quarterly basis. This should include the resolution of all "high-risk" vulnerabilities. Scans should also be performed via a PCI SSC Approved Scanning Vendor (ASV).

11.2.3 This is a shared control.

This control for Microsoft Azure is covered by the policies outlined in control 11.2.

The customer is responsible for performing vulnerability scans on at least a quarterly basis. Scan should be performed by qualified personnel.

11.3 – 11.3.4 This is a shared control.

Microsoft Azure validates services using third party penetration testing based upon the OWASP (Open Web Application Security Project) top ten using CREST-certified testers. The results of testing are tracked through a risk register, which is audited and reviewed on a regular basis to ensure compliance to security practices.

Microsoft also uses Red Teaming against Microsoft-managed infrastructure, services and applications. No end-customer data is deliberately targeted during Red Teaming and live site penetration testing. The tests are against Microsoft Azure infrastructure and platforms as well as Microsoft's own applications and data. Customer tenants, applications and data hosted in Azure are never targeted.

Microsoft Azure has employed an independent assessor to develop a system assessment plan and conduct a controls assessment. Controls assessments are performed annually, and the

results are reported to relevant parties. Results are reported to stakeholders and remediation is tracked by the team through closure.

The customer is responsible for performing penetrations tests as per PCI DSS compliance guidelines. These internal and external penetration tests should be performed annually or after significant changes to the system. Any vulnerabilities discovered should be patched or mitigated in an appropriate manner.

11.3.4.1 This is a shared control.

Procedures have been established to monitor the Microsoft Azure platform components for known security vulnerabilities.

Each quarter targeted comprehensive security vulnerability scanning against prioritized components of the Azure production environment is performed to identify security vulnerabilities. Results are reported to stakeholders and remediation is tracked by the team through closure.

Customers that are service providers are responsible for performing penetration tests on segmentation controls at least every six months.

11.4 This is a shared control.

Microsoft Azure conducts real-time analysis of events within its operational environment and Intrusion Detection Systems (IDS) generate near real-time alerts about events that could potentially compromise the system. Microsoft security researchers are constantly on the lookout for threats. They have access to an expansive set of telemetry gained from Microsoft's global presence in the cloud and on-premises. This wide-reaching and diverse collection of datasets enables Microsoft to discover new attack patterns and trends across its on-premises consumer and enterprise products, as well as its online services. As a result, Security Center can rapidly update its detection algorithms as attackers release new and increasingly sophisticated exploits.

The customer is responsible for monitoring network traffic via logging and intrusion-detection systems. The Azure Security Center with Threat Detection can monitor and report on malicious activities.

11.5 This is a shared control.

Microsoft Azure maintains and notifies customers of potential changes and events that may impact security or availability of the services through an online Service Dashboard. Changes to the security commitments and security obligations of Microsoft Azure customers are updated on the Microsoft Azure website in a timely manner.

The customer is responsible for deploying change-detection mechanisms. Personnel should be alerted to unauthorized modifications of the file system. With the use of Log Analytics aggregating audit log files, the who, what, when can be determined and reported on.

11.5.1 This is a shared control.

Azure monitoring event rules provide an increased level of monitoring for high risk operations and assets. Azure-managed network devices are monitored for compliance with established security standards.

The customer is responsible for deploying change-detection mechanisms. This should include a response process to generated alerts.

11.6 This is a customer-owned control.

The customer is responsible for documenting security policies and operational procedures for security monitoring and testing. These policies and procedures should be distributed to all associated parties.

Summary: Microsoft Azure makes this requirement much easier to meet due to the infrastructure of the systems being covered. Customers should also enlist external companies to do routine vulnerability and penetration test. Most of the scans are required yearly, a lot can happen in a year. Scanning more often can alert you to vulnerabilities more quickly thus helping to minimize exposure.

Requirement 12 – Maintain a Policy that Addresses Information Security for all Personnel

Requirement 12 states that customers must maintain a policy that addresses information security for all personnel. This requirement is primarily a customer requirement since it addresses information security policy regarding personnel. Microsoft Azure can provide 12.8% reference architecture requirement coverage.

This requirement is further broken down into the 39 controls listed below.

12.1 This is a customer-owned control.

The customer is responsible for establishing, maintaining, and publishing security policies. These policies should be distributed to all associated parties.

12.1.1 This is a customer-owned control.

The customer is responsible for reviewing security policies at least annually. These policies should be updated after environment changes.

12.2 This is a customer-owned control.

The customer is responsible for implementing a risk-assessment process.

12.3.1 – 12.3.10 These are customer-owned controls.

The customer is responsible for developing policies dictating proper usage of critical technologies within their cardholder data environment. These policies should include:

- Authorization requirements
- Authentication requirements
- An inventory of devices and authorized personnel
- A method to accurately determine ownership
- Acceptable usage
- Acceptable network locations
- Company-approved products
- Inactivity periods for automatic session closure
- Activation and immediate deactivation of remote access for vendors
- Prohibit the copying, moving, and storage of cardholder data outside the cardholder data environment.

12.4 This is a customer-owned control.

The customer is responsible for ensuring their security policy and operating procedures define responsibilities for personnel.

12.4.1 This is a customer-owned control.

Customers that are service providers are responsible for implementing a PCI DSS compliance program.

12.5.1 – 12.5.5 These are customer-owned controls.

The customer is responsible for assigning security management responsibilities. This should include:

- The establishment and dissemination of security policy and procedures
- Monitoring of security alerts
- Establishment and dissemination of incident response and escalation procedures
- Administration of user accounts
- Monitoring and controlling access to data.

12.6, 12.6.1, and 12.6.2 This is a customer-owned control.

The customer is responsible for implementing a formal security awareness program. This program should focus on cardholder data security and be distributed to all associated parties at least annually and when someone is hired. Personnel should acknowledge current security policies and procedures at least annually.

12.7 This is a customer-owned control.

The customer is responsible for performing personnel screening prior to hiring.

12.8.1 – 12.9 These are shared controls.

Microsoft requires all third parties (external information system services) who are engaged with Azure to sign a Microsoft Server Security Assessment (MSSA). The MSSA is a detailed review of the configuration of Microsoft servers to ensure that critical systems are configured to minimize exposure and risk and thus maximize security. They require the third party to comply with all applicable Microsoft security policies and implement security procedures to prevent disclosure of Microsoft confidential information. Microsoft includes provisions in the MSSA and any associated Statements of Work with each vendor addressing the need to employ appropriate security controls. Vendors that handle sensitive data must be compliant with Microsoft vendor privacy practices and data protection requirements.

The customer is responsible for policies and procedures to manage service providers with whom cardholder data is shared. This should include a list of service providers and their provided services, written agreements acknowledging service provider responsibilities, a process for engaging service providers, a program to monitor service providers' PCI DSS compliance status, at least annually, and include identifying PCI DSS responsibilities of service providers.

Customers that are service providers are responsible for acknowledging in writing their responsibilities for maintaining PCI DSS compliance.

12.10 – 12.10.6 These are shared controls.

The customer is responsible for implementing an incident response plan. This plan should be reviewed and tested, at least annually, include the designation of specific response personnel, include security breach response training, include alerts from security monitoring systems, and be reviewed and modified appropriately, considering lessons learned.

12.11 This is a shared control.

Customers that are service providers are responsible for reviewing that established security policies and operational procedures are followed by personnel.

12.11.1 This is a shared control.

Customers that are service providers are responsible for documenting the quarterly review process and requiring PCI DSS compliance personnel to sign-off on the results.

Summary: Requirement 12 is instructing the customer to have established processes for maintaining security polices, risk-assessments, and polices for usage of various critical

technologies. Microsoft Azure shares a few responsibilities which are mitigated by all third parties who are engaged with Azure having to sign an MSSA that requires them to comply with Microsoft security policies.

Conclusion

Any company that processes, transmits, or stores cardholder data must adhere to the strict guidelines that make up the PCI DSS compliance requirements. This is a costly and time-consuming process that requires organizations to be doing many daily tasks. Mid-size and large-size organizations have had to dedicate full-time resources to help keep the organization PCI DSS compliant.

By using Microsoft PaaS solutions, many controls are owned fully by Microsoft Azure such as around physical access to the datacenter, disposal of hardware, patching firmware and operating systems, and many more. Many security controls are also shared by both Microsoft Azure and the customer, for instance, relating to performing penetration tests and code reviews, secure coding, maintaining system logs, and access to environments. There are also controls that are the sole responsibility of the customer, most of which consist of defining polices for controls and maintaining documentation around controls.

The offerings for security and data protection within the Azure PaaS platform are vast:

- Customers can leverage built-in security features for performing vulnerability assessments, take advantage of a current preview feature for data classification and discovery, various auditing tools, and leverage Threat Detection to look for anomalies in access to their environment.
- Customers should be leveraging AAD for user access and role-based memberships.
- For further securing access to data, customers should be taking advantage of Always Encrypted to encrypt sensitive data within the database in conjunction with TDE for encryption at rest. RLS allows customers to restrict access to sensitive data based on the users' access while DDM mask sensitive data in the result set.
- For network controls, Azure is equally full of features for securing the network. Users can define Virtual Networks, configure firewall rules and network security groups, virtual private networks, ExpressRoute, VPN Gateways, and more.
- The creation of resources can all be done using the Azure Portal using the ARM, through APIs, or PowerShell.

Any organization considering Microsoft Azure for their data assets containing cardholder data will have peace of mind knowing that Microsoft takes security extremely seriously and has made provided security features to help customers protect their data a top priority.

Customers leveraging Microsoft Azure for their environment can transfer a significant portion of the PCI DSS requirement responsibility to Microsoft, saving time and money and providing them with peace of mind.

Appendix: PCI DSS Compliance Guidelines

This section of the whitepaper lists all the PCI DSS requirements.

Requirement 1

This requirement is broken down into the 22 controls listed below.

1.1.1 Establish and implement firewall and router configuration standards that include the following:

- A formal process for approving and testing all network connections and changes to the firewall and router configurations.

1.1.2 Establish and implement firewall and router configuration standards that include the following:

- Current network diagram that identifies all connections between the cardholder data environment and other networks, including any wireless networks.

1.1.3 Establish and implement firewall and router configuration standards that include the following:

- Current diagram that shows all cardholder data flows across systems and networks.

1.1.4 Establish and implement firewall and router configuration standards that include the following:

- Requirements for a firewall at each Internet connection and between any DMZ and the internal network.

1.1.5 Establish and implement firewall and router configuration standards that include the following:

- Description of groups, roles, and responsibilities for management of network components.

1.1.6 Establish and implement firewall and router configuration standards that include the following:

- Documentation and business justification for use of all services, protocols, and ports allowed, including documentation of security features implemented for those protocols considered to be insecure.

1.1.7 Establish and implement firewall and router configuration standards that include the following:

- Requirement to review firewall and router rule sets at least every six months.

1.2 Build firewall and router configurations that restrict connections between untrusted networks and any system components in the cardholder data environment.
Note: An "untrusted network" is any network that is external to the networks belonging to the entity under review, and/or which is out of the entity's ability to control or manage.

1.2.1 Restrict inbound and outbound traffic to that which is necessary for the cardholder data environment, and specifically deny all other traffic.

1.2.2 Secure and synchronize router configuration files.

1.2.3 Install perimeter firewalls between all wireless networks and the cardholder data environment, and configure these firewalls to deny or, if traffic is necessary for business purposes, permit only authorized traffic between the wireless environment and the cardholder data environment.

1.3 Prohibit direct public access between the Internet and any system component in the cardholder data environment.

1.3.1 Implement a DMZ to limit inbound traffic to only system components that provide authorized publicly accessible services, protocols, and ports.

1.3.2 Limit inbound Internet traffic to IP addresses within the DMZ.

1.3.3 Implement anti-spoofing measures to detect and block forged source IP addresses from entering the network.

(For example, block traffic originating from the Internet with an internal source address.)

1.3.4 Do not allow unauthorized outbound traffic from the cardholder data environment to the Internet.

1.3.5 Permit only "established" connections into the network.

1.3.6 Place system components that store cardholder data (such as a database) in an internal network zone, segregated from the DMZ and other untrusted networks.

1.3.7 Do not disclose private IP addresses and routing information to unauthorized parties.
Note: Methods to obscure IP addressing may include, but are not limited to:

- Network Address Translation (NAT).
- Placing servers containing cardholder data behind proxy servers/firewalls.
- Removal or filtering of route advertisements for private networks that employ registered addressing.
- Internal use of RFC1918 address space instead of registered addresses.

1.4 Install personal firewall software on any mobile and/or employee-owned devices that connect to the Internet when outside the network (for example, laptops used by employees), and which are also used to access the network. Firewall configurations include:

- Specific configuration settings are defined for personal firewall software.
- Personal firewall software is actively running.
- Personal firewall software is not alterable by users of mobile and/or employee-owned devices.

1.5 Ensure that security policies and operational procedures for managing firewalls are documented, in use, and known to all affected parties.

Requirement 2

This requirement is broken down into the 12 controls listed below.

2.1 Always change vendor-supplied defaults and remove or disable unnecessary default accounts before installing a system on the network.

2.1.1 For wireless environments connected to the cardholder data environment or transmitting cardholder data, change ALL wireless vendor defaults at installation, including but not limited to default wireless encryption keys, passwords, and SNMP community strings.

2.2 Develop configuration standards for all system components. Assure that these standards address all known security vulnerabilities and are consistent with industry-accepted system hardening standards.

2.2.1 Implement only one primary function per server to prevent functions that require different security levels from co-existing on the same server. (For example, web servers, database servers, and DNS should be implemented on separate servers.)

2.2.2 Enable only necessary services, protocols, daemons, etc., as required for the function of the system.

2.2.3 Implement additional security features for any required services, protocols, or daemons that are considered to be insecure. Note: Where TLS/early TLS is used, the requirements in Appendix A2 of the PCI DSS must be completed.

2.2.4 Configure system security parameters to prevent misuse.

2.2.5 Remove all unnecessary functionality, such as scripts, drivers, features, subsystems, file systems, and unnecessary web servers.

2.3 Encrypt all non-console administrative access using strong cryptography.

2.4 Maintain an inventory of system components that are in scope for PCI DSS.

2.5 Ensure that security policies and operational procedures for managing vendor defaults and other security parameters are documented, in use, and known to all affected parties.

2.6 Shared hosting providers must protect each entity's hosted environment and cardholder data. These providers must meet specific requirements as detailed in Appendix A of the PCI DSS.

Requirement 3

This requirement is broken down into the 22 controls listed below.

3.1 Keep cardholder data storage to a minimum by implementing data retention and disposal policies, procedures and processes that include at least the following for all cardholder data (CHD) storage:

- Limiting data storage amount and retention time to that which is required for legal, regulatory, and business requirements.
- Processes for secure deletion of data when no longer needed.
- Specific retention requirements for cardholder data.
- A quarterly process for identifying and securely deleting stored cardholder data that exceeds defined retention.

3.2 Do not store sensitive authentication data after authorization (even if encrypted). If sensitive authentication data is received, render all data unrecoverable upon completion of the authorization process. It is permissible for issuers and companies that support issuing services to store sensitive authentication data if there is a business justification and the data is stored securely. Sensitive authentication data includes the data as cited in the following Requirements 3.2.1 through 3.2.3.

3.2.1 Do not store the full contents of any track (from the magnetic stripe located on the back of a card, equivalent data contained on a chip, or elsewhere) after authorization. This data is alternatively called full track, track, track 1, track 2, and magnetic-stripe data.

Note: In the normal course of business, the following data elements from the magnetic stripe may need to be retained: the cardholder's name, primary account number (PAN), expiration date, and/or service code. To minimize risk, store only these data elements as needed for business.

3.2.2 Do not store the card verification code or value (three-digit or four-digit number printed on the front or back of the payment card) used to verify card-not-present transactions after authorization.

3.2.3 Do not store the personal identification number (PIN) or the encrypted PIN block after authorization.

3.3 Mask PAN when displayed (the first six and last four digits are the maximum number of digits to be displayed), such that only personnel with a legitimate business need can see the full PAN.

Note: This requirement does not supersede stricter requirements in place for displays of cardholder data – for example, legal or payment card brand requirements for point-of-sale (POS) receipts.

3.4 Render PAN unreadable anywhere it is stored (including on portable digital media, backup media, and in logs) by using any of the following approaches: one-way hashes based on strong cryptography, (hash must be of the entire PAN), truncation (hashing cannot be used to replace the truncated segment of PAN), index tokens and pads (pads must be securely stored), strong cryptography with associated key-management processes and procedures.

Note: It is a relatively trivial effort for a malicious individual to reconstruct original PAN data if they have access to both the truncated and hashed version of a PAN. Where hashed and truncated versions of the same PAN are present in an entity's environment, additional controls must be in place to ensure that the hashed and truncated versions cannot be correlated to reconstruct the original PAN.

3.4.1 If disk encryption is used (rather than file or column-level database encryption), logical access must be managed separately and independently of native operating system authentication and access control mechanisms (for example, by not using local user account databases or general network login credentials). Decryption keys must not be associated with user accounts.

3.5 Document and implement procedures to protect keys used to secure stored cardholder data against disclosure and misuse:

Note: This requirement applies to keys used to encrypt stored cardholder data, and applies to key-encrypting keys used to protect data-encrypting keys—such key-encrypting keys must be at least as strong as the data-encrypting key.

3.5.1 Additional requirement for service providers only: Maintain a documented description of the cryptographic architecture that includes:

- Details of all algorithms, protocols, and keys used for the protection of cardholder data, including key strength and expiry date.
- Description of the key usage for each key.
- Inventory of any HSMs and other SCDs used for key management.

3.5.2 Restrict access to cryptographic keys to the fewest number of custodians necessary.

3.5.3 Store secret and private keys used to encrypt/decrypt cardholder data in one (or more) of the following forms at all times:

- Encrypted with a key-encrypting key that is at least as strong as the data-encrypting key, and that is stored separately from the data-encrypting key.
- Within a secure cryptographic device (such as a hardware (host) security module (HSM) or PTS-approved point-of-interaction device).
- As at least two full-length key components or key shares, in accordance with an industry-accepted method.

Note: It is not required that public keys be stored in one of these forms.

3.5.4 Store cryptographic keys in the fewest possible locations.

3.6.1, 3.6.2, and 3.6.3 Fully document and implement all key-management processes and procedures for cryptographic keys used for encryption of cardholder data, including the generation of strong cryptographic keys.

Note: Numerous industry standards for key management are available from various resources including NIST, which can be found at http://csrc.nist.gov.

3.6.4 Fully document and implement all key-management processes and procedures for cryptographic keys used for encryption of cardholder data, including the following:

Cryptographic key changes for keys that have reached the end of their crypto period (for example, after a defined period of time has passed and/or after a certain amount of cipher-text has been produced by a given key), as defined by the associated application vendor or key owner, and based on industry best practices and guidelines (for example, NIST Special Publication 800-57).

Note: Numerous industry standards for key management are available from various resources including NIST, which can be found at http://csrc.nist.gov.

3.6.5 Fully document and implement all key-management processes and procedures for cryptographic keys used for encryption of cardholder data, including the following:
Retirement or replacement (for example, archiving, destruction, and/or revocation) of keys as deemed necessary when the integrity of the key has been weakened (for example, departure of an employee with knowledge of a clear-text key component), or keys are suspected of being compromised.

Note: If retired or replaced cryptographic keys need to be retained, these keys must be securely archived (for example, by using a key-encryption key). Archived cryptographic keys should only be used for decryption/verification purposes.

Note: Numerous industry standards for key management are available from various resources including NIST, which can be found at http://csrc.nist.gov.

3.6.6 Fully document and implement all key-management processes and procedures for cryptographic keys used for encryption of cardholder data, including the following:

If manual clear-text cryptographic key-management operations are used, these operations must be managed using split knowledge and dual control.

Note: Examples of manual key-management operations include, but are not limited to key generation, transmission, loading, storage and destruction. Numerous industry standards for key management are available from various resources including NIST, which can be found at http://csrc.nist.gov.

3.6.7 Fully document and implement all key-management processes and procedures for cryptographic keys used for encryption of cardholder data, including the following:

Prevention of unauthorized substitution of cryptographic keys.

Note: Numerous industry standards for key management are available from various resources including NIST, which can be found at http://csrc.nist.gov.

3.6.8 Fully document and implement all key-management processes and procedures for cryptographic keys used for encryption of cardholder data, including the following:

Requirement for cryptographic key custodians to formally acknowledge that they understand and accept their key-custodian responsibilities.

Note: Numerous industry standards for key management are available from various resources including NIST, which can be found at http://csrc.nist.gov.

3.7 Ensure that security policies and operational procedures for protecting stored cardholder data are documented, in use, and known to all affected parties

Requirement 4

This requirement is defined by the 4 controls listed below.

4.1 Use strong cryptography and security protocols (for example, TLS, IPSEC, SSH, etc.) to safeguard sensitive cardholder data during transmission over open, public networks, including the following:

- Only trusted keys and certificates are accepted.
- The protocol in use only supports secure versions or configurations.
- The encryption strength is appropriate for the encryption methodology in use.

Note: Where TLS/early TLS is used, the requirements in Appendix A2 of the PCI DSS must be completed.

Examples of open, public networks include but are not limited to:

- The Internet.
- Wireless technologies, including 802.11 and Bluetooth.

- Cellular technologies, for example, Global System for Mobile communications (GSM), Code division multiple access (CDMA).
- General Packet Radio Service (GPRS).
- Satellite communications.

4.1.1 Ensure wireless networks transmitting cardholder data or connected to the cardholder data environment, use industry best practices (for example, IEEE 802.11i) to implement strong encryption for authentication and transmission.

4.2 Never send unprotected PANs by end-user messaging technologies (for example, e-mail, instant messaging, SMS, chat, etc.).

4.3 Ensure that security policies and operational procedures for encrypting transmissions of cardholder data are documented, in use, and known to all affected parties.

Requirement 5

This requirement is broken down into the 6 controls listed below.

5.1 Deploy anti-virus software on all systems commonly affected by malicious software (particularly personal computers and servers).

5.1.1 Ensure that anti-virus programs are capable of detecting, removing, and protecting against all known types of malicious software.

5.1.2 For systems considered to be not commonly affected by malicious software, perform periodic evaluations to identify and evaluate evolving malware threats in order to confirm whether such systems continue to not require anti-virus software.

5.2 Ensure that all anti-virus mechanisms are maintained as follows:

- Are kept current.
- Perform periodic scans.
- Generate audit logs which are retained per PCI DSS Requirement 10.7.

5.3 Ensure that anti-virus mechanisms are actively running and cannot be disabled or altered by users, unless specifically authorized by management on a case-by-case basis for a limited time period.

Note: Anti-virus solutions may be temporarily disabled only if there is legitimate technical need, as authorized by management on a case-by-case basis. If anti-virus protection needs to be disabled for a specific purpose, it must be formally authorized. Additional security measures may also need to be implemented for the period during which anti-virus protection is not active.

5.4 Ensure that security policies and operational procedures for protecting systems against malware are documented, in use, and known to all affected parties.

Requirement 6

This requirement is broken down into the 26 controls listed below.

6.1 Establish a process to identify security vulnerabilities, using reputable outside sources for security vulnerability information, and assign a risk ranking (for example, as "high," "medium," or "low") to newly discovered security vulnerabilities.

Note: Risk rankings should be based on industry best practices as well as consideration of potential impact. For example, criteria for ranking vulnerabilities may include consideration of the CVSS base score, and/or the classification by the vendor, and/or type of systems affected. Methods for evaluating vulnerabilities and assigning risk ratings will vary based on an organization's environment and risk-assessment strategy. Risk rankings should, at a minimum, identify all vulnerabilities considered to be a "high risk" to the environment. In addition to the risk ranking, vulnerabilities may be considered "critical" if they pose an imminent threat to the environment, impact critical systems, and/or would result in a potential compromise if not addressed. Examples of critical systems may include security systems, public-facing devices and systems, databases, and other systems that store, process, or transmit cardholder data.

6.2 Ensure that all system components and software are protected from known vulnerabilities by installing applicable vendor-supplied security patches. Install critical security patches within one month of release.

Note: Critical security patches should be identified according to the risk ranking process defined in Requirement 6.1.

6.3 Develop internal and external software applications (including web-based administrative access to applications) securely, as follows:

- In accordance with PCI DSS (for example, secure authentication and logging).
- Based on industry standards and/or best practices.
- Incorporating information security throughout the software-development life cycle.

Note: this applies to all software developed internally as well as bespoke or custom software developed by a third party.

6.3.1 Remove development, test and/or custom application accounts, user IDs, and passwords before applications become active or are released to customers.

6.3.2 Review custom code prior to release to production or customers in order to identify any potential coding vulnerability (using either manual or automated processes) to include at least the following:

- Code changes are reviewed by individuals other than the originating code author, and by individuals knowledgeable about code-review techniques and secure coding practices. Code reviews ensure code is developed according to secure coding guidelines

- Appropriate corrections are implemented prior to release.
- Code-review results are reviewed and approved by management prior to release.

Note: This requirement for code reviews applies to all custom code (both internal and public-facing), as part of the system development life cycle. Code reviews can be conducted by knowledgeable internal personnel or third parties. Public-facing web applications are also subject to additional controls, to address ongoing threats and vulnerabilities after implementation, as defined at PCI DSS Requirement

6.4.1 Follow change control processes and procedures for all changes to system components. The processes must include the following:

- Separate development/test environments from production environments and enforce the separation with access controls.

6.4.2 Follow change control processes and procedures for all changes to system components. The processes must include the following:

- Separation of duties between development/test and production environments.

6.4.3 Follow change control processes and procedures for all changes to system components. The processes must include the following:

- Production data (live PANs) are not used for testing or development.

6.4.4 Follow change control processes and procedures for all changes to system components. The processes must include the following:

- Removal of test data and accounts before production systems become active.

6.4.5.1 Change control procedures for the implementation of security patches and software modifications must include the following:

- Documentation of impact.

6.4.5.2 Change control procedures for the implementation of security patches and software modifications must include the following:

- Documented change approval by authorized parties.

6.4.5.3 Change control procedures for the implementation of security patches and software modifications must include the following:

- Functionality testing to verify that the change does not adversely impact the security of the system.

6.4.5.4 Change control procedures for the implementation of security patches and software modifications must include the following:

- Back-out procedures.

6.4.6 Follow change control processes and procedures for all changes to system components. The processes must include the following:

- Upon completion of a significant change, all relevant PCI DSS requirements must be implemented on all new or changed systems and networks, and documentation updated as applicable.

6.5.1 Address common coding vulnerabilities in software-development processes as follows:

- Train developers in secure coding techniques, including how to avoid common coding vulnerabilities, and understanding how sensitive data is handled in memory.
- Develop applications based on secure coding guidelines.

Vulnerabilities include:

- Injection flaws, particularly SQL injection. Also consider OS Command Injection, LDAP and XPath injection flaws as well as other injection flaws.

Note: The vulnerabilities listed at 6.5.1 through 6.5.10 were current with industry best practices when this version of PCI DSS was published. However, as industry best practices for vulnerability management are updated (for example, the OWASP Guide, SANS CWE Top 25, CERT Secure Coding, etc.), the current best practices must be used for these requirements.

6.5.2 Address common coding vulnerabilities in software-development processes as follows:

- Train developers in secure coding techniques, including how to avoid common coding vulnerabilities, and understanding how sensitive data is handled in memory.
- Develop applications based on secure coding guidelines.

Vulnerabilities include:

- Buffer overflows.

Note: The vulnerabilities listed at 6.5.1 through 6.5.10 were current with industry best practices when this version of PCI DSS was published. However, as industry best practices for vulnerability management are updated (for example, the OWASP Guide, SANS CWE Top 25, CERT Secure Coding, etc.), the current best practices must be used for these requirements.

6.5.3 Address common coding vulnerabilities in software-development processes as follows:

- Train developers in secure coding techniques, including how to avoid common coding vulnerabilities, and understanding how sensitive data is handled in memory.
- Develop applications based on secure coding guidelines.

Vulnerabilities include:

- Insecure cryptographic storage.

Note: The vulnerabilities listed at 6.5.1 through 6.5.10 were current with industry best practices when this version of PCI DSS was published. However, as industry best practices for vulnerability management are updated (for example, the OWASP Guide, SANS CWE Top 25, CERT Secure Coding, etc.), the current best practices must be used for these requirements.

6.5.4 Address common coding vulnerabilities in software-development processes as follows:

- Train developers in secure coding techniques, including how to avoid common coding vulnerabilities, and understanding how sensitive data is handled in memory.
- Develop applications based on secure coding guidelines.

Vulnerabilities include:

- Insecure communications.

Note: The vulnerabilities listed at 6.5.1 through 6.5.10 were current with industry best practices when this version of PCI DSS was published. However, as industry best practices for vulnerability management are updated (for example, the OWASP Guide, SANS CWE Top 25, CERT Secure Coding, etc.), the current best practices must be used for these requirements.

6.5.5 Address common coding vulnerabilities in software-development processes as follows:

- Train developers in secure coding techniques, including how to avoid common coding vulnerabilities, and understanding how sensitive data is handled in memory.
- Develop applications based on secure coding guidelines.

Vulnerabilities include:

- Improper error handling.

Note: The vulnerabilities listed at 6.5.1 through 6.5.10 were current with industry best practices when this version of PCI DSS was published. However, as industry best practices for vulnerability management are updated (for example, the OWASP Guide, SANS CWE Top 25, CERT Secure Coding, etc.), the current best practices must be used for these requirements.

6.5.6 Address common coding vulnerabilities in software-development processes as follows:

- Train developers in secure coding techniques, including how to avoid common coding vulnerabilities, and understanding how sensitive data is handled in memory.
- Develop applications based on secure coding guidelines.

Vulnerabilities include:

- All “high risk” vulnerabilities identified in the vulnerability identification process (as defined in PCI DSS Requirement 6.1).

Note: The vulnerabilities listed at 6.5.1 through 6.5.10 were current with industry best practices when this version of PCI DSS was published. However, as industry best practices for vulnerability

management are updated (for example, the OWASP Guide, SANS CWE Top 25, CERT Secure Coding, etc.), the current best practices must be used for these requirements.

6.5.7 Address common coding vulnerabilities in software-development processes as follows:

- Train developers in secure coding techniques, including how to avoid common coding vulnerabilities, and understanding how sensitive data is handled in memory.
- Develop applications based on secure coding guidelines.

Vulnerabilities include:

- Cross-site scripting (XSS).

Note: The vulnerabilities listed at 6.5.1 through 6.5.10 were current with industry best practices when this version of PCI DSS was published. However, as industry best practices for vulnerability management are updated (for example, the OWASP Guide, SANS CWE Top 25, CERT Secure Coding, etc.), the current best practices must be used for these requirements.

6.5.8 Address common coding vulnerabilities in software-development processes as follows:

- Train developers in secure coding techniques, including how to avoid common coding vulnerabilities, and understanding how sensitive data is handled in memory.
- Develop applications based on secure coding guidelines.

Vulnerabilities include:

- Improper access control (such as insecure direct object references, failure to restrict URL access, directory traversal, and failure to restrict user access to functions).

Note: The vulnerabilities listed at 6.5.1 through 6.5.10 were current with industry best practices when this version of PCI DSS was published. However, as industry best practices for vulnerability management are updated (for example, the OWASP Guide, SANS CWE Top 25, CERT Secure Coding, etc.), the current best practices must be used for these requirements.

6.5.9 Address common coding vulnerabilities in software-development processes as follows:

- Train developers in secure coding techniques, including how to avoid common coding vulnerabilities, and understanding how sensitive data is handled in memory.
- Develop applications based on secure coding guidelines.

Vulnerabilities include:

- Cross-site request forgery (CSRF).

Note: The vulnerabilities listed at 6.5.1 through 6.5.10 were current with industry best practices when this version of PCI DSS was published. However, as industry best practices for vulnerability management are updated (for example, the OWASP Guide, SANS CWE Top 25, CERT Secure Coding, etc.), the current best practices must be used for these requirements.

6.5.10 Address common coding vulnerabilities in software-development processes as follows:

- Train developers in secure coding techniques, including how to avoid common coding vulnerabilities, and understanding how sensitive data is handled in memory.
- Develop applications based on secure coding guidelines.

Vulnerabilities include:

- Broken authentication and session management.

Note: The vulnerabilities listed at 6.5.1 through 6.5.10 were current with industry best practices when this version of PCI DSS was published. However, as industry best practices for vulnerability management are updated (for example, the OWASP Guide, SANS CWE Top 25, CERT Secure Coding, etc.), the current best practices must be used for these requirements.

6.6 For public-facing web applications, address new threats and vulnerabilities on an ongoing basis and ensure these applications are protected against known attacks by either of the following methods:

- Reviewing public-facing web applications via manual or automated application vulnerability security assessment tools or methods, at least annually and after any changes Note: This assessment is not the same as the vulnerability scans performed for Requirement 11.2.
- Installing an automated technical solution that detects and prevents web-based attacks (for example, a web-application firewall) in front of public-facing web applications, to continually check all traffic.

6.7 Ensure that security policies and operational procedures for developing and maintaining secure systems and applications are documented, in use, and known to all affected parties.

Requirement 7

This requirement is broken down into the 9 controls listed below.

7.1 Limit access to system components and cardholder data to only those individuals whose job requires such access.

7.1.1 Define access needs for each role, including:

- System components and data resources that each role needs to access for their job function.
- Level of privilege required (for example, user, administrator, etc.) for accessing resources.

7.1.2 Restrict access to privileged user IDs to least privileges necessary to perform job responsibilities.

7.1.3 Assign access based on individual personnel's job classification and function.

7.1.4 Require documented approval by authorized parties specifying required privileges.

7.2.1 Establish an access control system for systems components that restricts access based on a user's need to know and is set to "deny all" unless specifically allowed.
This access control system must include the following:

- Coverage of all system components.

7.2.2 Establish an access control system for systems components that restricts access based on a user's need to know, and is set to "deny all" unless specifically allowed.
This access control system must include the following:

- Assignment of privileges to individuals based on job classification and function.

7.2.3 Establish an access control system for systems components that restricts access based on a user's need to know and is set to "deny all" unless specifically allowed.
This access control system must include the following:

- Default "deny-all" setting.

7.3 Ensure that security policies and operational procedures for restricting access to cardholder data are documented, in use, and known to all affected parties.

Requirement 8

This requirement is broken down into the 24 controls listed below.

8.1.1 Define and implement policies and procedures to ensure proper user identification management for non-consumer users and administrators on all system components as follows:

- Assign all users a unique ID before allowing them to access system components or cardholder data.

8.1.2 Define and implement policies and procedures to ensure proper user identification management for non-consumer users and administrators on all system components as follows:

- Control addition, deletion, and modification of user IDs, credentials, and other identifier objects.

8.1.3 Define and implement policies and procedures to ensure proper user identification management for non-consumer users and administrators on all system components as follows:

- Immediately revoke access for any terminated users.

8.1.4 Define and implement policies and procedures to ensure proper user identification management for non-consumer users and administrators on all system components as follows:

- Remove/disable inactive user accounts within 90 days.

8.1.5 Define and implement policies and procedures to ensure proper user identification management for non-consumer users and administrators on all system components as follows:

- Manage IDs used by vendors to access, support, or maintain system components via remote access as follows:
    - Enabled only during the time period needed and disabled when not in use.
    - Monitored when in use.

8.1.6 Define and implement policies and procedures to ensure proper user identification management for non-consumer users and administrators on all system components as follows:

- Limit repeated access attempts by locking out the user ID after not more than six attempts.

8.1.7 Define and implement policies and procedures to ensure proper user identification management for non-consumer users and administrators on all system components as follows:

- Set the lockout duration to a minimum of 30 minutes or until an administrator enables the user ID.

8.1.8 Define and implement policies and procedures to ensure proper user identification management for non-consumer users and administrators on all system components as follows:

- If a session has been idle for more than 15 minutes, require the user to re-authenticate to re-activate the terminal or session.

8.2 In addition to assigning a unique ID, ensure proper user-authentication management for non-consumer users and administrators on all system components by employing at least one of the following methods to authenticate all users:

- Something you know, such as a password or passphrase.
- Something you have, such as a token device or smart card.
- Something you are, such as a biometric.

8.2.1 Using strong cryptography, render all authentication credentials (such as passwords/phrases) unreadable during transmission and storage on all system components.

8.2.2 Verify user identity before modifying any authentication credential—for example, performing password resets, provisioning new tokens, or generating new keys.

8.2.3 Passwords/phrases must meet the following:

- Require a minimum length of at least seven characters.
- Contain both numeric and alphabetic characters.

Alternatively, the passwords/phrases must have complexity and strength at least equivalent to the parameters specified above.

8.2.4 Change user passwords/passphrases at least once every 90 days.

8.2.5 Do not allow an individual to submit a new password/phrase that is the same as any of the last four passwords/phrases he or she has used.

8.2.6 Set passwords/phrases for first-time use and upon reset to a unique value for each user and change immediately after the first use.

8.3 Secure all individual non-console administrative access and all remote access to the CDE using multi-factor authentication.

Note: Multi-factor authentication requires that a minimum of two of the three authentication methods (see Requirement 8.2 for descriptions of authentication methods) be used for authentication. Using one factor twice (for example, using two separate passwords) is not considered multi-factor authentication.

8.3.1 Incorporate multi-factor authentication for all non-console access into the CDE for personnel with administrative access.

8.3.2 Incorporate multi-factor authentication for all remote network access (both user and administrator and including third-party access for support or maintenance) originating from outside the entity's network.

8.4 Document and communicate authentication procedures and policies to all users including:

- Guidance on selecting strong authentication credentials.
- Guidance for how users should protect their authentication credentials.
- Instructions not to reuse previously used passwords.
- Instructions to change passwords if there is any suspicion the password could be compromised.

8.5 Do not use group, shared, or generic IDs, passwords, or other authentication methods as follows:

- Generic user IDs are disabled or removed.
- Shared user IDs do not exist for system administration and other critical functions.
- Shared and generic user IDs are not used to administer any system components.

8.5.1 Additional requirement for service providers only: Service providers with remote access to customer premises (for example, for support of POS systems or servers) must use a unique authentication credential (such as a password/phrase) for each customer.

Note: This requirement is not intended to apply to shared hosting providers accessing their own hosting environment, where multiple customer environments are hosted.

8.6 Where other authentication mechanisms are used (for example, physical or logical security tokens, smart cards, certificates, etc.), use of these mechanisms must be assigned as follows:

- Authentication mechanisms must be assigned to an individual account and not shared among multiple accounts.
- Physical and/or logical controls must be in place to ensure only the intended account can use that mechanism to gain access.

8.7 All access to any database containing cardholder data (including access by applications, administrators, and all other users) is restricted as follows:

- All user access to, user queries of, and user actions on databases are through programmatic methods.
- Only database administrators have the ability to directly access or query databases.
- Application IDs for database applications can only be used by the applications (and not by individual users or other non-application processes).

8.8 Ensure that security policies and operational procedures for identification and authentication are documented, in use, and known to all affected parties.

Requirement 9

This requirement is broken down into the 26 controls listed below.

9.1 Use appropriate facility entry controls to limit and monitor physical access to systems in the cardholder data environment.

9.1.1 Use video cameras and/or access control mechanisms to monitor individual physical access to sensitive areas. Review collected data and correlate with other entries. Store for at least three months, unless otherwise restricted by law.

Note: "Sensitive areas" refers to any data center, server room or any area that houses systems that store, process, or transmit cardholder data. This excludes public-facing areas where only point-of-sale terminals are present, such as the cashier areas in a retail store.

9.1.2 Implement physical and/or logical controls to restrict access to publicly accessible network jacks.

For example, network jacks located in public areas and areas accessible to visitors could be disabled and only enabled when network access is explicitly authorized. Alternatively, processes could be implemented to ensure that visitors are escorted at all times in areas with active network jacks.

9.1.3 Restrict physical access to wireless access points, gateways, handheld devices, networking/communications hardware, and telecommunication lines.

9.2 Develop procedures to easily distinguish between onsite personnel and visitors, to include:

- Identifying onsite personnel and visitors (for example, assigning badges).

- Changes to access requirements.
- Revoking or terminating onsite personnel and expired visitor identification (such as ID badges).

9.3 Control physical access for onsite personnel to the sensitive areas as follows:

- Access must be authorized and based on individual job function.
- Access is revoked immediately upon termination, and all physical access mechanisms, such as keys, access cards, etc., are returned or disabled.

9.4 Implement procedures to identify and authorize visitors.

9.4.1 Visitors are authorized before entering, and escorted at all times within, areas where cardholder data is processed or maintained.

9.4.2 Visitors are identified and given a badge or other identification that expires and that visibly distinguishes the visitors from onsite personnel.

9.4.3 Visitors are asked to surrender the badge or identification before leaving the facility or at the date of expiration.

9.4.4 A visitor log is used to maintain a physical audit trail of visitor activity to the facility as well as computer rooms and data centers where cardholder data is stored or transmitted. Document the visitor's name, the firm represented, and the onsite personnel authorizing physical access on the log. Retain this log for a minimum of three months, unless otherwise restricted by law.

9.5 Physically secure all media.

9.5.1 Store media backups in a secure location, preferably an off-site facility, such as an alternate or backup site, or a commercial storage facility. Review the location's security at least annually.

9.6.1 Maintain strict control over the internal or external distribution of any kind of media, including the following:

- Classify media so the sensitivity of the data can be determined.

9.6.2 Maintain strict control over the internal or external distribution of any kind of media, including the following:

- Send the media by secured courier or other delivery method that can be accurately tracked.

9.6.3 Maintain strict control over the internal or external distribution of any kind of media, including the following:

- Ensure management approves any and all media that is moved from a secured area (including when media is distributed to individuals).

9.7 Maintain strict control over the storage and accessibility of media.

9.7.1 Properly maintain inventory logs of all media and conduct media inventories at least annually.

9.8 Destroy media when it is no longer needed for business or legal reasons.

9.8.1 Shred, incinerate, or pulp hard-copy materials so that cardholder data cannot be reconstructed. Secure storage containers used for materials that are to be destroyed.

9.8.2 Render cardholder data on electronic media unrecoverable so that cardholder data cannot be reconstructed.

9.9 Protect devices that capture payment card data via direct physical interaction with the card from tampering and substitution.

Note: These requirements apply to card-reading devices used in card-present transactions (that is, card swipe or dip) at the point of sale. This requirement is not intended to apply to manual key-entry components such as computer keyboards and POS keypads.

9.9.1 Maintain an up-to-date list of devices. The list should include the following:

- Make, model of device
- Location of device (for example, the address of the site or facility where the device is located)
- Device serial number or other method of unique identification.

9.9.2 Periodically inspect device surfaces to detect tampering (for example, addition of card skimmers to devices), or substitution (for example, by checking the serial number or other device characteristics to verify it has not been swapped with a fraudulent device).

Note: Examples of signs that a device might have been tampered with or substituted include unexpected attachments or cables plugged into the device, missing or changed security labels, broken or differently colored casing, or changes to the serial number or other external markings.

9.9.3 Provide training for personnel to be aware of attempted tampering or replacement of devices. Training should include the following:

- Verify the identity of any third-party persons claiming to be repair or maintenance personnel, prior to granting them access to modify or troubleshoot devices.
- Do not install, replace, or return devices without verification.
- Be aware of suspicious behavior around devices (for example, attempts by unknown persons to unplug or open devices).

- Report suspicious behavior and indications of device tampering or substitution to appropriate personnel (for example, to a manager or security officer).

9.10 Ensure that security policies and operational procedures for restricting physical access to cardholder data are documented, in use, and known to all affected parties.

Requirement 10

This requirement is broken down into the 32 controls listed below.

10.1 Implement audit trails to link all access to system components to each individual user.

10.2.1 Implement automated audit trails for all system components to reconstruct the following events:

- All individual user accesses to cardholder data.

10.2.2 Implement automated audit trails for all system components to reconstruct the following events:

- All actions taken by any individual with root or administrative privileges.

10.2.3 Implement automated audit trails for all system components to reconstruct the following events:

- Access to all audit trails.

10.2.4 Implement automated audit trails for all system components to reconstruct the following events:

- Invalid logical access attempts.

10.2.5 Implement automated audit trails for all system components to reconstruct the following events:

- Use of and changes to identification and authentication mechanisms—including but not limited to creation of new accounts and elevation of privileges—and all changes, additions, or deletions to accounts with root or administrative privileges.

10.2.6 Implement automated audit trails for all system components to reconstruct the following events:

- Initialization, stopping, or pausing of the audit logs.

10.2.7 Implement automated audit trails for all system components to reconstruct the following events:

- Creation and deletion of system-level objects.

10.3.1 Record at least the following audit trail entries for all system components for each event:

- User identification.

10.3.2 Record at least the following audit trail entries for all system components for each event:

- Type of event.

10.3.3 Record at least the following audit trail entries for all system components for each event:

- Date and time.

10.3.4 Record at least the following audit trail entries for all system components for each event:

- Success or failure indication.

10.3.5 Record at least the following audit trail entries for all system components for each event:

- Origination of event.

10.3.6 Record at least the following audit trail entries for all system components for each event:

- Identity or name of affected data, system component, or resource.

10.4 Using time-synchronization technology, synchronize all critical system clocks and times and ensure that the following is implemented for acquiring, distributing, and storing time.

Note: One example of time synchronization technology is Network Time Protocol (NTP).

10.4.1 Critical systems have the correct and consistent time.

10.4.2 Time data is protected.

10.4.3 Time settings are received from industry-accepted time sources.

10.5 Secure audit trails so they cannot be altered.

10.5.1 Limit viewing of audit trails to those with a job-related need.

10.5.2 Protect audit trail files from unauthorized modifications.

10.5.3 Promptly back up audit trail files to a centralized log server or media that is difficult to alter.

10.5.4 Write logs for external-facing technologies onto a secure, centralized, internal log server or media device.

10.5.5 Use file-integrity monitoring or change-detection software on logs to ensure that existing log data cannot be changed without generating alerts (although new data being added should not cause an alert).

10.6 Review logs and security events for all system components to identify anomalies or suspicious activity.

Note: Log harvesting, parsing, and alerting tools may be used to meet this Requirement.

10.6.1 Review the following at least daily:

- All security events.
- Logs of all system components that store, process, or transmit CHD and/or SAD.
- Logs of all critical system components.
- Logs of all servers and system components that perform security functions (for example, firewalls, intrusion-detection systems/intrusion-prevention systems (IDS/IPS), authentication servers, e-commerce redirection servers, etc.).

10.6.2 Review logs of all other system components periodically based on the organization's policies and risk management strategy, as determined by the organization's annual risk assessment.

10.6.3 Follow up exceptions and anomalies identified during the review process.

10.7 Retain audit trail history for at least one year, with a minimum of three months immediately available for analysis (for example, online, archived, or restorable from backup).

10.8 Additional requirement for service providers only: Implement a process for the timely detection and reporting of failures of critical security control systems, including but not limited to failure of:

- Firewalls.
- IDS/IPS.
- FIM.
- Anti-virus.
- Physical access controls.
- Logical access controls.
- Audit logging mechanisms.
- Segmentation controls (if used).

10.8.1 Additional requirement for service providers only: Respond to failures of any critical security controls in a timely manner. Processes for responding to failures in security controls must include:

- Restoring security functions
- Identifying and documenting the duration (date and time start to end) of the security failure
- Identifying and documenting cause(s) of failure, including root cause, and documenting remediation required to address root cause

- Identifying and addressing any security issues that arose during the failure
- Performing a risk assessment to determine whether further actions are required as a result of the security failure
- Implementing controls to prevent cause of failure from reoccurring
- Resuming monitoring of security controls

10.9 Ensure that security policies and operational procedures for monitoring all access to network resources and cardholder data are documented, in use, and known to all affected parties.

Requirement 11

This requirement is broken down into the 17 controls listed below.

11.1 Implement processes to test for the presence of wireless access points (802.11), and detect and identify all authorized and unauthorized wireless access points on a quarterly basis.

Note: Methods that may be used in the process include but are not limited to wireless network scans, physical/logical inspections of system components and infrastructure, network access control (NAC), or wireless IDS/IPS.

Whichever methods are used, they must be sufficient to detect and identify both authorized and unauthorized devices.

11.1.1 Maintain an inventory of authorized wireless access points including a documented business justification.

11.1.2 Implement incident response procedures in the event unauthorized wireless access points are detected.

11.2 Run internal and external network vulnerability scans at least quarterly and after any significant change in the network (such as new system component installations, changes in network topology, firewall rule modifications, product upgrades).

Note: Multiple scan reports can be combined for the quarterly scan process to show that all systems were scanned, and all applicable vulnerabilities have been addressed. Additional documentation may be required to verify non-remediated vulnerabilities are in the process of being addressed.

For initial PCI DSS compliance, it is not required that four quarters of passing scans be completed if the assessor verifies 1) the most recent scan result was a passing scan, 2) the entity has documented policies and procedures requiring quarterly scanning, and 3) vulnerabilities noted in the scan results have been corrected as shown in a re-scan(s). For subsequent years after the initial PCI DSS review, four quarters of passing scans must have occurred.

11.2.1 Perform quarterly internal vulnerability scans and rescans as needed, until all "high-risk" vulnerabilities (as identified in Requirement 6.1) are resolved. Scans must be performed by qualified personnel.

11.2.2 Perform quarterly external vulnerability scans, via an Approved Scanning Vendor (ASV) approved by the Payment Card Industry Security Standards Council (PCI SSC). Perform rescans as needed, until passing scans are achieved.

Note: Quarterly external vulnerability scans must be performed by an Approved Scanning Vendor (ASV), approved by the Payment Card Industry Security Standards Council (PCI SSC). Refer to the ASV Program Guide published on the PCI SSC website for scan customer responsibilities, scan preparation, etc.

11.2.3 Perform internal and external scans, and rescans as needed, after any significant change. Scans must be performed by qualified personnel.

11.3 Implement a methodology for penetration testing that includes the following:

- Is based on industry-accepted penetration testing approaches (for example, NIST SP800-115).
- Includes coverage for the entire CDE perimeter and critical systems.
- Includes testing from both inside and outside the network.
- Includes testing to validate any segmentation and scope-reduction controls.
- Defines application-layer penetration tests to include, at a minimum, the vulnerabilities listed in Requirement 6.5.
- Defines network-layer penetration tests to include components that support network functions as well as operating systems.
- Includes review and consideration of threats and vulnerabilities experienced in the last 12 months.
- Specifies retention of penetration testing results and remediation activities results.

11.3.1 Perform external penetration testing at least annually and after any significant infrastructure or application upgrade or modification (such as an operating system upgrade, a sub-network added to the environment, or a web server added to the environment).

11.3.2 Perform internal penetration testing at least annually and after any significant infrastructure or application upgrade or modification (such as an operating system upgrade, a sub-network added to the environment, or a web server added to the environment).

11.3.3 Exploitable vulnerabilities found during penetration testing are corrected and testing is repeated to verify the corrections.

11.3.4 If segmentation is used to isolate the CDE from other networks, perform penetration tests at least annually and after any changes to segmentation controls/methods to verify that

the segmentation methods are operational and effective, and isolate all out-of-scope systems from systems in the CDE.

11.3.4.1 Additional requirement for service providers only: If segmentation is used, confirm PCI DSS scope by performing penetration testing on segmentation controls at least every six months and after any changes to segmentation controls/methods.

11.4 Use intrusion-detection and/or intrusion-prevention techniques to detect and/or prevent intrusions into the network. Monitor all traffic at the perimeter of the cardholder data environment as well as at critical points in the cardholder data environment, and alert personnel to suspected compromises. Keep all intrusion-detection and prevention engines, baselines, and signatures up to date.

11.5 Deploy a change-detection mechanism (for example, file-integrity monitoring tools) to alert personnel to unauthorized modification (including changes, additions, and deletions) of critical system files, configuration files, or content files; and configure the software to perform critical file comparisons at least weekly.

Note: For change-detection purposes, critical files are usually those that do not regularly change, but the modification of which could indicate a system compromise or risk of compromise. Change-detection mechanisms such as file-integrity monitoring products usually come pre-configured with critical files for the related operating system. Other critical files, such as those for custom applications, must be evaluated and defined by the entity (that is, the merchant or service provider).

11.5.1 Implement a process to respond to any alerts generated by the change-detection solution.

11.6 Ensure that security policies and operational procedures for security monitoring and testing are documented, in use, and known to all affected parties.

Requirement 12

This requirement is broken down into the 39 controls listed below.

12.1 Establish, publish, maintain, and disseminate a security policy.

12.1.1 Review the security policy at least annually and update the policy when the environment changes.

12.2 Implement a risk-assessment process that:

- Is performed at least annually and upon significant changes to the environment (for example, acquisition, merger, relocation, etc.),
- Identifies critical assets, threats, and vulnerabilities, and
- Results in a formal, documented analysis of risk.

Examples of risk-assessment methodologies include but are not limited to OCTAVE, ISO 27005 and NIST SP 800-30.

12.3.1 Develop usage policies for critical technologies and define proper use of these technologies.

Note: Examples of critical technologies include, but are not limited to, remote access and wireless technologies, laptops, tablets, removable electronic media, e-mail usage and Internet usage.

Ensure these usage policies require the following:

- Explicit approval by authorized parties.

12.3.2 Develop usage policies for critical technologies and define proper use of these technologies.

Note: Examples of critical technologies include, but are not limited to, remote access and wireless technologies, laptops, tablets, removable electronic media, e-mail usage and Internet usage.

Ensure these usage policies require the following:

- Authentication for use of the technology.

12.3.3 Develop usage policies for critical technologies and define proper use of these technologies.

Note: Examples of critical technologies include, but are not limited to, remote access and wireless technologies, laptops, tablets, removable electronic media, e-mail usage and Internet usage.

Ensure these usage policies require the following:

- A list of all such devices and personnel with access.

12.3.4 Develop usage policies for critical technologies and define proper use of these technologies.

Note: Examples of critical technologies include, but are not limited to, remote access and wireless technologies, laptops, tablets, removable electronic media, e-mail usage and Internet usage.

Ensure these usage policies require the following:

- A method to accurately and readily determine owner, contact information, and purpose (for example, labeling, coding, and/or inventorying of devices).

12.3.5 Develop usage policies for critical technologies and define proper use of these technologies.

Note: Examples of critical technologies include, but are not limited to, remote access and wireless technologies, laptops, tablets, removable electronic media, e-mail usage and Internet usage.

Ensure these usage policies require the following:

- Acceptable uses of the technology.

12.3.6 Develop usage policies for critical technologies and define proper use of these technologies.

Note: Examples of critical technologies include, but are not limited to, remote access and wireless technologies, laptops, tablets, removable electronic media, e-mail usage and Internet usage.

Ensure these usage policies require the following:

- Acceptable network locations for the technologies.

12.3.7 Develop usage policies for critical technologies and define proper use of these technologies.

Note: Examples of critical technologies include, but are not limited to, remote access and wireless technologies, laptops, tablets, removable electronic media, e-mail usage and Internet usage.

Ensure these usage policies require the following:

- List of company-approved products.

12.3.8 Develop usage policies for critical technologies and define proper use of these technologies.

Note: Examples of critical technologies include, but are not limited to, remote access and wireless technologies, laptops, tablets, removable electronic media, e-mail usage and Internet usage.

Ensure these usage policies require the following:

- Automatic disconnect of sessions for remote-access technologies after a specific period of inactivity.

12.3.9 Develop usage policies for critical technologies and define proper use of these technologies.

Note: Examples of critical technologies include, but are not limited to, remote access and wireless technologies, laptops, tablets, removable electronic media, e-mail usage and Internet usage.

Ensure these usage policies require the following:

- Activation of remote-access technologies for vendors and business partners only when needed by vendors and business partners, with immediate deactivation after use.

12.3.10 Develop usage policies for critical technologies and define proper use of these technologies.

Note: Examples of critical technologies include, but are not limited to, remote access and wireless technologies, laptops, tablets, removable electronic media, e-mail usage and Internet usage.

Ensure these usage policies require the following:

- For personnel accessing cardholder data via remote-access technologies, prohibit the copying, moving, and storage of cardholder data onto local hard drives and removable electronic media, unless explicitly authorized for a defined business need.
- Where there is an authorized business need, the usage policies must require the data be protected in accordance with all applicable PCI DSS Requirements.

12.4 Ensure that the security policy and procedures clearly define information security responsibilities for all personnel.

12.4.1 Additional requirement for service providers only: Executive management shall establish responsibility for the protection of cardholder data and a PCI DSS compliance program to include:

- Overall accountability for maintaining PCI DSS compliance
- Defining a charter for a PCI DSS compliance program and communication to executive management

12.5.1 Assign to an individual or team the following information security management responsibilities:

- Establish, document, and distribute security policies and procedures.

12.5.2 Assign to an individual or team the following information security management responsibilities:

- Monitor and analyze security alerts and information and distribute to appropriate personnel.

12.5.3 Assign to an individual or team the following information security management responsibilities:

- Establish, document, and distribute security incident response and escalation procedures to ensure timely and effective handling of all situations.

12.5.4 Assign to an individual or team the following information security management responsibilities:

- Administer user accounts, including additions, deletions, and modifications.

12.5.5 Assign to an individual or team the following information security management responsibilities:

- Monitor and control all access to data.

12.6 Implement a formal security awareness program to make all personnel aware of the importance of cardholder data security.

12.6.1 Educate personnel upon hire and at least annually.

Note: Methods can vary depending on the role of the personnel and their level of access to the cardholder data.

12.6.2 Require personnel to acknowledge at least annually that they have read and understood the security policy and procedures.

12.7 Screen potential personnel prior to hire to minimize the risk of attacks from internal sources. (Examples of background checks include previous employment history, criminal record, credit history, and reference checks.)

Note: For those potential personnel to be hired for certain positions such as store cashiers who only have access to one card number at a time when facilitating a transaction, this requirement is a recommendation only.

12.8.1 Maintain and implement policies and procedures to manage service providers with whom cardholder data is shared, or that could affect the security of cardholder data, as follows:

- Maintain a list of service providers including a description of the service provided.

12.8.2 Maintain and implement policies and procedures to manage service providers with whom cardholder data is shared, or that could affect the security of cardholder data, as follows:

- Maintain a written agreement that includes an acknowledgement that the service providers are responsible for the security of cardholder data the service providers possess or otherwise store, process or transmit on behalf of the customer, or to the extent that they could impact the security of the customer's cardholder data environment.

Note: The exact wording of an acknowledgement will depend on the agreement between the two parties, the details of the service being provided, and the responsibilities assigned to each party. The acknowledgement does not have to include the exact wording provided in this requirement.

12.8.3 Maintain and implement policies and procedures to manage service providers with whom cardholder data is shared, or that could affect the security of cardholder data, as follows:

- Ensure there is an established process for engaging service providers including proper due diligence prior to engagement.

12.8.4 Maintain and implement policies and procedures to manage service providers with whom cardholder data is shared, or that could affect the security of cardholder data, as follows:

- Maintain a program to monitor service providers' PCI DSS compliance status at least annually.

12.8.5 Maintain and implement policies and procedures to manage service providers with whom cardholder data is shared, or that could affect the security of cardholder data, as follows:

- Maintain information about which PCI DSS requirements are managed by each service provider, and which are managed by the entity.

12.9 Additional requirement for service providers only: Service providers acknowledge in writing to customers that they are responsible for the security of cardholder data the service provider possesses or otherwise stores, processes, or transmits on behalf of the customer, or to the extent that they could impact the security of the customer's cardholder data environment.

Note: The exact wording of an acknowledgement will depend on the agreement between the two parties, the details of the service being provided, and the responsibilities assigned to each party. The acknowledgement does not have to include the exact wording provided in this requirement.

12.10 Implement an incident response plan. Be prepared to respond immediately to a system breach.

12.10.1 Create the incident response plan to be implemented in the event of system breach. Ensure the plan addresses the following, at a minimum:

- Roles, responsibilities, and communication and contact strategies in the event of a compromise including notification of the payment brands, at a minimum.
- Specific incident response procedures.
- Business recovery and continuity procedures.
- Data backup processes.
- Analysis of legal requirements for reporting compromises.
- Coverage and responses of all critical system components.

- Reference or inclusion of incident response procedures from the payment brands.

12.10.2 Review and test the plan, including all elements listed in Requirement 12.10.1, at least annually.

12.10.3 Designate specific personnel to be available on a 24/7 basis to respond to alerts.

12.10.4 Provide appropriate training to staff with security breach response responsibilities.

12.10.5 Include alerts from security monitoring systems, including but not limited to intrusion-detection, intrusion-prevention, firewalls, and file-integrity monitoring systems.

12.10.6 Develop a process to modify and evolve the incident response plan according to lessons learned and to incorporate industry developments.

12.11 Additional requirement for service providers only: Perform reviews at least quarterly to confirm personnel are following
security policies and operational procedures.

Reviews must cover the following processes:

- Daily log reviews.
- Firewall rule-set reviews.
- Applying configuration standards to new systems.
- Responding to security alerts.
- Change management processes.

12.11.1 Additional requirement for service providers only: Maintain documentation of quarterly review process to include:

- Documenting results of the reviews.
- Review and sign-off of results by personnel assigned responsibility for the PCI DSS compliance program.