Achieving PCI 2.0 Compliance with Virtual and Cloud Computing with IBM i

By Mel Beckman
The PCI-DSS standard has had a tough time keeping up with the pace of information technology. Version 1.0 was obsolete before it was completed, failing to account for a nascent yet pervasive technology: wireless networking. Indeed, the largest credit card breach in history—the massive TJ Maxx security fiasco—occurred just a year after PCI-DSS went into force, enabled by weak data encryption (Wired Equivalent Privacy, or WEP) allowed by the standard. Even the PCI-DSS 1.1 revision didn’t address the wireless vulnerability problem.

PCI-DSS 1.2 debuted in 2008, and while it finally resolved wireless exposures, it was still behind the technology power curve. Infrastructure virtualization was in full swing by then, but the 1.2 standard never mentions the term, and affects virtualization security issues only obliquely. Section 2.2.1 requires that a server perform only one primary function, which would seem to preclude the benefits of virtualization, which enables multiple functions running on a single physical host. Yet IT administrators readily interpreted this rule to apply to virtual and physical servers, despite the fact that version 1.2 fails to consider any of the unique security risks of hypervisors and virtual network environments.

In June 2011 the PCI Security Standards Council finally recognized virtualization in a 40-page Information Supplement entitled PCI DSS Virtualization Guidelines—which while providing some helpful recommendations, fails to specify compliance criteria, and barely touches the subject of cloud computing. Yet both of these technologies are routinely employed in modern IT administration, both for their cost advantages and their ability to improve business competitiveness through more agile responses to market changes. Virtualization extends to both public and private clouds, with public clouds delivering valuable software-as-a-service (SaaS) functions such as tokenization, and private clouds delivering Infrastructure-as-a-Service within a single enterprise.

Getting back to PCI, the Council, rather than create new compliance criteria directed at virtualization and cloud security, chose to instead empower industry organizations—providers of networking and server hardware, software, and services—to devise their own PCI-compliant best practices in the form of new PCI-compliant cloud reference architectures. And in fact, one large coalition

\[1\] www.pcisecuritystandards.org/documents/Virtualization_InfoSupp_v2.pdf
of such vendors has released a viable reference architecture, discussed here.

Understanding the new cloud reference architecture approach is the key to leveraging modern IT practices for the secure processing of retail payments. These architectures define methods for including logical, as well as physical, facilities in the cardholder data environment: virtual machines, networks, and storage. This paper examines how a reference architecture addresses virtualization and cloud computing security risks, by expanding the scope of the cardholder data environment. It also looks at how IBM is uniquely positioned to exploit the new cloud marketplace, both in consuming software and services, and in delivering them. You’ll learn the significance of hypervisor design in minimizing vulnerabilities, how your PCI-DSS compliance posture must change, and how to select an architecture that works for your payment processing workflow.

Scope Leap
The original cardholder data environment (CDE) described by PCI-DSS 1.2 and earlier was straightforward and easily isolated in order to limit the scope of PCI-compliant protective measures (Figure 1).

![Figure 1: The Pre-2.0 Cardholder Data Environment](image)

The CDE comprised every physical device which unprotected cardholder data traversed—servers, databases, and user terminals—as well as all the border elements, such as firewalls, routers and switches. Everything outside the CDE must be encrypted, and the area within the CDE perimeter is subject to all six PCI-DSS areas of compliance:

1. Build and maintain a secure network
2. Protect cardholder data
3. Maintain a vulnerability management program
4. Implement strong access control measures
5. Regularly test and monitor networks
6. Maintain an information security policy

The PCI-DSS Virtualization Guidelines document brings several new components into the CDE: the hypervisor, the virtualization management network fabric, virtualization management consoles, and the virtual machine networking fabric (Figure 2).

![Figure 2: CHD Environment Scope Extension for Virtualization](image)

Effectively securing this drastically enlarged scope is the key challenge faced by any PCI-compliant cloud reference architecture. The core element, the hypervisor, presents a variety of new risks. Should the hypervisor itself be compromised, every virtual machine running above it is also compromised. This risk extends to the management consoles that control hypervisor operation, starting and stopping virtual machines (VMs), and moving VMs between physical hosts. These consoles interact with the hypervisor over a network fabric that should ideally be physically separate and isolated from all other networks, including the public Internet.

Similarly, virtual machines that run payment-processing software, such as web servers and databases, may coexist on the same physical host as completely unrelated applications. A reference architecture must identify how PCI and non-PCI workloads are kept separate. These workloads communicate with each other over a VM network fabric that must ensure traffic from separate realms never co-mingle, and that no risk of penetration of the PCI virtual networks exists from either other VMs or the outside world.

Other risk mitigations include the need for additional role definitions and authentication, and control over possible information leakage points such as VM snapshots and live mobility.
As you’ll see shortly, it is possible to design an architecture, and implement one, that meets these requirements. But before examining reference architecture details, it’s worth considering how IBM i can play competitively in this new PCI arena.

IBM i’s Virtualization Pedigree

As you’ll see shortly, it is possible to design an architecture, and implement one, that meets these requirements. But before examining reference architecture details, it’s worth considering how IBM i can play competitively in this new PCI arena.

The System/370 and its successors continued to use virtualization effectively, but IBM’s smaller systems, including the midrange System/3, /32, and /34, lacked the technology. IBM’s System/38 laid the foundation for midrange virtualization through its single-level store memory architecture and object-oriented high-level machine interface. In 1990, long before PCs began to support virtualization, IBM added the Logical Partition (LPAR) virtualization capability to the System/38 follow-on, the AS/400. It wasn’t until 1999 that VMware shipped its first x86 enterprise virtualization product.

The AS/400 became the iSeries, then the System i, and finally the IBM i that currently runs on IBM’s Power platform, including the newest Pure Flex “cloud in a box” systems. Over time, IBM migrated the LPAR feature into hardware and firmware, to create a cross-platform hypervisor called PowerVM, which can run AIX and Linux in addition to IBM i OS.

Unlike x86 virtualization employed by VMware and Microsoft, PowerVM has a truly immutable hypervisor, making it much more hardened substrate upon which to run virtual workloads. This positions it well as a service delivery platform for payment card services, since one of the most critical components of virtual infrastructure—the hypervisor—has a significantly smaller attack surface that competing hypervisors.

Along with this material design advantage, IBM is already well positioned as a major public and private cloud provider. IBM’s SmartCloud public cloud offering, while not currently delivering IBM i services, can host Linux- and AIX-based payment card services that merchants running IBM i systems often purchase. New payment modalities, such as nearfield communications, and data protection measures such as tokenization, are typical services IBM i applications would consume.

IBM i plays a role in private cloud hosting, where its Virtual System (LPAR) capability is widely used in disaster recovery and business continuity, especially in the financial sector. Several third party providers sell IBM i VMs as IaaS commodities, providing multi-tenancy that can be incorporated into a PCI-compliant implementation through a reference architecture. And IBM’s Pure Flex line will make i-Clouds more practical to deploy, both for IaaS providers, and in private cloud installations operated by individual enterprises.

Given that IBM i is already the business hub for many vertical markets, virtualization extends the life of applications whose business logic is embedded in the i platform. The proven reliability and security of i make it a strong contender in e-commerce and other payment card venues, and its ability to scale rapidly and rearrange workloads to ensure application performance levels reinforces that strength.

Hypervisors: Type 1 vs. Type 2

Cloud services are built on virtualization, and virtualization is built on the hypervisor. But not all hypervisors are created equal. Two broad classes of hypervisor exist, and one is probably more suitable for PCI compliance than other.

The Type 1 hypervisor (Figure 3) is a thin software layer running directly on the physical processor of the host server. The hypervisor connects directly to all I/O and storage devices, through embedded device drivers created specifically for the hypervisor. On top of the hypervisor is a layer of three support software components: virtual management interfaces, device virtualization that multiplexes I/O requests from virtual workloads, and the Virtual Machine Monitor, a console for starting and stopping workloads. The virtual workloads themselves, termed virtual machines, run at the top layer, and contain guest operating systems and applications.
The red badges in Figure 3 depict the possible attack surfaces presented by a Type 1 hypervisor. The hypervisor itself is not a general purpose operating system, but consists of just the code necessary to perform virtual machine execution control functions. It runs in read-only storage, making it impossible to alter. Thus it has few, if any, internal vulnerabilities. The most vulnerable points are where the hypervisor interfaces with I/O devices and management components, and these can be protected by isolating them from external network access.

Contrast this with the Type 2 hypervisor, which runs on top of a traditional general-purpose operating system such as Linux, Windows, or Unix (Figure 4).

Figure 3: The Type 1 (Bare Metal) Hypervisor

The host OS, being a complex component designed to provide all manner of computational and management functions, has a huge code base, and runs in read/write memory, making it vulnerable to a host of attacks. These attacks include code modification, buffer overflow, Trojan injection, and virus infection, to name just a few. It thus provides a much larger attack surface than a Type 1 hypervisor. Type 2 hypervisors have their place, but secure environments such as payment processing are not one of them.

A key requirement of any PCI-compliant reference architecture is thus a mandate that a Type-1 hypervisor be at the core of the implementation. Examples of Type 1 hypervisors that fit this role are:

- IBM PowerVM (including IBM i)
- VMware ESX
- Citrix XenServer
- Microsoft Hyper-V Server (bare metal edition)

Examples of Type 2 hypervisors that should never be employed in a PCI-compliant architecture include:

- Microsoft Windows 2008 Server Hyper-V
- VMware Workstation
- VirtualBox

It should be noted that Microsoft considers its Windows 2008 Server Hyper-V to be a Type 1 hypervisor because it loads before the OS. However, only the bare-metal Hyper-V can be launched and run from read-only boot storage, leaving it less secure than competing Type 1 hypervisors.

PCI-DSS Compliance for Virtualization and Cloud Computing

Although neither the PCI DSS Virtualization Guidelines published in 2011, nor the PCI-DSS 2.0 standard promulgated in 2012, give specific compliance requirements for virtualization and cloud, the documents do provide some useful guidance that reference architecture designers must consider.
The PCI DSS Virtualization Guidelines don’t set forth any requirements for prospective reference architecture developers. In general, any competent technologist or group of technologists could create a reference architecture. The architecture isn’t approved in advance by the PCI Security Standards Council, but is instead assessed by the compliance certification process each time it’s implemented.

At this writing, only one public architecture has been published. An ad-hoc industry association formed by Cisco, Coalfire, HyTrust, Savvis, Trend Micro, and VMware published their untitled reference architecture in August 2011, and updated it in December. Studying this architecture is helpful even if you do not end up using it. At a minimum it makes an excellent basis for extension, and it’s been used successfully to achieve PCI DSS 2.0 compliance.

**Compliance Within Cloud Applications**

No amount of external security measures can ensure PCI-DSS compliance if the payment application itself, running in the cloud, has security vulnerabilities. There are three key aspects to keeping an application compliant once it moves to private or public cloud infrastructure:

- Enhanced role-based authentication for all users, particularly administrative (“power”) users, enforcing “least privilege” access.
- Audited cardholder data protection measures (e.g., encryption)
- Independently secured application access (e.g., web sessions, VPN tunnels) using access control measures such as two-factor authentication

Most applications in traditional IT environments rely on close coupling between the application and local user credential databases. In both public and private clouds, however, the credential database is likely located elsewhere – in a separate VM or in a completely different cloud. Implementing role-based authentication securely across separate application domains requires careful attention to detail. Authentication transactions must be conducted across protected channels, such as VPN.
tunnels, that guard against man-in-the-middle attacks. And the authentication process and rights management must be fine-grained enough to ensure that no users can gain undeserved elevated privileges without. So-called “power” users, in particular, who by definition have enhanced access, must be limited to the minimum capabilities needed to accomplish their tasks.

All processes involved in protecting cardholder data, including encryption engines, key management facilities, and tunnel endpoint controls, require enhanced monitoring, to demonstrate that they continue to operate as planned. Ideally this monitoring should include real-time event generation to alert security staff to out-of-compliance conditions. And all audit logs should be immutable: impossible to alter once written.

And where users in a legacy infrastructure environment ensured protection with a combination of user ID, password, and physical control, cloud-based user authentication must be augmented with at least two-factor authentication. If additional authentication factors are added through OS hooks, such as login exception processing, those hooks must operate over secure channels.

The Cisco et al Reference Architecture

As the first reference architecture, Cisco and its partners went beyond a simple specification, prefacing their design description with their detailed rationale. This rationale identifies several challenges posed by the PCI DSS Virtualization Guidelines:

- Defining the scope of compliance in virtualization and cloud realms
- Addressing multi-tenancy and co-mingled assets
- Demonstrating PCI compliance in the absence of clear directives

The document defines common terms and elements, illustrating these in a cloud-provider scope diagram (Figure 6).

![Figure 6: Cisco et al, Reference Architecture Scope Diagram](image)

The scope diagram draws a clear boundary around the virtualization environment, identifying the perimeter security elements such as application firewalls, the border firewall, intrusion detection and prevention, and data leak controls. It then identifies three network layers that attach to that environment: distribution/core, physical, and virtual. The model also depicts various operational component, including customer portals, server and device management components, and outside provider services and staff.

![Figure 7: Reference Architecture Business Model](image)

The reference architecture then describes data flow in the form of a business model (Figure 7). This diagram constitutes the actual reference architecture that protects the network scope depicted in Figure 6. It shows the data flow between compo-
nents, and identifies four layers of management: application, business, service, and infrastructure. Data flow begins with a cloud application (which is could be implemented on either a public or private cloud).

The cloud application layer defines the interface to end users of the application, and limits both user and administrative access to only this layer, prohibiting direct access to lower layers in the architecture.

The application layer communicates with the business orchestration layer, which layer contains a service catalog and the service design, composed made of configuration elements and policies. The idea is that this layer is managed by non-technical staff concerned only with business aspects of the application, not low-level implementation details. This provides a secure boundary for administrative purposes.

The service orchestration layer contains the provisioning logic for the cloud infrastructure. This layer provides a portal for configuring virtual compute, network, storage, and security elements. This is a technical process requiring IT skills, but is still insulated from the physical hardware. At the bottom is the infrastructure layer, which directly administers physical computers, switches, routers, storage arrays, and all higher management interfaces. This layer is highly sensitive from a security standpoint because it is where hypervisor control is vested. Thus access to this layer should require the highest caliber authentication controls.

An overriding principle of this layered architecture is the requirement for role-based access controls (RBAC), which ensures separation of duties among staff members, which helps prevent malicious collusion. By extending RBAC across the entire infrastructure, both logical and physical, the reference architecture goes beyond the letter of the law embodied in the DSS Virtualization Guidelines, which hopefully will make it more flexible in adapting to future PCI-DSS standards.

The reference architecture coalition actually implemented their design as a validation step, and provides a high-level diagram of that implementation (Figure 8). This design verification step employed specific hardware and software components, also detailed in the reference architecture document, with enough detail that an IT professional with moderate to advanced skills could built an environment that passes PCI-DSS assessment audits. A significant aspect of the implementation is that not all components are off-the-shelf. For example, hypervisor security had to be augmented with custom code using hypervisor introspection APIs.

Cloud-Capable

PCI-DSS 2.0 provides enough guidance—and permission—that a reasonably skilled IT team can devise its own PCI-compliant architecture for virtualization and cloud payment processing environments. IBM’s strong virtualization capabilities makes it a viable platform for deploying and consuming cloud services. Although you are free to build your own PCI-compliant reference architecture from scratch, the published reference architecture provides both valuable design insight and an excellent starting point for a provably practical PCI cloud implementation.
Achieving PCI 2.0 Compliance with Virtual and Cloud Computing with IBM i

An interview with Don Nguyen, Enforcive's Director of Technical Services

Q. PCI compliance in virtual environments requires diligent management of multiple security tasks. How does Enforcive help administer the whole PCI-DSS effort?

To help our customers quickly and efficiently comply with PCI-DSS on IBM i, Enforcive developed the PCI Accelerator Package as an add-on to the Enterprise Security Suite. It provides a set of pre-defined reports, alerts and templates that greatly reduce compliance project timelines. Enforcive's templates allow system administrators, auditors and executives to utilize a shared interface to manage policies, spot compliance deviations and enforce policies throughout the enterprise.

Q. Is customizable auditing necessary for PCI-DSS compliance in a cloud environment?

It is critical to have customizable auditing. Section 10 of the PCI-DSS document requires that all access to Card Holder Data is audited and readily available for reporting. The Enterprise Security Suite has built-in tools that will streamline the auditing process. The PCI Accelerator Package includes canned reports that can be easily duplicated, customized and categorized into policy groups, and scheduled to run across multiple IBM i servers or partitions.

Q. Continuous vulnerability assessment is the best way to detect zero-day risks and mitigate them. How can IBM i users add continuous VA to their security capabilities?

IBM i users can enhance vulnerability assessment with Enforcive's real-time alerting capabilities. In Enforcive's Alert Center, users can include triggers in their alerts that not only notify of, but react to suspicious or undesirable transactions. Triggers can be configured to revoke special authority, disable a user profile or prompt the execution of a custom program.

Q. PCI-DSS places extra demands on role-based authentication and granular, least-privilege-based access controls. How can IBM i users leverage the platform’s intrinsic object-based security to simplify role-based controls?

Most ERPs in use today were developed years ago with very broad data access rules. Users typically belong to a very powerful group profile and inherit its data access rights. In the past, data access was granted through a menu scheme where data access is easily regulated. With the introduction of Exit Points (back doors) to the IBM i, a menu scheme is no longer secured. With Enforcive's Exit Point Access Control module, these ‘back doors’ can be protected by implementing object based security which drastically simplifies role-base controls.

Mel Beckman is a senior technical editor for Penton Windows IT Pro. He has built two regional Internet service providers and is currently president of Beckman Software Engineering, a technical consultancy specializing in large-scale, high-bandwidth networks. His past clients include Apple Computer, the City and County of Santa Barbara, DuPont Displays, IBM, Loral Federal Systems, United Airlines, the U.S. Department of Agriculture, and the U.S. Department of Energy. Mel has presented seminars on computer programming and network technology throughout the United States, Europe, and Asia.