CHAPTER

Securing the Cloud: Architecture

4

INFORMATION IN THIS CHAPTER

- Security Requirements for the Architecture
- Security Patterns and Architectural Elements
- Cloud Security Architecture
- Planning Key Strategies for Secure Operation

Chapter 2 presented the National Institute of Standards and Technology (NIST) definition of cloud computing as an Information Technology (IT) model for “enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction.”[1] But what does that translate to when you are building a cloud? At a high level: A data center (a.k.a. infrastructure life support), hardware (servers, storage, and networking), a broad set of enabling software, a staff with broad and deep experience, and process to make it work. Figure 4.1 depicts a high-level view of these components.

Operating a cloud securely and efficiently entails a great deal of advance planning. At a high level, we start with a data center and redundant Internet connections that connect to a cloud ingress. This ingress constitutes the technology portion of an information security boundary[A] that is comprised of some combination of network devices that serve to safely enable communication. NIST defines it as “the process of uniquely assigning information resources to an information system defines the security boundary for that system.”[2]

Inside this boundary we have a massive amount of gear that is racked and cabled following defined patterns. There will also need to be some infrastructure that is used to manage the cloud and its resources as it operates. Going further, each component—server, storage, and network—requires some degree of configuration. This overall picture is one of numerous components that are organized in part according to visually evident patterns.

When designing or planning any complex system, it is important to look ahead and consider the processes and procedures that will be necessary for operation. Although it is possible to build a small cloud without much planning,

[A] A security boundary can be defined by a set of systems and components that come under a single administrative control.

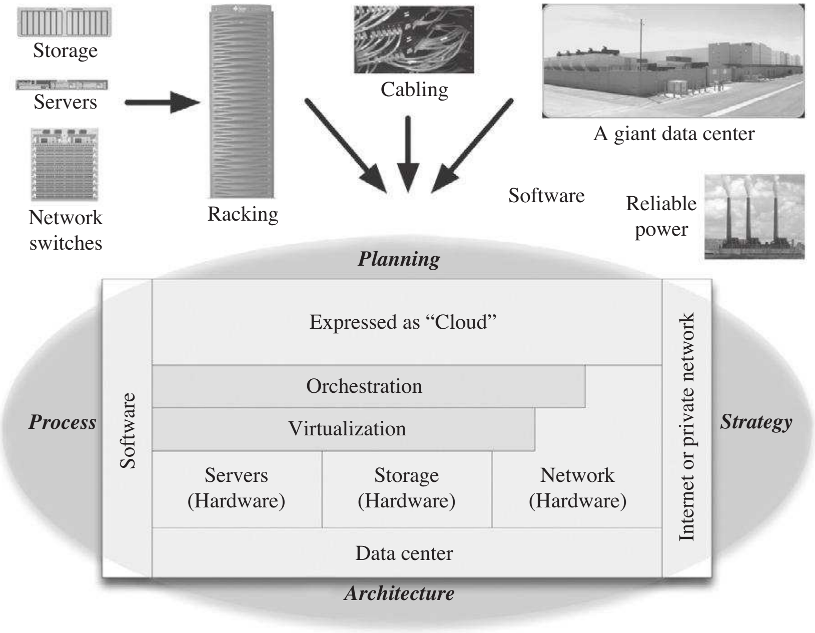

FIGURE 4.1

Cloud architecture and cloud implementation.

anything more substantial entails significant planning and design. Failing to plan appropriately will typically lead to higher ongoing costs due to inefficiencies in design and process and with operations that are not up to the time domain needs of managing a highly dynamic cloud. But what constitutes appropriate planning? Overplanning often entails misreading the future and doing so can result in significant rework and excessive cost. But failing to anticipate any change will result in a dead end and halted work. A better approach entails prudent architecture that accepts the need for inevitable evolution and reserves flexibility for such adaption as you face it.

Remember, in Chapter 1 (in the section Cloud is Driving Broad Changes), we introduced the notion that cloud offers advantages toward simplifying IT; in Chapter 2 (in the section Cloud Reference Architecture), we discussed the on-demand and self-service aspects of accessing cloud services. If cloud is to deliver on these promises, then the architecture must be designed and planned accordingly.

In this chapter, we will take a close look at the architectural components that can be used to build a cloud with an eye on security. We begin by identifying requirements for a secure cloud architecture along with key patterns and architectural elements. With that as a background, we will present and discuss several different cloud architectures. We will finish off the chapter with a brief discussion on key strategies for secure operation. Although this material is focused on work that will be undertaken by the CSP, tenants are often in the role of service providers for other users and hence this material will also apply to them.

SECURITY REQUIREMENTS FOR THE ARCHITECTURE

One goal for architecture is that it should be appropriate in meeting needs. This section surveys key architectural requirements for a typical cloud implementation. Several factors serve as the underlying motivation for requirements; these include:

- **Costs and Resources** The cloud provider's financial resources will act to constrain investment in technology, security controls included. But it is important to recognize that the absence of unlimited resources can be very motivating to how one designs, architects, and builds. For instance, if you know that your staff will be small, then this can force you toward process improvement and greater automation. Likewise, cost is also a motivation for the consumer of cloud services. The nature of these constraints tends toward the development of services with operating characteristics that are not ideal for all consumers.
- **Reliability** This is a quality that refers to the degree you can depend on a system to deliver its stated services. Reliability can be described as a guarantee that the underlying technology can provide delivery of services.
- **Performance** A measure of one or more qualities that have to do with the usefulness of a system. By example, common measures include responsiveness to input and the amount of throughput the system can handle.
- **The Security Triad** The essential security principles of confidentiality, integrity, and availability apply to most systems; the responsibility of a security architect is to match security controls with security requirements that sometimes must be derived from the need to assure the other three drivers (reliability, performance, and cost).
- **Legal and regulatory constraints** (we have covered these to some extent in Chapter 3) Legal and regulatory constraints can lead to the need for many additional requirements having to do with technical security controls, access policies, and retention of data among many others.

We begin with an unusual area of requirements for system security requirements: Physical security. But, by the time we are done you will see what motivates this.

Physical Security

Beginning with the facility that the cloud data center is hosted in, physical security is as important as any other security controls that seek to protect the security and operation of the cloud. Physical facilities are subjected to various threats, including natural hazards, human actions, and disaster. Building your cloud data center on a floodplain is as unwise as granting privileged access to all users.

The scope of issues in physical security is significant, and it involves a range of measures to avoid, prevent, detect, and respond to unauthorized access to the facility or to resources or information in the facility. Physical security for a facility should itself be viewed as a system for protection, with the individual security elements complementing each other in a multifaceted and layered defense. These elements will include aspects of environmental design, access control (including mechanical, electronical, and procedural), monitoring (including video, thermal, proximity along with environmental sensors), personnel identification and access controls, and intrusion detection in conjunction with response systems (lights, gates, and locking zones).

EPIC FAIL

In 2005, the author was involved in a substantial build of a public-facing grid computing data center. The site was in London in a former brewery (very thick walls) in a neighborhood that was at the edge of a wild bar and club scene.

On Monday mornings, the build team would arrive at the site being careful to avoid countless broken beer bottles at the entrance of the facility. The street entrance to the facility consisted of regular unreinforced glass that ran from floor to ceiling with an automatically locking glass door that operated very unreliably. Inside this space sat a regular office desk with an unarmed guard sitting behind it. The guard had two buttons to push, one unlocked the street door and the second button unlocked the door to the interior of the building. On the wall next to the guard was an unlocked cage key case; on the desk was a computer that was used to program the tenant's access cards to various floors, rooms, and cages in the building. The restrooms in the building were located to the rear, and each had a tilt-out window that an adult could easily climb through. A ladder tall enough to reach the second floor restroom window casually lay nearby.

The moral of this story is that this facility looked like it had physical security, but it was paper thin and ineffectively layered. On one occasion, the author's access card did not work for the correct zones and the guard was new to his role. After watching that the guard fails to navigate the process of using his computer console to grant correct access privileges, the author asked the guard if he might try driving the card security access software program. The answer was an astonishing *yes*…

The moral of this story was that because security control was undermined from the physical facility up, writing SLAs for tenants in this place was nearly impossible.

Physical security for a facility should be layered with each element integrated within an overall automated control and monitoring center. Planning for effective physical security entails deep consideration of circumstances that will be faced, these will include regular activities and unanticipated situations. Layered physical security elements must be supported by procedures that are appropriate and best implemented by a trained and professional staff. This physical security staff must be dedicated to the mission of protecting the assets and maintaining physical security procedures and processes even when a disaster unfolds. Given the scope and complexity of planning for physical security, a best practice is to engage experienced and recognized experts from the planning stage onward.

As stated in the introduction to this section, including physical security requirements in a section on requirements for security architecture may raise some eyebrows.

However, we live in a world where the boundaries between physical and virtual security are being increasingly blurred. There are obvious reasons why we should be concerned about the physical security of our cloud, but there are also virtual security reasons. As we will discuss later in the book (Chapter 6 in the section Security Monitoring), environmental sensors, physical security sensors, and camera imagery all represent information sources that can help resolve system security events and illuminate security situations that might otherwise raise alarm. In other words, security monitoring greatly benefits from such physical security sensor data.

Cloud Security Standards and Policies

Although some security requirements may be unique to the cloud implementation, it is important that requirements for cloud security should be consistent with appropriate standards, such as International Organization for Standardization (ISO) 27001 and ISO 27002—if one is to leverage a large body of practical experience, best practices, and review. Further, all aspects of security should be captured in a cloud security policy, which is best to develop as a formal document that has the complete approval and blessing of management. A security policy should be seen as the foundation from which all security requirements derive. Security policy should not detail technical or architectural approaches (as these may change more frequently than the policy) rather the policy should set forth the underlying requirements from an organizational or business standpoint. For instance, policy should explain the need for the use of standard-based encryption via use of a formally evaluated commercial product, rather than spelling out the use of Transport Layer Security, Secure Sockets Layer, or other specific means for communication security.

The security policy should also call for the development of several supporting documents, these should include:

- A set of guidelines for enabling security in development of infrastructure software, infrastructure management processes, and operational procedures.
- An acceptable use policy for each category of user, from internal operations, administrative, and other staff to tenants and end users. This policy should identify categories of use that are prohibited, why they are prohibited, and what the consequences for infractions are.
- A set of security standards for all aspects of the cloud, from development to operation. Security standards for a cloud should include:
  - **Access Controls** These should be at a granularity necessary to guide implementation of physical access to facilities and logical access to systems and applications.
  - **Incident Response and Management** This shall detail all roles and responsibilities of various parties along with procedures and timelines from detection through postmortem reporting.
  - **System and Network Configuration Backups** It is important to have a current and authoritative copy of all configurations including infrastructure components, servers, and switches as well as for all hosted systems.

- **Security Testing** The cloud provider must perform and document the results of initial and periodic security testing. This standard should include roles and responsibilities as well as detailing when third-party testing or reviews should be performed.
- **Data and Communications Encryption** This standard should detail functional areas (such as web server traffic), the approved cryptographic algorithms and the required key lengths.
- **Password Standards** This standard should detail the qualities that acceptable passwords must comply with (notably length and composition) and how the cloud provider will test compliance.
- **Continuous Monitoring** This standard should detail how configuration management and change control are performed to support ongoing security of the baseline as it evolves and is updated.

There are several other areas under the control of the cloud provider that benefit from the development of formal standards. Some of these include Termination of inactive sessions; Definition of roles and responsibilities for cloud personnel; Rotation of duties and vacation schedules; Magnetic and electronic media handling, including assured destruction procedures for media that can no longer be erased; Off-premises removal or use of equipment; The timely removal of user privileges; and Disaster recovery and continuity of operations.

Cloud Security Requirements

The security architecture of the cloud should be consistent with the intent of the security policy. Thus, the first security requirement is to develop a security policy for the cloud. An appropriate second requirement is the development of placeholders for each of the documents and standards listed in the previous section. At some point, a separate set of activities will revolve around identifying granular requirements that are preliminary in developing the cloud security architecture. Representative security requirements that are likely to apply to your cloud architecture are listed in the remainder of this section.

Cloud-wide Time Service

Since the correct operation of systems and authoritative system logs depend on the correct time, all systems must be synchronized to the same time source. Typically, this will be achieved by use of Network Time Protocol (NTP), which is one of the oldest Internet Protocols (IPs) that is still in use. Correct and synchronized time becomes especially important when you have communicating computers that reside in different locations, but which need to have their record and event timestamps synchronized to a single source. Once clocks drift between network devices and/or computers, a cloud infrastructure is subjected to all manner of errors and made difficult to diagnose failures.

In overview, correct time information comes from authoritative national time standards via multiple paths, including radio, cellular, satellite, and hard-wired transmissions to primary time servers. From these it is distributed via NTP subnets to literally millions of secondary servers and from there to end-clients. NTP provides Coordinated Universal Time (UTC), all time zone or daylight saving time information must be provided separately.

> **WARNING**
>
> Physical cloud infrastructure should include accurate, reliable, and verifiable time sources, such as WWV and GPS. The time system should be based on at least two reliable time source paths and devices for resilient and secure operation. All computers and network devices must obtain their time information for correct synchronization and reliable cloud operations. Best practices for managing NTP are the following:
>
> - Configure clients to reference at least two time servers to provide redundant time.
> - Accurate time synchronization depends on how frequently clients update their time from time servers.
> - Limit input network or radio broadcast signals to authoritative and *legal* ones.

Identity Management

Identity is a key element in the security of an operating cloud. This information must be correct and available to cloud components that have a validated need for access. Requirements include as follows:

- Controls must be implemented to protect the confidentiality, integrity, and availability of identity information.
- Implement an identity management system that will support the needs for authenticating cloud personnel.
- Implement an identity management system that will support the larger scale needs for authenticating cloud tenants and users.
- Consider using a federated identity system to allow for identity portability for the user population and to present a single mechanism for internal access as well as tenant and user access. A federated identity management system will allow for interoperability with customer and third-party identity providers or realms as may be appropriate.
- Verify identities of users at registration time in accordance with policy and legal requirements.
- Assure that when identities are deprovisioned, historical information for users is maintained to allow for future legal investigations.
- Assure that when user identities are deprovisioned and identities are recycled, access by a new user is not granted to a previous users data, contexts, or other private information resources. This amounts to assure that at the appropriate level in the identity system, user identities are never actually reused thus preventing future conflicts or confusion.
- Implement the means for customers to verify assertions of identity by cloud provider personnel.

Access Management

Access controls use identity information to enable and constrain access to an operating cloud and its supporting infrastructure. Requirements include as follows:

- Cloud personnel shall have restricted access to customer data in general. Cloud personnel may require access to a hypervisor on a customer-allocated machine or to storage devices that host customer VMs or customer data, but such access shall be tightly constrained and limited to specific operations that are well defined by the security policy and SLAs. Implement need-to-know procedures for cloud personnel to prevent unnecessary opportunity for access to customer data.
- Implement multifactor authentication for highly privileged operations. Apply additional security controls for highly privileged operations. Assure that authorization mechanisms for cloud management are constrained and do not allow for cloud-wide access.
- Do not allow the use of accounts that are shared (such as administrator), instead use *sudo* or the equivalent to gain auditable privilege and only allow such access for users who are members of the appropriate role.
- Implement the least privilege principal (LPP) when assigning permissions. Implement role-based access controls (RBAC) to appropriately constrain access by authorized users on the basis of their role.
- Implement whitelisted source IP addresses for all remote control or remote access by operations personnel. Where whitelisted IPs are not feasible, require access to proceed through additional mechanisms, such as hardened jump hosts or gateways.

TIP

Under unusual circumstances, it may become necessary for the cloud provider to gain emergency access to certain cloud control functions or to tenant VMs. In anticipation of this sort of circumstance, you should consider the use of alarmed *break glass* strategy. With break glass (the name derives from breaking the glass to pull a fire alarm), security controls that are always in place can be bypassed in the event of an emergency.

A break glass procedure must be clearly defined and well understood, it should be well documented and tested. Such a strategy can be based on prestaged emergency-only privileged accounts that should only be used under specifically defined circumstances.

However, the consequences of doing so must be severe if the circumstances are found to not warrant having done so. Part of the procedure should include formal reporting on the circumstances that lead to the need to invoke break glass. These procedures should also include steps to cleanup after such emergency accounts or procedures are used.

Key Management Requirements

In a cloud, encryption is a primary means to protect data at rest (storage) and between storage and processing phases. Requirements for key management include as follows:

- Ensure that appropriate controls are in place to limit access to keying material that the cloud provider maintains control over.
- Ensure that root level and signing keys are managed appropriately.

- For multiple site cloud infrastructure, ensure that key revocation is performed without side effects or undue delay.
- Ensure that procedures are effective for recovering from compromised keys.
- Protect and encrypt all customer data and VM images at all appropriate phases of their life cycle.

System and Network Auditing

System and network security event logs are a keystone for managing the ongoing security of any system. In a cloud, audit events will be generated in fundamentally different trust zones; these range from highly secured network and security components to systems where the CSP grants significant control to tenants or users. Thus, security events should be recognized as having different degrees of integrity. The following are key requirements for the generation and management of audit events:

- Auditing is required for all operational systems, from infrastructure system and network components up to but not necessarily including customer VMs. Tenant confidentiality agreements along with service contracts may set the boundary for what data can be collected within a tenant VM, and in many cases tenant virtual networks.
- All security-relevant events must be recorded with all relevant information that is necessary to analyze the event; this shall include the correct time, resolvable system, and user IDs and appropriate event codes and supporting information.
- Generated audit events must be logged in a near-real-time manner. The correct operation of auditing and logging shall be verified on an ongoing basis using means such as heartbeat or call-and-respond.
- All audit events and logs shall be continually and centrally collected to ensure their integrity and to support timely alerting and monitoring.
- All audit events and logs shall be retained and securely archived for at least as long as the security policy requires, preferably indefinitely to support retroactive long-term analysis to either support legal action or to improve security and security monitoring.
- As necessary to support the validated legal or operational needs of tenants or customers, audit records will be sanitized to allow sharing with tenants and customers—either as a part of a security service or as needed.
- Controls must be implemented to protect the confidentiality, integrity, and availability of audit events, audit log collection, log centralization, archiving, processing, and reporting.

Security Monitoring

Security monitoring is predicated on audit logs, network security monitoring (using traffic inspection such as snort, and so on), and environmental data (see section Physical Security, above). Requirements for security monitoring include as follows:

- Security monitoring shall be a highly available and hardened service that is accessible internally or remotely in a secure manner.
- Security monitoring shall include.

- The generation of alerts based on automated recognition that a critical security event or situation has taken place or is detected.
- The delivery of critical alerts via various means in order that security and management are made aware in a timely manner.
- The means for security personnel to investigate and prosecute an unfolding incident or simply to review logs to improve alerting mechanisms or to manually identify security incidents.
- Implement a cloud-wide intrusion and anomaly detection capability and consider expressing this as a service for tenants or users (see Figure 4.2 for an overview of security event management and how it relates to security monitoring).
- Consider functionality to allow customers to implement intrusion/anomaly detection for platform-as-a-service (PaaS) or infrastructure-as-a-service (IaaS) and further to allow them to send appropriate event sets or alerts to the cloud provider's security monitoring system. (This is discussed further in the Security Monitoring section in Chapter 6.)
- Ensure that security monitoring is implemented to be reliable and correct even under circumstances of failure in the pathway of event generation and collection through reporting. Security logs must be retained in a manner that is compliant with law, applicable regulation, and the security policy.

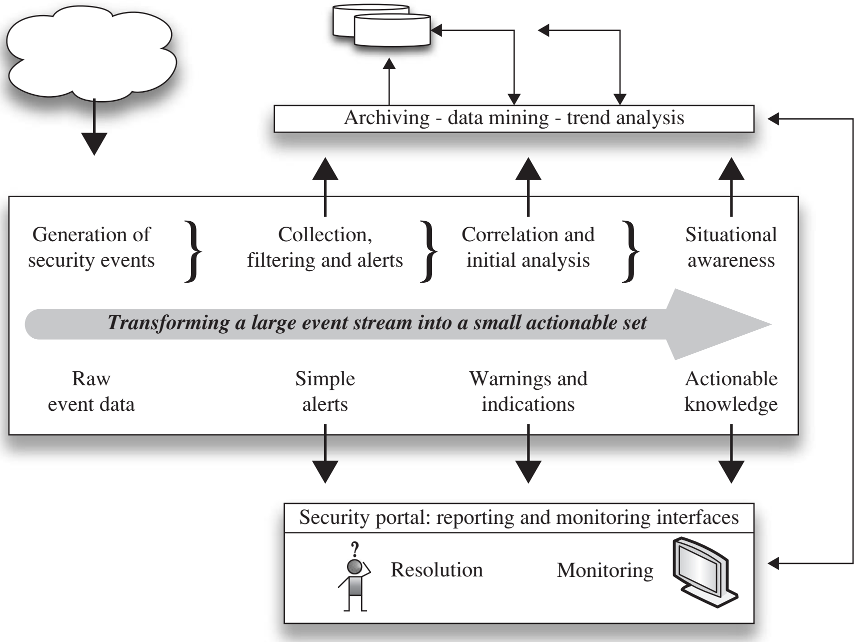
Archiving - data mining - trend analysis
Generation of security events
Collection, filtering and alerts
Correlation and initial analysis
Situational awareness
Transforming a large event stream into a small actionable set
Raw event data
Simple alerts
Warnings and indications
Actionable knowledge
Security portal: reporting and monitoring interfaces
?
Resolution
Monitoring

FIGURE 4.2

Overview of security event management and its role in monitoring.

Incident Management

Ensure that incident management and response will be inline with SLAs and the security policy:

- Ensure that incidents can reliably be managed and their impact contained. There must be a formal process in place to detect, identify, assess, and respond to incidents. This should be detailed in a standard or formal process, and it must be tested on a periodic basis.
- Ensure that incident management includes clear and reliable means for customers and tenants to report situations or events to the provider.
- The incident management process should include periodic reviews and reporting.

Security Testing and Vulnerability Remediation

Security testing shall be performed for all software before approval is granted for production. It is important to implement a vulnerability and penetration testing capability for near continual operation. To be most effective, this capability should be coordinated with monitoring and configuration management changes to prevent false alarms and incident response. Specific requirements include the following:

- Separate environments shall be used for development, testing, staging, and production of all cloud provider software and systems, including the fielding of patches into production.
- Patch management procedures must be defined for all infrastructure components, servers, storage, virtualization software, applications, and security components. Although the term patching typically refers to *live systems*, this dangerous practice can largely be avoided in cloud because of the faster allocation and provisioning mechanisms that are necessary to begin with.
- Define an integrated strategy for vulnerability remediation or compensating controls that can be used for a range of circumstances from responding to immediate or eminent threats, to less critical patching to improve the security or reliable operation of the cloud. Some vulnerabilities will come from vendor software and will require either vendor patches, a vendor-identified work around or in-house development of compensating controls. Other vulnerabilities can be introduced by the cloud provider through custom software, insecure design elements that result in security flaws or controls that are simply misconfigured. Figure 4.3 represents an example process that can be implemented to fix provider realm security flaws.

System and Network Controls

These should be implemented for infrastructure systems, systems that host customer data and applications and all networking gear. This should include all physical and virtualized components or services. Specific requirements include as follows:

- Ensure proper isolation, configuration, and security for security components.
- Implement network isolation between different functional areas in the cloud infrastructure, begin by implementing completely separate networks—including

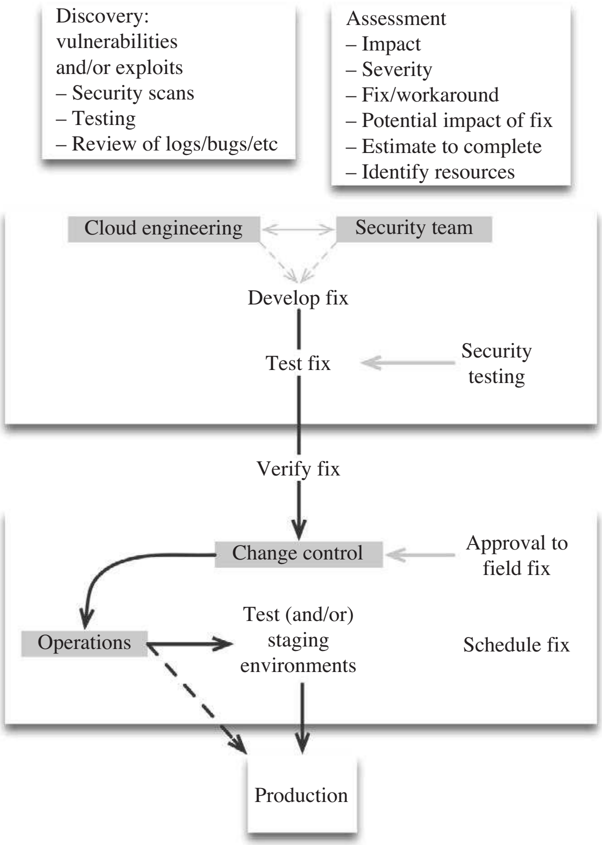

FIGURE 4.3

Security patch/fix process.

use of physical separation and network virtualization—for public accessible components (VM hosts or storage interfaces in a public cloud), infrastructure management components, and security and network administration. Reinforce this by use of other network controls and by use of software firewalls on machines.

- Hardware platform access separation from operating system (OS) (or VM) access to prevent a user with management access to the hardware from gaining access to the VM or publically accessible side. Access from the reverse (VM to platform) should also be prevented.
- These same controls should also serve to reinforce the isolation between executing VMs belonging to different customers.
- Appropriate controls will be implemented to assure the integrity of OSes, VM images, infrastructure applications, network configurations, and all customer platform software and data.

- The cloud provider shall implement the means to vet software and system upgrade releases before placing any into production. Code vulnerability checking shall be used along with malicious code scanners and other means.

TOOLS

Among several common uses, whitelisting is used in networking to identify trusted IP source addresses and in systems to identify permitted applications. As used in networking, constraining source IP addresses to those that are on a whitelist, you can effectively shun all nontrusted traffic from any other IP addresses. In a similar manner, one can constrain the allowed set of applications that can be run by use of a whitelisting product. When a whitelisted application is run by a user, the system checks the list and verifies its execution. Applications can be listed with their characteristics, including size, location, and so on, and all others will be explicitly denied. Examples of companies that are operating in this space include CoreTrace (www.coretrace.com/) with their Bouncer product and Bit9 (www.bit9.com/) with their Parity product.

The downside of using a whitelisting product is that you have to ensure that all the applications that you are going to execute on that system are authorized in the whitelisting product, and you will have to update that list whenever you upgrade any applications. (Note that such information can also be maintained by a CMDB and subsequently verified by a script or application.)

The use of whitelisting products has increased over the last couple of years and should be considered as a potential tool to be used when you deploy a cloud infrastructure, especially if you wish to minimize the number of upgrades you have to perform on an ongoing basis.

Configuration Information

With a highly dynamic cloud infrastructure and VM provisioning/deprovisioning, it is critical that one maintain a current list of all cloud assets to include hardware, systems, software, configurations, allocations, and any cloud asset that is managed or monitored in operation. Requirements for this include as follows:

- A best practice is to use a CMDB, which we will discuss further in this chapter in the section The Importance of a CMDB.
- Classify all assets and cloud components in terms of their function, sensitivity, criticality, and other characteristics that have a material impact on managing their security or understanding the security impact if they fail or are compromised.

General Infrastructure Security Requirements

In addition to the other cloud security requirements discussed in this section, there are numerous more general requirements for infrastructure security; these include as follows:

- Seek to leverage vendor and community best practices, which are the distillation of experience. If a best practice isn't applicable, we can still gain benefit from that knowledge as well.
- VMs should be hardened and minimized by default.

- Open ports should be the minimum needed for initial provisioning and allocation to a customer. When an operational process requires a port to be opened, it should be done only as needed and only for as long as needed.
- Implement the means to assure continuity of operations inline with service level agreements. Periodically verify that the recovery point objective (RPO) and recovery time objective (RTO) are reliably met.
- Ensure that network connectivity is maintained by use of multiple pathways to the cloud services. Ensure the use of diverse and redundant physical and logical network connectivity. Verify that redundant connectivity does not resolve to the same physical or logical backbone or service that is simply rebranded by a second provider. Ensure to the extent possible that physical links which enter the facility (and from there to the cloud infrastructure) are not subjected to a single point of failure under some catastrophic event.
- Ensure that the facility has ample power recovery capabilities and that power is distributed to the infrastructure in a manner that allows for redundant key infrastructure in the event that power is lost to some part of the facility. In other words, there should be ample power for the cloud provider to maintain some core capability either in support of remote continuity of operations or in support over maintaining security of the facility until it is restored to fully operational status.
- Ensure that deprovisioned internal cloud IP addresses, such as one previously assigned to a tenant for a VM, are sufficiently aged before being recycled for use by another user to prevent access by the new user to the previous user's resources.
- Expect continued innovation and changes in cloud computing and underlying technologies, and plan to modify, adapt, or extend infrastructure in ways that you may not be able to fully anticipate in advance.

SECURITY PATTERNS AND ARCHITECTURAL ELEMENTS

This section examines several patterns and elements that support or contribute to cloud security. Investing effort to develop such patterns will pay dividends during the build process, during operations and will often contribute to better security.

Defense In-depth

The term *Defense in-depth* in computer and network security was first documented in a 1996 paper *Information Warfare and Dynamic Information Defense*,[3] and was adopted from military operations. This approach has been used for system and network security under a number of names, including *layered defense*. Essentially, this is a strategy that accounts for the fact that individual security controls are typically incomplete or otherwise not sufficient, and that multiple reinforcing mechanisms or controls will compose a more complete and robust security solution. Such reinforcing controls can be similar and redundant, but can also be implemented or layered at different levels throughout the implementation. When using a series of layers consisting of even the same type of mechanism, residual risk can be significantly reduced.

From an architectural standpoint, it is wise to design for mutually reinforcing controls to increase assurance. By example, defense in-depth for access control mechanisms might first require the use of a virtual private network (VPN) (defense layer 1) for remote administrative access. Second, a VPN connection attempt may be shunned by the ingress router for any non-whitelisted source IP (defense layer 2). In this manner, only traffic for a single port (or service) is allowed to connect to an internal VPN termination point and only if the source address is whitelisted—thus, the amount of random Internet *door knocking* is greatly reduced at the edge of the infrastructure, reducing all manner of associated consequences compared to otherwise passing such traffic deeper into the network before it is identified as undesired. As a third level, the use of access control for remote administrative users could require use of a dynamically changing code that is generated by a device owned by the remote administrator. Such security tokens are used to offer greater assurance in verifying the identity of an administrative user (defense layer 3).

NOTE

The concept of defense in-depth has been around for thousands of years and was applied to castles long before it was applied to computer systems. Castles built throughout Europe during the Medieval period are a classic example of a defense in-depth. These were typically surrounded by a moat, which is a large trench usually filled with water. The castle was accessed by one of two entrances that provided a drawbridge over this moat. Attackers would find it difficult to cross the moat other than via the drawbridges as wading or swimming through water is not easy and makes anyone undertaking it a clear target for the defenders. The drawbridges could be raised or lowered acting as a double barrier. When they were raised, they covered the entrance door blocking the attackers from coming in and removing the passage way across the moat.

The entrance to the castle was usually through a small enclosed area, which was often further protected by a portcullis—a latticed gate made of metal or wood—which could be lowered down from above to block the entrance. Any attacker, should they get past the drawbridge, would have another defense to get through. The defenders would usually have additional defenses in this entrance way to further delay the attackers—often times holes where hot or burning oil or water could be poured onto the attackers or small slits in the floor where arrows could be fired down from.

Although all this was occurring, the defenders would be behind the walls of the castle, which were thick and built to withstand the attackers. These walls had numerous slits in them known as murder holes, which allowed the defenders to fire arrows at the attackers while being relatively safe. Castle walls were also built high so allowing defenders to rain arrows down.

The design of the castles had a twofold result—the castle was difficult to penetrate and could be defended by a small force. A similar principal applies when you design and implement a cloud.

Honeypots

A honey pot is a well-known and sophisticated network decoy technique. In an enterprise network, the goal of a honeypot is to create a false or nonproduction system that appears enticing for an attacker to target. After the attacker is lured to that target, the honeypot is used to observe, distract, and potentially alarm on the attacker's network penetration. In any event, the objective is that if the attacker is wasting time in the honeypot, they aren't in your production systems.

The same technique can apply to cloud computing. It can be used in network zones that are controlled by the CSP, and it can be used by tenants within zones that they control. A honeypot virtual machine can be deployed and then used to monitor and report on any attempt to access it, which would generally indicate *exploratory* snooping at the least. Honeypots could also be used by the CSP in a CSP honeypot VM for each hardware server. In this scenario, if there is a hypervisor level threat, then there is a good chance that changes are going to be made on the honeypot VM. This can serve as a form of intrusion detection at the hypervisor.

Sandboxes

Sandboxing, at the software layer, by its very definition uses a form of virtualization or abstraction between the software or code being executed from the OS in which it is running. As a result, it's very similar to hypervisor-based virtualization, running one layer up between the OS and the hardware, instead of between the OS and the application.

One of the goals of the defense in-depth model is to add layers of security. Without a doubt, a sandboxed environment adds such a layer of security between the applications running within a guest virtual machine and the hypervisor.

Network Patterns

Cloud infrastructure deviates from traditional IT infrastructure at many levels, including networking. Public clouds face several challenges in terms of ensuring sufficient network isolation between tenants, especially when VMs that are assigned to different tenants are colocated on a physical server.

Isolation of VMs

Switching infrastructure in the cloud can't isolate traffic between VMs that reside on a single hardware platform because this traffic is limited to a shared physical machine and does not enter the cloud network. Without use of encryption for this traffic, VMs could observe traffic that belongs to an adjacent VM—but the ability to do this will be a function of how the hypervisor implements networking. The use of encryption for VM network traffic can result in effective network isolation between adjacent VMs. The overall security in this case heavily depends on the security controls of each VM and on the isolation between VMs that the hypervisor affords. Thus, the security architecture patterns here are as follows:

- Select VM technology that affords network isolation between adjacent VMs.
- Encrypt communication traffic into VMs.

- Harden and tighten the security controls on VMs, especially ports and associated services.
- Filter traffic to a VM by using a software firewall or similar mechanisms to shun traffic that is not whitelisted.

Isolation of Subnets

There are other network patterns that can be followed for cloud architecture. By segregating the network into physically separate networks, you can improve isolation between public-accessed subnets and infrastructure control subnets (as depicted in Figure 4.4). Network isolation can be achieved to a point by use of network virtualization, but this is subjected to vulnerabilities and misconfiguration. Physical separation is also prone to error, but process controls can be used to minimize the probability.

Isolation really should be physically separate for administrative and operational traffic, for security and network operations traffic, for storage networks, and for public accessible components (user and tenant access to SaaS, PaaS, and IaaS). But such isolation is only effective if traffic is not routed between these separate networks. Thus, network isolation should be reinforced by additional layers of security. Firewalls are a traditional means of achieving this, and when used in conjunction with network controls, a firewall can act as an additional reinforcing layer; this is especially useful when multiple subnets would benefit from a common service, such as directory. It should be pointed out that having multiple networks to support isolation may drive up infrastructure costs, which is a point of tension between security and overall cloud costs.

Impact of Isolation Strategies on Network Device Selection

There are several cost aspects associated with multiple networks or a higher port count, the most notable ones are the cost to implement and the operational costs

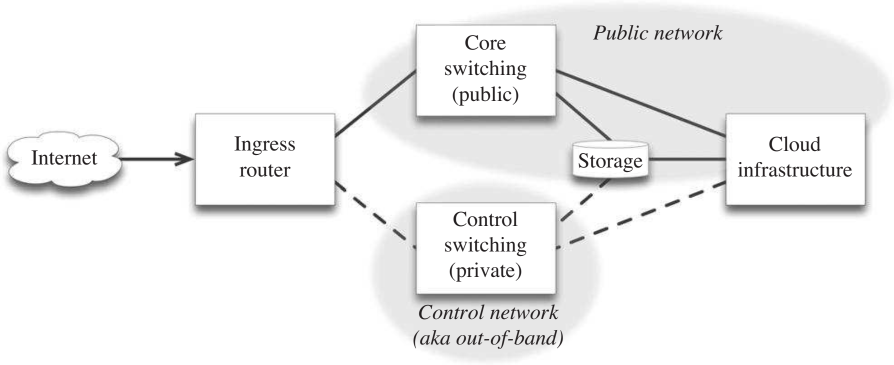

FIGURE 4.4

Basic network isolation of control and public traffic.

to manage them. Port count can be reduced through aggregation strategies, but there are trade-offs as well. For the purpose of this book, this discussion around networks and ports really comes down to several qualities that go to the security of the infrastructure.

As background, a typical rack has 42 rack units (RU), servers will require one or several RUs. Each server needs at least one Ethernet (or fiber, Infiniband, so on) port for communications. One of those ports will be for *public* traffic, and another one will likely be dedicated to directly access the hardware platform itself (power on, hardware health, so on). Servers can be extended with additional network cards to support a variety of networking strategies.

There are many possible approaches to create a networked infrastructure, but one of the points of decision is the use of in-rack switches to consolidate the traffic within a single rack of servers. In general, there are several aspects to this. First, the introduction of additional hardware introduces a potential point for failure. Although the consequence of a single switch failing may be limited to the connectivity of a single rack, there are other factors to consider. Often, in-rack switches are cabled in a left-and-right dual path manner for redundancy (by interconnecting adjacent racks). Given the number of racks that may be required to implement a cloud, this switch and traffic arrangement may experience traffic problems and even more frequent failure than the use of a centralized core switching arrangement would. Consider also the fact that since individual servers will likely have at least one public data port and one hardware platform port, the number of in-rack switches can exceed one.[B] In experience, such a networked infrastructure is not as resilient as you might expect. The number of switches is huge, and this switch sprawl must be managed at the physical and logical level. This drives up operational costs and lowers overall reliability, and it probably will result in occasional switch misconfiguration.

On the other hand, the use of a single core switch for a large number of racks will have very serious consequences if the switch fails to an extent that eclipses the consequences of a single in-rack switch failure. In this regard, consider that carrier-grade core switches are significantly more reliable than the aggregate reliability of a higher number of in-rack switches. Factoring in the cost of acquisition along with the impact of replacing switches upon failure, a core switch may be far more effective than numerous in-rack switches. Carrier-grade switches typically also have failure modes that affect fewer ports at once than in-rack aggregation switches do. Given the higher reliability of a carrier-grade core switch, the ongoing operational and reliability benefits probably outweigh the apparent benefits of a network of switches.

[B] A typical 1RU switch will have 48 data ports. The typical rack will have 42 RU for servers, assuming 46 1RU servers requiring a minimum 46 public data ports on a switch. These servers will also require 46 platform ports somewhere, but since the traffic across those ports is relatively low and the need for reliable links may be less stringent than compared to the public data ports—you might get away with daisy chaining the server platform ports in order to limit the need for the same number of in-rack switches to service platform ports.

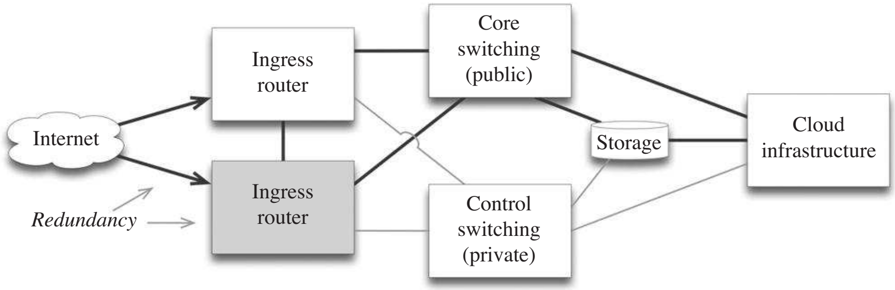

FIGURE 4.5

Redundancy to improve reliability and availability.

Availability and Redundancy

Another network pattern is the use of redundant components, load balancing, and multiple links between critical components to improve reliability and availability. Figure 4.5 depicts the use of redundant Internet drops along with redundant ingress devices. Depending on the need for availability, this pattern can be repeated but at the expense of cost, increased complexity, and higher operational overhead. For instance, a third ingress can be added for greater reliability, but given the use of carrier-grade equipment, the cost benefit is unlikely to warrant it.

The Use of Patterns

A different and more cost-effective approach would be to architect the infrastructure in repeating patterns, whereby the amount of infrastructure drives the need for the addition of another ingress—and the increased bandwidth that comes from it. At that point, the architecture resembles a series of similar blocks where each additional block expands the amount of processing and storage for the cloud. This is depicted in Figure 4.6; but by adding additional blocks of cloud computing infrastructure, we also have the opportunity to leverage identical components to improve the overall reliability of connectivity. It should be noted that the core and control switching infrastructure in both Figures 4.3 and 4.4 could be made redundant for greater reliability as well, but that topic is reserved for later in this chapter when we examine a few example architectures in greater detail.

The Importance of a CMDB

A CMDB is an information repository for managing the components of an IT system. The term comes from ITIL, where it is used to refer to the authorized configuration of components of the IT environment. CMDB implementations can include data from additional sources, such as asset management records.

A CMDB records configuration items (CIs) along with their attributes and relationships. CIs generally store information about the CI and its relationships to other CIs.

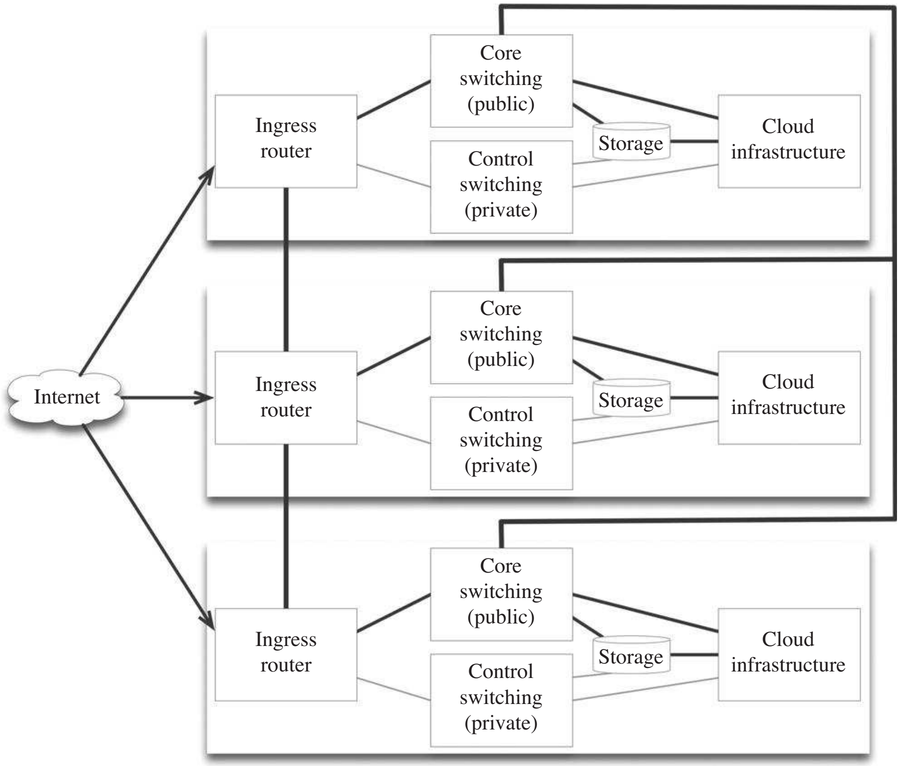

FIGURE 4.6

Regular patterns contribute to reliability and availability.

A CMDB can be used to create and manage an accurate and complete representation of the IT environment it records information about. In that regard, it is critical that the CMDB be maintained if it is to accurately reflect all infrastructure changes.

A CMDB offers tremendous advantages to the operation of a cloud. If cloud management software operates the cloud based in part on information in the CMDB, and if it updates the CMDB with relevant information as it operates the cloud—then automation is enabled to an unprecedented level for functions far beyond provisioning and deprovisioning VMs. One such area is security. The CMDB maintains contextual information about the environment that security systems are reporting on and monitoring.

To date, little work has been done in coordinating security management and monitoring in conjunction with a CMDB, but this area holds great promise for cloud security. In such a pattern, the CMDB—as an accurate model of the system state/configuration—needs to operate closer to real time than how CMDB products are generally used today. For such a cloud use case, the CMDB *reflects* the configuration of the cloud and must be tightly integrated with all the processes that change the state. Verification of the correctness of the state data can indicate process errors or malicious activity. Either case would need to generate alerts to begin resolving the cause of the differences.

The common CMDB activity of discovery would also need to be expanded to continuously verify that the CMDB has a complete and correct perspective of the IT environment. CI attributes would need to be extended as well to support security management and monitoring.

As described above, a CMDB would become a critical component for security and because of the synergy between the operational security realm and the CMDB, security monitoring could transition from alerting and reporting on the detection of security events, issues, and incidents and respond to many common situations with intelligent and contextually valid feedback mechanisms.

Cabling Patterns

Often overlooked in small system builds and many server closets, cabling patterns contribute to a faster and more reliable implementation of infrastructure. The use of the equivalent cable port for the equivalent network connectivity on each machine in a pattern may seem a trivial example, but it is worthwhile to consider the effect that this has on daily operation and during incident response. Likewise, the development of cable color standards does not only make the implementation appear more organized but also reinforces separation of networks when data center personnel perform emergency repairs at 03:00 A.M. when only *Red Bull*, *Monster*, or *Dracula* rule in peak performance.

This becomes far more critical when infrastructure is scaled. A well-designed data center will have a data cable plan that almost explains itself visually. This will reduce common errors, and it will make eventual hardware changes and upgrades faster and more reliable. Following regular cabling patterns also enables periodic physical security audits of infrastructure components. But trivial systems such as color coding will not go far enough to solve real operational problems.

The same holds true for power cables. Many modern data center servers, especially cloud friendly blade servers will have multiple power supplies and multiple power cords. Furthermore, the typical data center will deliver power to racks from at least two separate circuits; thus, it would defeat the purpose to plug both power cords for a dual supply server into the same power circuit. This level of redundancy is intended not only to overcome server power supply failure but also to overcome circuit failure.

Finally, it would be a significant improvement for even a well-designed power and data cable plan if both ends of all cables came with unique factory encodings that are both visually unique and that can be scanned by a hand-held reader. As a cable is assigned to a port or power port, the cable is scanned and the association is uploaded and recorded in the CMDB. Any subsequent inquiry or replacement can either be manually verified by checking both ends of a cable or by CMDB lookup. This use of tags on cables has tremendous benefit to both drive down operational costs and decrease errors that are associated with not being certain where the other end of a given cable terminates.

Resilience and Grace

As true for traditional implementations as it is for cloud computing and cloud services, failure should be expected. The question is How will it be handled? Individual compute resources can exhibit poor performance or failure. If the application is a critical one, then your application logic and strategy must take performance and failure risk into consideration. For cloud architecture, it is important that resource elasticity is gracefully managed for not only adding or shedding a resource but also for when a resource behaves badly or fails. How a system or component fails or responds to failure is becoming an increasingly important area as increasingly more systems are directly involved in operating devices whose failure could have serious and life-threatening consequences. Already today, many mobile apps and mission critical applications are being driven by cloud computing-based services. Where business success depends on an application, it does not matter so much where that application is powered, what matters is that is reliable and that failures are met with appropriately.

The term resilience has to do with the ability to maintain and continue to provide an acceptable level of service when a system is subjected to faults and deviations from normal operation. In this, the cloud model offers great benefit. Individual components can fail with little lasting impact. In fact, components such as disk drives, faulty memory, or even malfunctioning servers can fail and be remotely powered off. These devices can be automatically removed from the pool of available resources and left in a deactivated state until enough failed components warrant sending a cloud engineer to fix or replace them. By its repeated patterns and its scale, a cloud is a very resilient and dependable infrastructure and more fault-tolerant than traditional IT infrastructure. The ease of removing a failed resource from the pool is to the provider's benefit, whereas rapidly allocating and provisioning a new resource for the tenant is their benefit. Also, it is usually trivial for a tenant to have better control of an application architecture that is more resilient to component failures using cloud service offerings (because the virtual data center configuration is easier to specify and control).

Failing in place is a strategy that only works when you have enough resources to allow for it and only when individual resources are not critical to the operation of the cloud. Clearly, there is a difference between a large and expensive ingress router failing compared to one cloud resource such as a server failing.

Another important aspect to resilience has to do with where key infrastructure components are racked. From the standpoint of surviving a power outage or surviving critical equipment fire or water damage, it simply does not make sense to colocate your key redundant components in the same or adjacent racks.

By example, if your security requirements entail the use of multiple syslog or security event archives, a better strategy would be to separate them rather than rack them one above the other!

There are other important aspects to reliability (or in security terms *availability*), but a serious discussion of this topic area is beyond the scope of this book.

Planning for Change

As mentioned earlier in this chapter (General Infrastructure Security Requirements), cloud computing is still a young and evolving field with changes certain to both the models and the underlying technology components. Planning on the future need for change can drive how you design and implement key infrastructure components and how you organize infrastructure. Patterns that you can define should include reserving RUs in infrastructure management or security racks to allow for future expansion if there is any question that your cloud will change its mission in a manner that would require additional support.

Change can also come in the form of dramatic changes to the physical network that implements the cloud. How cable runs to individual systems are initially implemented can make it very difficult to upgrade server or storage hardware that has a greater port density than the current cable runs support. Changing the physical cables in a cable run from a core switch to individual servers can be extraordinarily difficult and at minimum risky from a disruption standpoint. One approach to minimize this is to run Ethernet from core and out-of-band (OOB) switches to terminate in patch panels above server racks and from there run patch cables to server ports. A very good strategy is to consider replacing six Ethernet cable runs from a central switch to an above rack patch panel with a single MRJ21 cable, thereby simplifying the cable plan for the network.

An important plan-for-change strategy is simply to use a Lean/Agile style of planning thought. This is more of a just-in-time way of handling growth and change. No part of the work should be done overly far in advance because it makes too much of a commitment for all the dependent equipment, cabling, networks, and power. Keeping a tight rein on incremental completion with a minimum number of advancing edges allows the next new thing to be as different as needed without ripping up (refactoring) as much of the existing work.

CLOUD SECURITY ARCHITECTURE

The first part of this chapter identified requirements for security and patterns to architect cloud security. Taking that material and composing some of those elements into representative security architectures is our goal for this section.

To some, the security of a cloud computing architecture can be summarized in one phrase: Everything in a cloud is *at scale*. Cloud providers deploy massive amounts of infrastructure to capture economies at scale, tenants and users adopt that infrastructure at scale, and some believe that the threats that occur at the cloud level are threats that may be realized at scale and by everyone in the cloud.

The cloud security space is still evolving, as is the technology used to implement clouds. It appears that the technology that powers the cloud is progressing at a rate that is faster than the technology used to secure clouds. In part, this goes far beyond any particular vendor or software and reflects on the state of systems and security in general.

TIP

Quoted from NIST 800-53[4]:

> Building more secure information systems is a multifaceted undertaking that requires

- Well-defined security requirements and security specifications;
- Well-designed and well-built information technology products;
- Sound systems/security engineering principles and practices to effectively integrate information technology products into information systems;
- State-of-the-art techniques and methods for IT product/information system assessment; and
- Comprehensive system security planning and life cycle management.

Nevertheless, the advantages of clouds are real and as such their security must be addressed. In part, the security of clouds can benefit a great deal from taking a closer look at the relative maturity of cloud computing along with some supporting work done by the Jericho Forum.

Cloud Maturity and How It Relates to Security

In the information security space, in general, the maturity of a particular technology, algorithm, piece of code, or even a process, procedure, or framework can relate, at least in part, to how secure it actually is. Stated simply is the *test of time* tried and true?

One excellent example of this principle in action is the field of cryptology. For a new algorithm to be considered *cryptographically strong*, the maturity of the algorithm is a very important contributing factor. How long an algorithm has been in the field and vetted against attacks inherently contributes to how much value it actually can provide. 3DES is a widely used encryption cipher which is an application of the Data Encryption Standard (DES) cipher algorithm, which was originally developed in the early 1970s. DES was selected as the official Federal Information Processing Standard for the United States in 1976 for governmental usage after a long vetting period. However, through its maturity, weaknesses were discovered and 3DES evolved out of addressing those weaknesses. As a result, it can be said that 3DES has had nearly four decades of testing and evolution.

Another example of this principle in action can be made in the field of software maturity. The more mature a particular piece of software is can also contribute to how secure it actually is. Open source software benefits from this principle immensely. The more widely adopted a particular project is the more peer review it receives. Because open source code is inherently public, peers can scrutinize security very quickly. Threats can be identified, tested, and then corrected in the form of patches. The group as a whole might be able to contribute these patches directly as well. This iterative process inherently makes the project more secure. Therefore, it can be generally said that the longer the project has been around and benefited from this process the better its level of security. The project, after all, through the iterative process it's more secure than when it first started. One caveat to this line of reasoning—the discovery of vulnerabilities can take decades to surface! But, the principle of maturity benefiting security is true—just don't expect each flaw and vulnerability to be vetted by the passage of time.

The same principle that applies to vetting source code for flaws and vulnerabilities also applies to architectures and processes. In the physical world, we benefit from decades of experience with house and commercial building practices. The amount of learning that has been gained from building experts examining the root causes of house fires, structural failures and human injury have collectively lead to the development and continual refinement of building and electric codes.

In the world of software and systems, we also have our *building codes*, but they are not as evolved. ISO 27001, 27002, COBIT, and numerous other compliance efforts are oriented toward making systems more reliable, more secure. Best practices for coding, for building systems, also exist. Nonetheless, large IT systems still fail for reasons that never seem to be so unique. A kind way of putting some failures is to say that the benefits they produced are greatly exceeded by the costs to implement them. Enterprise systems are especially interesting in this regard due to the potential for business crippling costs of failure. At this stage, what might work best to manage the risk of IT failures like these is to adopt an encompassing enterprise risk framework coupled with clear business objectives and a plan to address contingencies. Coupling the architectural approaches of cloud computing as a target for IT, and pursuing this with a unified and coherent plan for the entirety of the business need, and executed in a continual learning process as you build in an agile manner may produce quick results that either work or that can be refined to work more quickly than waiting for the entire waterfall to run dry.

Jericho Forum

It is worth exploring another perspective on security that is articulated by the Jericho Forum (www.opengroup.org/jericho/). This is an Open Group consortium of IT security officers that has been in existence since 2004, originally from a loose affiliation of corporate CISOs in the United Kingdom. One of the issues that Jericho Forum has articulated is deperimeterization. In their view, the corporate perimeter has eroded over the last several years due to various factors. Although the traditional model for corporate network presented the perimeter firewall at the dividing line between the inside (presumed safe) and the outside (presumed dangerous), this boundary has steadily been eroded or bypassed:

- Using a new Internet-based service exposes, a machine located inside a network to be exposed to the service on the Internet.
- Accessing an internal corporate service from outside the network requires passing that traffic through the firewall and terminating it at the service point.
- The internal-trusted environment is continually under assault from mobile computing devices (such as smart phones and laptops) that have been subverted or introduce malware into the corporate network.
- Business servers located external to the company network (that is, in a cloud).

Although organizations in the past have been worried about external threats, the erosion of the perimeter has turned the threats into internal ones. In the view of the Jericho Forum, it is necessary to identify those components that are vital or critical to the operation and ensure that those are adequately secured, whether that is from an external or an internal threat. In a fully deperimeterized environment, every component will have adequate security measures installed on it to ensure that confidentiality, integrity, and availability are assured.

Representative Commercial Cloud Architectures

Although the concept of a cloud has been around for decades in reality, cloud computing in the forms, we know today, are relatively new. For example, below are the dates at which the various types of public and private clouds, SaaS, PaaS, IaaS providers, and technologies associated with them have been in existence!

- **Amazon Web Services (Public Cloud, IaaS)**
  Arguably one of the most mature clouds, launched in July 2002 not really with an IaaS offering, more just pieces of it. It's EC2, or Elastic Compute Cloud, which is classified as an IaaS offering launched officially (non-beta) in October 2008. Many new components of this cloud are still being launched today—see *Amazon VPC*.
- **Amazon Virtual Private Cloud or VPC (Hybrid Cloud Technology)**
  Marries, an Amazon public cloud with an enterprise's private cloud, is still in beta at the time of publishing in 2010.
- **Rackspace Cloud Hosting (Public Cloud, IaaS)**
  Launched publicly in February 2008.
- **GoGrid (Public Cloud, IaaS)**
  Launched in April 2008.
- **Salesforce.com (Public Cloud, SaaS, and PaaS)**
  Although the company was launched March 1999, Salesforce's PaaS, Force.com was launched in January 2008.
- **Google Apps Engine (Public Cloud, PaaS)**
  Its first public beta was launched in April 2008. GovCloud, Google's form of Google Apps that addresses and meets government security mandates was only launched in September 2009.

- **VMware (Private Cloud Technology Provider)**
  Although the company was officially founded in 1998, VMware Server didn't exist publicly until 2001.
- **Microsoft Hyper-V (Private Cloud Technology Provider)**
  Virtualization technology created by Microsoft and deployed in Windows Server 2008, officially launched in June, 2008.

It would be fair to summarize that most modern cloud computing architectures, in their form as they exist today, are generally around 3 years old. This is a far cry from the maturation of modern architectures or common security standards.

NOTE

It's interesting to note that Amazon's *cloud* wasn't even launched originally as a cloud in their official press release[5]:

> *SEATTLE, Jul 16, 2002—Today Amazon.com (Nasdaq: AMZN) launched its first version of "Amazon.com Web Services," a platform for creating innovative Web solutions and services designed specifically for developers and web site owners.*

Representative Cloud Security Architectures

To this point in this chapter we have reviewed cloud security requirements, examined common patterns in cloud computing infrastructure, and reviewed numerous aspects of IT and architecture that relate to security. Next, we will compose some of this into a few examples of cloud security architecture.

Example 1: IaaS, Identity as a Service and DBMS as a Service

Figure 4.7 depicts a public cloud that offers several distinct services:

- Hosted DBMS
- Hosted Identity
- PaaS

Starting at the bottom of Figure 4.7, we see two distinct entry points into the cloud infrastructure. The first, on the left, is referred to as *Access to Control Network*, also referred to as the OOB network. Access to this network must be tightly limited to a subset of the cloud operations and management team. Access via the OOB access router may be limited to coming from whitelisted IP source addresses or from secured jump hosts that are outside the security perimeter. In other words, the OOB routers may simply shun (or drop) all connection attempts that are not from IP addresses that are known to be associated with legitimate operations personnel. In addition, such access must be authenticated for identity, which in the case of administrative level, operations and security personnel really needs to be fairly robust. Two-factor authentication (token card, plus pin) is necessary for several reasons that together increase the assurance that authentication is secure. This is our first in-practice example of defense-in-depth.

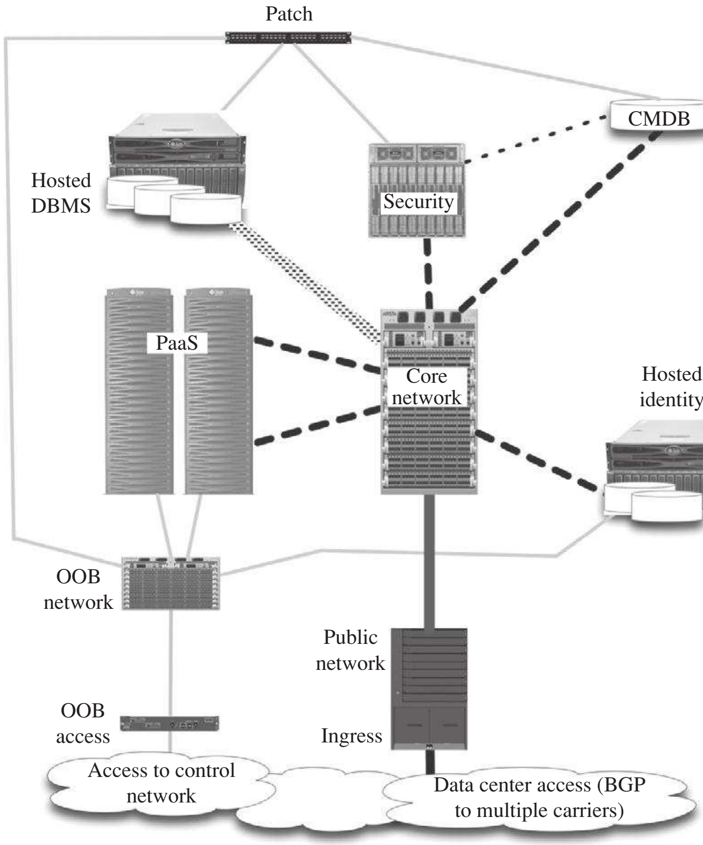

FIGURE 4.7

A hypothetical public DBMS, PaaS, and identity as a service.

The second entry into the cloud infrastructure is the ingress, which is composed of two (redundant) industrial-grade routers. This implements the entrance to the public side of the cloud infrastructure.

Figure 4.7 also depicts three primary internal networks:

- **OOB** This network offers access to the management side of the cloud infrastructure and is physically separated from the core network.
- **Core** This network is the one that all user traffic transects.
- Also shown is a network link between the DMBS and the core switch, but this (as for any link) can be implemented as an aggregated set of links for bandwidth and reliability.

Figure 4.7 also depicts the hosted DBMS which would be expressed as a service of some sort, depending on the CSP APIs. End users operating from within a remote enterprise might access this service remotely or locally via leased services in the PaaS that is depicted below the DMBS.

Likewise, a hosted identity service is depicted, which would be accessed remotely or locally via leased services in the PaaS. Also depicted are a CMDB and security services, which were previously discussed and which we will go into greater detail with the next example.

Example 2: A Storage and Compute-rich Cloud for IaaS

The second security architecture diagram we will look at is Figure 4.8, which depicts a fairly complex implementation of a hypothetical public cloud. This infrastructure has a generous amount of storage and computing resources, and it enjoys a very beefy network hardware suite for public ingress, management entry, and internal switching. It follows several patterns for high availability, and it has a dedicated pair of security stacks.

Network Entry

As in the previous architecture (Figure 4.7), in second architecture (Figure 4.8), we see two distinct entry points into the cloud infrastructure: An OOB and a public access point. In Figure 4.8, we have a redundant pair of OOB access routers

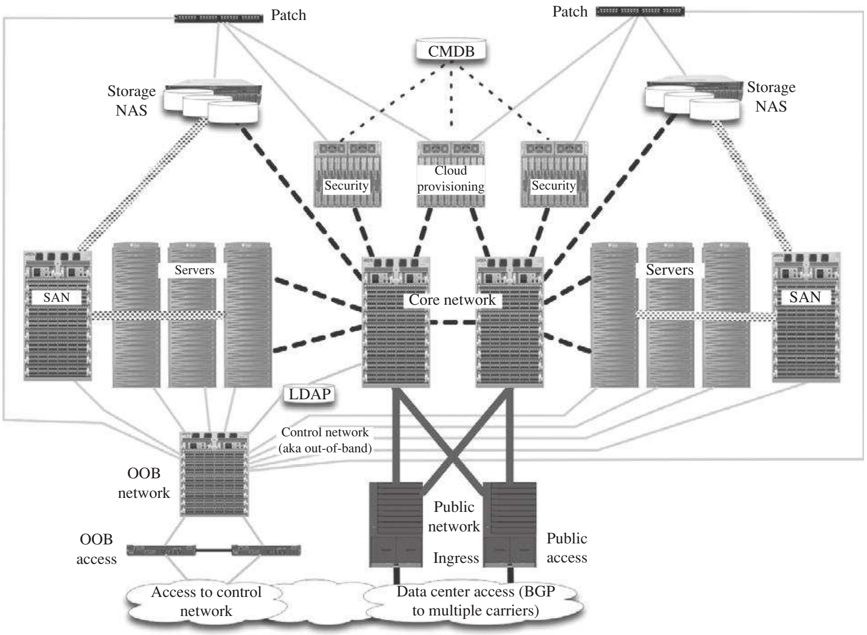

FIGURE 4.8

A hypothetical public cloud architecture.

(with sophisticated firewall capabilities). And, similar to the first architecture diagram earlier (Figure 4.7), the second entry into the cloud infrastructure is the ingress, which in this case is composed of two (redundant) industrial-grade routers. These will typically have several layers of security functionality in them, including traffic inspection, black listing capabilities, and so on. (Figure 4.8 also shows two patch panels that serve the purpose of simplifying the depiction of the OOB connectivity in the drawing.)

Where the public cloud entry is likely to be a pair of substantial and expensive carrier-grade network routers, the OOB access devices will not need to be as substantial due to several reasons. First, they can be configured to shun most traffic. Second, they will only allow a tightly constrained set of protocols (SSH for instance). Third, the amount of traffic and concurrent sessions will be very light compared to the public network. It is certainly possible to use the ingress routers to serve as the network entry for OOB traffic, as long as that traffic is immediately routed to a separate network.

Also, the OOB access won't be the primary method used to access and manage the infrastructure components. It provides the mechanisms to remotely bootstrap and initially configure some of the components, and it serves as the last-ditch mechanism to deal with disasters. Most of the normal operation and configuration of the infrastructure is done within normally secured partitions of the core network.

Separate Networks

The remainder of Figure 4.8 depicts three separate networks: The OOB, the public network, and a storage area network (SAN). As in Figure 4.7, the OOB offers access to the management side of the cloud infrastructure and is completely separated from the other two networks. However, things can be a bit more complicated. Depending on need, the OOB entry point may afford access to additional networks for administrative purposes, but this will be a function of your specific needs and your security approach as specified in your security policy. One approach is to have the management servers live on the OOB, and thereby mixing management traffic with platform service traffic. Clearly, separate networks are better from a security perspective, but they incur a cost in terms of entailing extra steps by operations when cloud personnel manage the infrastructure.

Switches

Figure 4.8 depicts three switches, two in the center serve as the core network and one on the left as the OOB network. The core switches are most efficient and minimize switch sprawl when core switches direct connect to each device, typically via Gb or faster Ethernet. Switch port density will, thus, be a critical component as to determine how many servers can be switched by a single switch. Extending this number can be achieved by various means, but doing so by repeating the pattern of n servers per core switch is very effective in practice. This drives toward a number of efficiencies, including minimizing switch configuration and management overhead. Large switches do not tend to fail at the chassis level as frequently as they do at the interface card level (a subset of the overall ports on the entire switch). In fact, large switches tend to fail less frequently than do smaller switches. So, this pair of characteristics lends itself to lights-out-operation in a large cloud. Passive backplanes in these large switches make the individual line cards fail independently (assuming redundant power). The line cards are often built as parallel chunks of switching circuitry with few shared components and where each chunk handles only a handful of ports. The failures seen in practice generally affect only one port or a small group of ports, whereas the rest of the switch continues to run unaffected.

The OOB switch is generally going to be a smaller capacity device where port aggregation via smaller switches or daisy chaining OOB service ports on platforms is a completely viable strategy given a number of factors including the lesser amount of traffic and the nature of that traffic (delays are far more tolerable than with the core network).

The OOB switch gives access to operations to control the platforms and the network devices in the infrastructure by connecting to the component management ports, service processors, and consoles. Not depicted, are several other networks that can further isolate specific traffic, such as security management or network management, from the remainder of the environment. The key is that these would not route among themselves.

Compute Servers and Storage

Figure 4.8 also depicts a pair of SAN switches along with a pair of SANs. By keeping the storage traffic OOB from the public or core data network, we can improve performance and gain advantages for security as well. Note that the computing servers and SAN are connected to three networks in a manner that does not bridge or route across these. In contrast, integrating SAN functionality into the core network can drastically reduce network costs but requires more careful handling of the storage service security and potential performance conflicts.

Also evident in Figure 4.8 are the core servers, which together with storage are largely the point of the cloud. There are numerous strategies for servers, from the standpoint for high end gear that offers performance and an upgrade path; blade servers seem to have great appeal. A point of some discussion will likely be the presence of internal disk drives on servers, but it and other hardware topics are best left for other books as the technology changes quite quickly and the trade-offs are complex. However, from the cost of ongoing operations and eventual hardware refresh, it makes sense to consider the pros and cons of different server strategies before building your *Cloud 1.0*; least you discover, your operational costs are significantly higher than those of others and you have no easy path out. Of all the infrastructure components, it is likely that your servers and storage will warrant more complete upgrades faster than other components.

CMDB and Cloud Control/Provisioning

As stated earlier in this chapter, The Importance of a CMDB, as an information repository, is critical for managing the components of an IT system. A CMDB stores key information not only for the operation of the cloud but also for managing security of the cloud.

Provisioning and cloud control software is a rapidly evolving set of capabilities that will likely continue to evolve and drive supporting changes into the very OSes and VM frameworks that are used in cloud infrastructure. It is likely that these changes will bring greater integration of security across these various levels from hardware via service consumption.

Security Servers

Figure 4.8 depicts a pair of blade chassis that are fully loaded with blades dedicated to security. The range of activities performed by these security devices will be broad and critical to the health and continued operation of the cloud. From auditing, monitoring, expressing a virtual security operations center (SOC), and security scanning, the need for computing power and bandwidth are as important as the reliability of these functions.

Chapter 6 will cover cloud security monitoring in great depth, but by way of introduction, Figure 4.9 depicts the various typical security functions that a cloud security stack might have.

Note that the security stack has a dedicated security network and that individual security functions are expressed as a combination of dedicated physical blades (syslog archive) and virtual machines. The list of functions includes:

- **Jumphost & VPN** This is a security team-only set of mechanisms, to gain access to the security network. A security engineer would enter the cloud via the core or OOB routers, which would direct the connection to the security jumphost or VPN, depending on source address.
- **Virtual SOC** This would be a series of user interfaces to monitor consoles and other security consoles for scanning, reporting, and analysis. Since the cloud is probably being operated in a lights out and largely remote manner, these security interfaces should be accessible in that manner as well. Some information may be expressed via a broader consumption dashboard that could depict outages or ongoing incidents to allow collaboration between security and other teams.
- **Collection & Analysis** This is a broad set of capabilities that starts with the collection of syslog and other security information from computing SAN and other systems and is routed via the core and OOB networks to the syslog archive. From there it is relayed to the analysis, alerting, and IDS components.
- **Directed Network Monitoring** There are further forms of monitoring that in part involve inspection of network traffic and in part involve the periodic vulnerability scanning of systems in the environment.

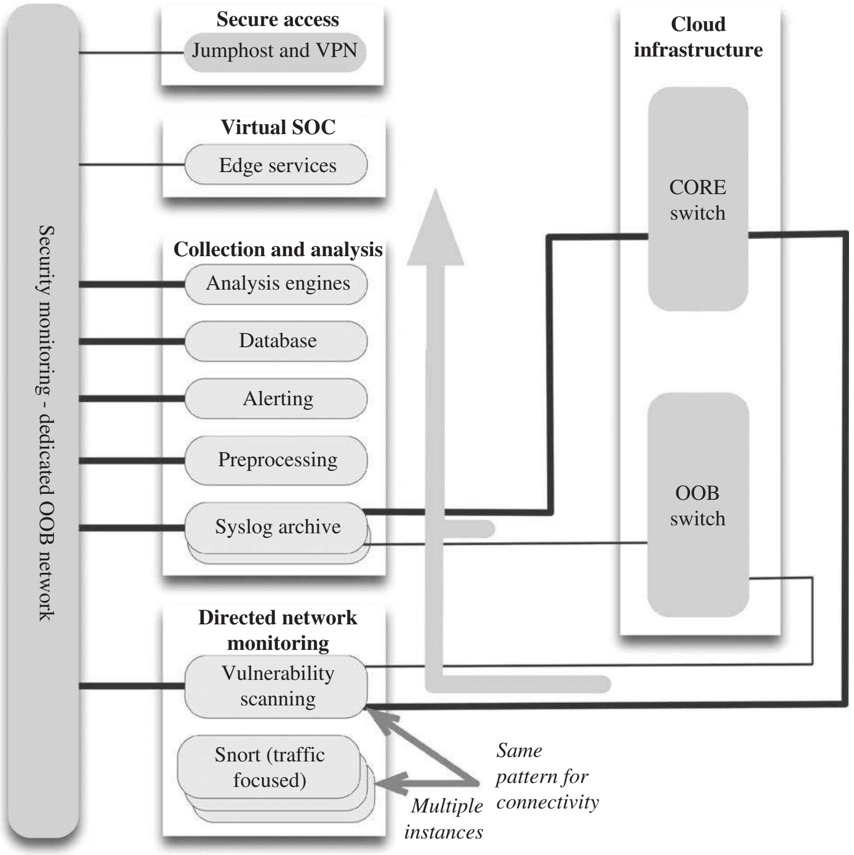

FIGURE 4.9

Overview of cloud security monitoring architecture.

PLANNING KEY STRATEGIES FOR SECURE OPERATION

The process of architecting a cloud can benefit from planning for the activities of operating the cloud. Understanding eventual operational processes and constraints can lead to better architecture and to a cloud that is more effectively operated and more secure. This section explores several areas that can offer key strategies that will pay off later in the cloud life cycle.

Classifying Data and Systems

Knowing what you have and having a formal structure for it is a great advantage when planning for how to protect it. To begin, one can identify categories of information that can be processed with lesser security concern and fewer controls than other kinds of data. That sort of information classification would lend itself to a public cloud, to hybrid cloud processing, or to a community cloud.

Various types of data bring with them the need for higher security concern, regulatory handling requirements, and even national security level processing requirements (you know who you are). National security information, be it Federal, military, or intelligence data will generally fall under the following hierarchical classification levels: Unclassified, Sensitive But Unclassified, Confidential, Secret, and Top Secret. These levels are hierarchical in terms of entailing increasing levels of security and additional handling requirements. Users are vetted before they can obtain a clearance to access data at a given classification level, and then access is generally granted on a need-to-know basis. Additional subcategories of classification can be as sedimented within a classification level and entail the need to maintain separation even from users who are cleared at the same, say Top Secret level but who have not been read into the category in question. The national security information classification scheme is very mature and quite effective in managing control over and access to classified information. However, it also tends toward overclassifying information based on the consequences of data exposure.

In the commercial world, different categories generally apply, but these tend not to be hierarchical.

If data falls under the need for PCI or other regulatory requirements, then it could still be processed in a public cloud, but the cloud provider would need to be compliant with the regulatory requirements…It is most likely that as time progresses, more cloud providers will architect for higher security and will invest in the compliance testing necessary to support managing and processing data for customers whose regulatory compliance needs could not formerly be met by the public cloud model. In a sense, the solution is more of a community cloud than a public cloud.

Define Valid Roles for Cloud Personnel and Customers

This section discusses two broad kinds of *roles*. Some define authorization classes for operational segregation, whereas the other roles define authority for policy, design, and standards. There will be several roles for internally infrastructure-focused personnel, system-focused personnel, security-focused personnel, management-focused personnel, externally service consumer-focused personnel as well as end user roles. Understanding these various roles is critical for policy, operations, and developing an effective and well-run cloud. The following list is derived from the Open Security Architecture 6, and serves as an example for such roles[6]:

- **End Users** Will require security awareness training and *access agreements*. To support users, need: Access management, access enforcement, user identification and authentication, device identification and authorization, cryptographic keys …
- **Architect** Information flow enforcement, acquisitions, information system documentation …

- **Business Manager** Responsible for risk assessment, risk assessment updates, allocation of resources …
- **IT Manager** Access control policies and procedures, supervision and review of access controls, security awareness and training policy, among many similar functions.
- **Other** Other roles include Independent Auditors, Developers, Security Administrators, Server Administartors, and Network Administrators.

SUMMARY

In this chapter, we presented a number of security requirements for cloud computing architecture. We took those requirements and for several times, we identified security patterns and architectural elements that make for better security. We then looked at a few representative cloud security architectures and discussed several important aspects of those. We ended by examining several key strategies that if considered during design can present considerable operational benefits.

In the next chapter, we will examine the broad topic of data security in the cloud. As we will see, sensitive data and control data should be encrypted for confidentiality. Network traffic to and from access points in the cloud should be encrypted for confidentiality, integrity, and ongoing availability (protection against compromise). Information and data encryption should be used for data at rest to protect confidentiality and integrity. Whether encryption of data is performed at the granularity data elements, files, directories, or volumes can be complicated by many factors including performance and functionality.

Endnotes

1. Mell P, Grance T. The NIST Definition of Cloud Computing Version 15; 2009, National Institute of Standards and Technology, Information Technology Laboratory.
2. Swanson M, Hash J, Bowen P. NIST Special Publication 800-18 "Guide for Developing Security Plans for Federal Information Systems," US Department of Commerce; 2006.
3. Winkler J, O'Shea C, Stokrp M. Information Warfare and Dynamic Information Defense, 1996, Proceedings: 1996 Command and Control Symposium, Naval Postgraduate School, Monterey CA.
4. Ross R, et al. NIST Special Publication 800-53 Revision 2, Recommended Security Controls for Federal Information Systems, Computer Security Division Information, Technology Laboratory, National Institute of Standards and Technology Gaithersburg, MD 20899-8930; 2007.
5. Amazon Web Services. http://findarticles.com/p/articles/mi_m0EIN/is_2002_July_16/ai_89075779/ [accessed 22.03.11].
6. Open Security Architecture, http://www.opensecurityarchitecture.org/ [accessed 22.03.11].