Cloud Customer Architecture for Securing Workloads on Cloud Services

April, 2017

Contents

Acknowledgements

Cloud Customer Architecture for Securing Workloads on Cloud Services is a collaborative effort that brings together diverse customer-focused experiences and perspectives into a single guide for IT and business leaders who are considering cloud adoption and necessary security measures. The following participants have provided their expertise and time to this effort: Chris Dotson (IBM), Mike Edwards (IBM), John Meegan (IBM), Nya Murray (Trac-Car), Osakpamwan Osaigbovo (IBM), Yaron Raps (IBM), Karl Scott (Satori Consulting), Wisnu Tejasukmana (Schlumberger), Puneet Thapliyal (Trusted Passage), Alexander Tumashov (Schlumberger) and William Tworek (IBM).

Executive Overview

The aim of this guide is to provide a practical reference to help IT architects and IT security professionals architect, install, and operate the information security components of solutions built using cloud services.

This guide is intended as an in-depth extension of the high level advice offered in the CSCC white paper *Security for Cloud Computing: Ten Steps to Ensure Success V2.0* [1] and the CSCC white paper *Cloud Security Standards: What to Expect & What to Negotiate V2.0* [2].

Cloud Computing Architecture for Security

Many cloud services are now available, covering infrastructure, platform and application capabilities. Building business solutions using these cloud services requires a clear understanding of the available security services, components and options, allied to a clear architecture which provides for the complete lifecycle of the solutions, covering development, deployment and operations.

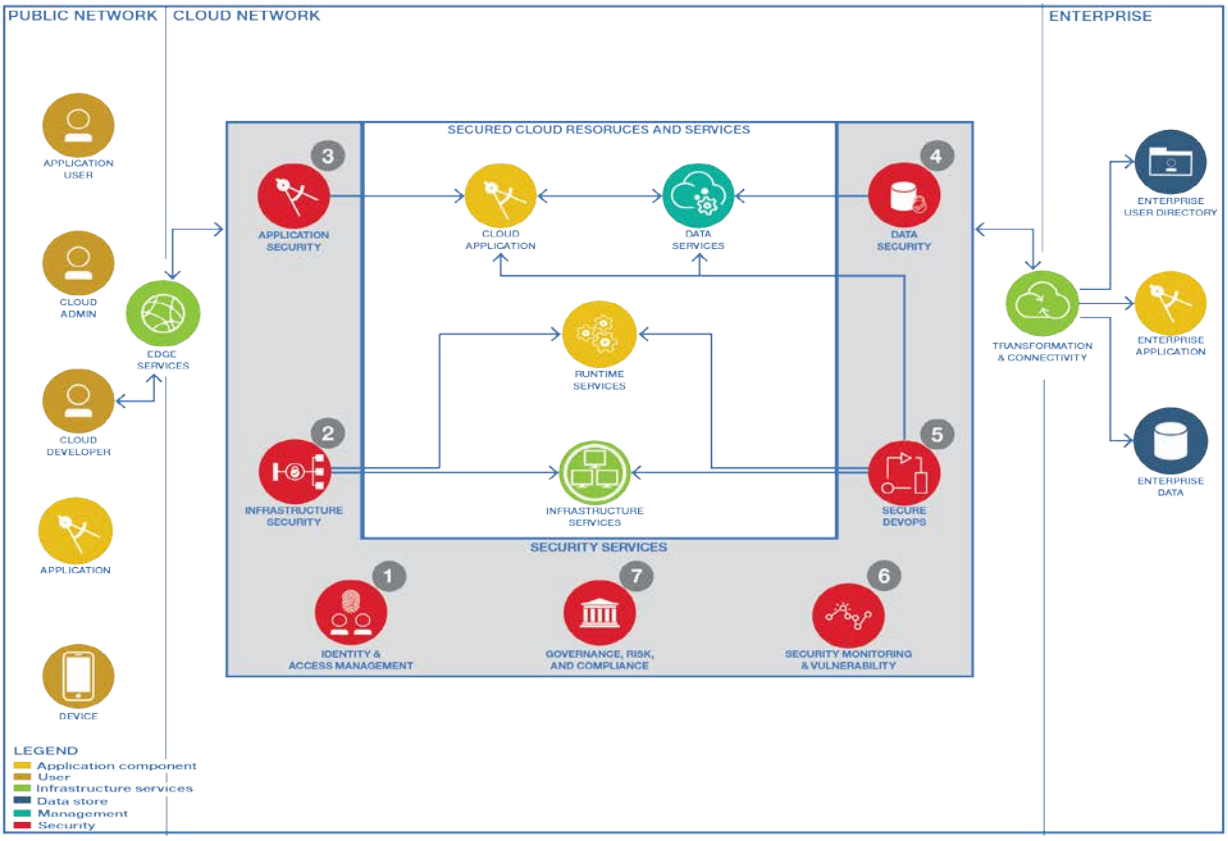

Figure 1: Architecture for Security of Cloud Service Solutions

Figure 1 provides a high level architecture for the roles and components involved in the security architecture for cloud service solutions. This architecture divides the solutions into three domains, based on the networks being used, which are usually separately secured: the public network, the cloud network, and the enterprise network.

The public network (typically the internet) contains the various parties who interact with the cloud solution, along with their end-user devices and the applications that run on those devices. Figure 1 shows three main roles: application users, cloud administrators, and cloud developers.

The enterprise network contains the existing (non-cloud) enterprise components that usually need to be used by the cloud solution – namely, the enterprise user directory, the enterprise applications and the enterprise data systems.

The cloud network contains the major components of the cloud-based solution, running in cloud services – the cloud applications, the data services, the runtime services and the infrastructure services. Associated with these components are the security services which are a focus of this document (the numbers in this list correspond to the numbers displayed in Figure 1):

1. **Identity and Access Management**
   Manage identity and access for your cloud administrators, application developers and application users.
2. **Infrastructure Security**
   Handles network security, secure connectivity and secure compute infrastructure.
3. **Application Security**
   Address application threats, security measures and vulnerabilities.
4. **Data Security**
   Discover, categorize and protect data and information assets including protection of data at rest and in transit.
5. **Secure DevOps**
   Securely acquire, develop, deploy and maintain cloud services, applications and infrastructure.
6. **Security Monitoring and Vulnerability**
   Provide visibility into cloud infrastructure, data and applications in real time and manage security incidents.
7. **Security Governance, Risk and Compliance**
   Maintain security policy, audit and compliance measures, meeting corporate policies, solution-specific regulations and governing laws.

Key Aspects of Security

The following are the key aspects that should be included when designing a secure cloud solution:

1. **Manage identity and access**: Consistent way to manage identities and access for platform and application users.
2. **Protect infrastructure, data, and application**: Ensure tenant isolation and protection for compute, data and networking. Safeguard against application and network threats, exploits, and vulnerabilities. Provide secure connectivity to data at the enterprise and protect sensitive data both in transit and at rest on the cloud.
3. **Security monitoring and intelligence**: Gain visibility into virtual infrastructures by collecting and analyzing data in real time across the various cloud components and cloud services.
4. **Optimize cloud security operations**: Optimizing the processes, methods, and tools for running security operations is key to keeping the overall cost low. Consistently assess security practices, plans, and designs and evolve them in a timely manner to stay ahead of threats.

The Impact of Cloud Service Categories on Workload Security

The broad categories of cloud services are Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). Each category has its own set of security requirements and set of security responsibilities that are split between the cloud service provider and the cloud service customer. It is necessary to understand these requirements and responsibilities when considering a solution that involves one or more cloud services.

Generally, IaaS cloud services involve the cloud service customer taking a substantial amount of responsibility for the security of data, applications, systems, and networks. The capabilities supplied by the cloud service are low-level infrastructure resources. The customer is typically responsible for the security of data both at rest and in motion and must choose appropriate encryption techniques where necessary. For applications, the customer must deploy appropriate security components such as firewalls, identity and authorization management and is also responsible for the security elements of the complete software stack used by the application.

PaaS cloud services typically involve the cloud service provider taking responsibility for security aspects of the software which makes up the platform, including the operating system and any middleware and runtimes made available for use by the customer. The customer still has the responsibility to configure the software appropriately and to choose appropriate options when deploying applications and data to the cloud service – for example, the customer might need to configure a database for encryption of the data it stores.

The cloud service provider typically takes most of the responsibility for security of SaaS cloud services, since the software, data stores and networks are usually configured and controlled entirely by the provider.

Security for Different Cloud Deployment Models

Cloud deployment models have a significant impact on security. Public cloud deployment involves the use of resources shared with other tenants and isolation is an issue that needs consideration. Private cloud deployment (whether on-premises or in a cloud service provider datacenter) involves no sharing with other organizations, although sharing may take place between different parts of the cloud service customer organization, which could require some consideration concerning access to data and applications.

Hybrid cloud deployment, particularly where it also involves solutions that connect cloud services with non-cloud enterprise systems, requires careful analysis of the security elements of each of the connected systems to ensure that the overall security of the solution meets the customer requirements and that no breaches can occur due to a mismatch between security controls applied to each of the components of the system.

Roles and Responsibilities

It is vital that roles and responsibilities relating to information security for the use of cloud services are clearly defined. The ISO/IEC 17789 standard makes it clear that information security responsibilities are split between the cloud service customer and the cloud service provider and that the split of responsibility varies between different cloud service categories and between different cloud service deployment models. The ISO/IEC 27017 standard emphasizes that the roles and responsibilities should be documented in the cloud service agreement between the customer and the provider to avoid ambiguity and to clarify relationships. Separation of duties is also a key principle to improve security.

For the cloud service customer, the information security manager has overall responsibility for the customer's IT systems, including all cloud services that are used. The information security manager works in conjunction with the cloud service administrator and the cloud service integrator. The cloud service administrator administers the security of each cloud service used (for example, managing authentication and authorization of users), monitors the service for correct operation and handles any problem reports including security incidents. The cloud service integrator is responsible for the integration of each cloud service with the systems of the cloud service customer – including any applications and datasets which are placed into the cloud service.

For the cloud service provider, the cloud service manager has overall responsibility for the cloud services used by the cloud service customer, and is responsible for performing service level management which includes information security related elements of the cloud service agreement. Operational aspects of the cloud services are delegated to the cloud service operations manager who monitors and administers the cloud services and provides audit data. The cloud service security and risk manager is responsible for managing the security and risks associated with cloud services and for ensuring compliance with cloud service provider policies including the requirements of any regulations or certifications applicable to the cloud service.

The Importance of Standards for Workload Security

A variety of security standards can help cloud service customers to achieve workload security when using cloud services. These standards are fully described in the CSCC white paper *Cloud Security Standards: What to Expect & What to Negotiate V2.0* [2].

Broadly, the security standards include:

- High-level management systems standards such as ISO/IEC 27001 and its associated cloud service specific standards ISO/IEC 27017 (for security) and ISO/IEC 27018 (for protection of personal data)
- Security standards specific to certain aspects of cloud computing including ISO/IEC 27033 (for networking), ISO/IEC 27034 (for application security), ISO/IEC 19086 (cloud service SLAs)
- Security technology standards which detail specific technologies used to implement security controls, such as OASIS KMIP (key management), FIPS 140-2 (approved cryptographic modules), OASIS SAML 2.0 (security assertions, used in IdAM implementations).

Security Architecture for Workloads

Component 1: Identity and Access Management

When creating secure, cloud-enabled workloads, enabling identity and access management is an essential first step. With identity and access management, users are authenticated and authorized, providing user-specific access to cloud resources, services, and applications. Three major elements have to be considered for the cloud environment – cloud user role, device type, and access type.

Cloud User Roles

Within cloud environments, different types of users and identities need to be managed. These include:

- Privileged access users
- Developer users
- Application users

A comprehensive security strategy encompasses the identity and access management requirements of all of these roles. The solution must be catered to a wide audience, including organizational users, Internet and social-based users, third-party business partner organizations, and vendors.

Privileged access users

Privileged access users include the following roles:

- **Application publishers, operators, and cloud administrators** require access to staging and production spaces to create, update, and delete data, applications and their service instances.

- **Managers and team leads** need insight into their team members' or employees' activities and require access to the environments that are used by developers, operators, and administrators.
- **Auditors** require access to the cloud services and the applications which run on them. This could be related to legal investigations, security incident audits or for general audits and certification purposes.

Privileged access users' accounts are very sensitive, as they are typically authorized to read sensitive information and to execute potentially destructive actions. Privileged access users' accounts also require an increased level of auditing. Attackers that are able to access such accounts can extract data from database services, deploy malicious applications, or deface or destroy existing applications.

Developer users

Developers can create, update, and delete applications. Additionally, developers can create cloud service instances and bind those instances to applications. Developer user accounts are authorized to read sensitive information and manipulate applications. They require an increased level of auditing.

Application users

Application users usually have access and control of data which can be sensitive data whose loss or modification could affect business reputation as well as lead to legal issues, and other losses.

User accounts for services (service accounts) may have access for bulk data updates on the cloud environment. In some cases, this type of account can be used to gather data from an IoT environment to a cloud service.

Device Type

Within the cloud environment, several device types might be used to access an application. Based on device type, different access rules might be required even for the same user role. These include:

Managed Device – devices that have Directory Services membership and/or are controlled by IT under compliance policies in terms of malware protection, patch management and other security controls. This can include mobile devices managed by a Mobile Device Management (MDM) solution.

Unmanaged Device – devices which include personal devices (laptops, mobile devices), printers, cameras, etc. Unmanaged devices are controlled by the cloud user and may have different access controls based on the cloud application type and user role. For instance, data download from cloud services may be disabled for these types of devices.

Access Type

Cloud services can be accessed from almost anywhere but this can be a drawback as there may be an associated loss of control. It may be desirable to limit user capabilities based on access type. There are two basic access types:

Internal access – user accesses the cloud service from an enterprise network or uses a VPN. In this case, the user can be granted full access for their role.

External access – user accesses the cloud service from the Internet. In this case, the user is granted a specific subset of functionality or alternatively additional authentication steps are required to authorize access.

Geolocation – Some countries limit access to data and applications from other countries based on local regulations. An organization needs to consider additional access controls based on government regulations related to data residency, privacy, trade and customs compliance.

Key Components of Identity and Access Management

User role, device type and access type have to be taken into account by the Identity and Access Management (IdAM) system used for cloud services. The components described below help to enforce and ensure proper IdAM for cloud services.

- **Identity lifecycle management –** management of accounts and roles.
- **Segregation of duties** – controlling access to capabilities based on user role.
- **Identity-as-a-Service (IDaaS)** – enables applications deployed to the cloud to externalize the authentication of users to a range of different identity providers.
- **Federation Services –** also known as a Single Sign-on (SSO).
- **Privileged Account Management (PAM)** – a set of additional controls for privileged access accounts.
- **Multifactor authentication (MFA) –** additional levels of authentication for higher security.
- **Mobile Device Management (MDM) –** ensure that mobile devices are compliant with corporate policies.
- **Reporting –** view of access by users.
- **Audit and compliance –** validates security controls.
- **Cloud Security Services –** tools for handling security across cloud environments, for example Cloud Access Security Broker (CASB), Cloud Security Gateway (CSG) and API Management.

Identity Lifecycle Management

Identity lifecycle management enables cloud service customers to manage user identities in cloud-based platforms, applications and services. Cloud-deployed applications can provision and de-provision user profiles with minimal human interaction. This streamlines access control based on the role, organization, and access policies defined by the cloud service customer.

Identity lifecycle management involves:

- **User Provisioning** – control resource access by role taking into account device and access type.
- **Password policies** – control minimum password length, password complexity and password expiration.

- **Access Request and Modification** – resource access requests and modification – monitored and traceable.
- **Access de-provisioning** – revoke the user permissions and disable accounts.

Segregation of duties

Segregation of duties eliminates conflicts of interest and enables detection of control failures (human errors):

- **In-Function separation** – different roles for different actions within one function.
- **Functional separation** – different functions are performed by different roles.
- **Third Party separation** – a cloud environment is managed by the cloud service provider. Capabilities available to the cloud service provider should be limited; for example, no direct access to cloud service customer data.
- **Legal Requirements** – only specific roles should access sensitive data such as personally identifiable information (PII).
- **Device and Access Type** – some roles need limitations based on device and access type.

Identity-as-a-Service (IDaaS)

Identity as a Service enables cloud deployed applications to authenticate users at an application level, based on a range of identity providers.

For example, the identity service recognizes a subset or combination of the following identity providers:

- Directory Services, including on-premises and cloud based directory services.
- Social identity providers, using OAuth 2.0 or similar protocols.
- Identity APIs.

IDaaS can store the directory on-premises or in a cloud service based on security requirements. IDaaS can limit enterprise directory service exposure by limiting attributes used for authentication.

Directory services support the identity service by hosting user profiles and associated credentials:

- User identities and group or role membership to determine access policies, rights and privileges
- Resource and service descriptions and locations

Types of directory services include:

- **LDAP based directory services** – use LDAP access protocol (e.g., Microsoft Active Directory, OpenLDAP, etc.).
- **Directory cloud services** – manage user profiles, associated credentials and password policies. A directory cloud service means that cloud applications do not need their own user repository.

Federation Services

Federation Services, also known as single sign-on (SSO), provide for seamless transition between applications – in and out of a cloud environment – without the need to have an authentication mechanism for each application. SSO can support multiple Identity Providers (IDP), directory services and IDaaS.

There are several common SSO configurations:

- Kerberos based – This method will initially prompt users for username and password and will acquire ticket-granting ticket from IDP.
- Security Assertion Markup Language (SAML) – XML-based solution that exchanges user authentication information between IDP and service provider seamless for the user based on current credentials.
- Other shared authentication schemes are OAuth, OpenID, OpenID Connect, Facebook Connect, etc.

It is recommended to use an encrypted connection between service provider and identity provider for authentication data exchange.

Privileged Account Management (PAM)

Privileged accounts are the accounts that have administrative/system (such as root, admin, sys, etc.) access to cloud resources (servers, virtual environment, databases, network devices, etc.). The need for special management measures for such accounts is necessary because of the high impact of any security breach related to these accounts, even more so where the applications and data involved are subject to special regulations (e.g. PII, health data, financial records, etc.).

PAM tools provide solutions for privileged accounts including centralized management, delegation, logging and monitoring. PAM tools provide increased security such as multi factor authentication, threat analytics, and session and password management. These tools can be used for on premises services as well as cloud services.

Multifactor authentication (MFA)

The use of multiple authentication controls can combat increasing levels of identity theft. Examples of MFA include single-use passwords, certificates, tokens, and the like. It is important to note that authentication has to come from more than one source.

To maintain the user experience while improving login security, risk-based authentication controls are used. These controls change the level of required authentication based on a user's location, past activity, operation being performed, preferences, or other factors.

Multifactor authentication is available with IDaaS, federation services and PAM tools.

Mobile Device Management (MDM)

The use of mobile devices, especially devices owned by employees ("Bring Your Own Device") introduces new security risks to the enterprise. MDM tools help to segregate corporate/personal data and applications on such devices to reduce security risks. MDM can also govern authentication to a different corporate resources by using a secure application catalog based on specific policies. An example is where functionality can be varied for the same user with the same application based on device type and a set of policies.

In addition, most mobile devices do not have device level encryption capabilities. MDM solutions can provide application level encryption to sensitive data.

Reporting

Reporting provides a user-centric view of access to resources, or a resource-centric view of access by users. The reports address:

- Which users have access to each resource
- Which access is being exploited by each user and under what conditions
- Which users have changed access rights

Cloud service providers have different types of IdAM reporting tools, which may vary by service model. The cloud service customer must check that these reports satisfy their requirements. Integration of cloud service provider reporting tools with existing on-premises reporting and incident response systems needs consideration.

Audit and Compliance

Audit and compliance is an important aspect of IdAM. Audit validates implemented security controls against the organization's security policy and regulatory compliance requirements and reports deviations. Regulations such as HIPAA and PCI/DSS are mandatory for specific industries - audit reports help an organization demonstrate compliance to these regulations. Another audit and compliance concern is e-discovery and forensics, where it is necessary to trace activities in systems and data stores.

Audit logs must exist for cloud services and need to include information such as all successful and unsuccessful authentication attempts and for privileged access to any systems or application components. There is also a requirement to track access to and modification of PII under various privacy regulations such as the EU General Data Protection Regulation (GDPR) [3].

More information about current landscape for information security standards and regulations can be found in the CSCC's whitepaper *Cloud Security Standards: What to Expect & What to Negotiate V2.0* [2].

Cloud Security Services

Cloud Access Security Brokers (CASB), Cloud Security Gateways (CSG), endpoint protection services and API security monitoring all provide control points for various aspects of security applying to cloud

services. These tools can be hosted either on-premises or as a cloud service. They consolidate enforcement of multiple types of security policies and services which can include authentication, SSO, authorization, credential mapping, device profiling, encryption, logging, alerting, malware detection and application of security analytics. These security services provide visibility, compliance, data security and threat protection to the cloud service customer. They achieve this by integrating with the APIs made available by cloud services, in order to obtain information and to control configurations and operations in those cloud services.

Component 2: Infrastructure Security

Physical security

For IaaS cloud services, physical security for that service is the sole responsibility of the cloud service provider. The provider should give assurances about physical security, such as an independent SOC 2 report or ISO 27001 certification.

Where there is on-premises infrastructure, the implementer needs good physical security practices. For example, doors with anti-tailgating measures, video surveillance, slab-to-slab barriers, and password protected consoles are common controls. Detailed implementation guidance is in ISO 27002 section 11, PCI DSS 3.2 requirement 9, and other standards.

Infrastructure isolation options

When using an IaaS cloud service, it is necessary to determine what level of isolation is required for compute, network, and storage resources.

IaaS services typically use a shared network (network security is discussed below). Similarly, IaaS services typically make use of shared storage. Encryption of data at rest can provide appropriate protection if implemented properly (please refer to the Data Security section for more information). For compute resources, there are several isolation options:

- Bare metal systems or dedicated hosts provide the most isolation, because no other tenant's workloads will be running on the same physical hardware.
- Virtual machines (VMs) provide a significant amount of isolation, because each virtual machine runs its own copy of the operating system. Although often difficult in practice, virtual machines may be subject to "side channel" attacks, where one VM may be able to affect other tenants on the same host using techniques such as rowhammer [4]. The attack surface for a hypervisor is usually very small, but virtual machines may also be subject to compromise via a VM "escape", where a user on one VM gains access to the hypervisor and total control over other VMs on that host [5].
- Containers provide a lower level of isolation. All containers use the same OS kernel, which is a larger attack surface. Although containers have many significant benefits, containers may not be advisable for deployment of sensitive workloads unless they are deployed on top of a virtual machine or bare metal system that is separated from other workloads and properly secured.

Network security

Controlled network boundaries are an important aspect of security when a customer uses a cloud service. When PaaS or SaaS services are used, the responsibility for network and perimeter security is often in the hands of the cloud service provider, and is one of the items to evaluate before choosing a service. Therefore, this section focuses on network security for IaaS and on-premises deployments.

Some important factors to consider:

- Proper network segmentation is important. Network segmentation is application specific, but some good ideas are to separate by application area (for example, front end vs. back end) and also by the type of user (for example, system administrators versus end users). Beware of having a single large flat network for all traffic, but also beware of having so many small networks that complicated firewall rules become an availability and security concern. A common pattern is a three tier approach, where one network segment talks directly to the users (such as a web server), one network segment does processing (such as an application server), and the most protected network segment holds machines that perform persistent data storage (such as a database server).
- Controls, such as firewall rules, between segmented networks are also important. Make firewall rules as narrow as possible to meet business objectives, but beware of making them so narrow that the large collection cannot be successfully audited. Review the logs periodically, particularly on internal boundaries, for misconfigurations or attackers trying to spread laterally. Also consider using an IPS or next generation firewall that can alert for malicious traffic or reconnaissance efforts.
- Use transport level security, such as TLS, in any cases where sensitive data is transmitted. Employ certificate checking. See the Data Security section for more information on encrypting data in transit.
- For Internet facing web services, consider the use of a Web Application Firewall (WAF). A WAF acts as a proxy between end users and the service and can provide an extra layer of protection to block common application attacks such as SQL injection.
- Consider your audience. If everyone who needs to access the application is on the corporate network, make it accessible only from the corporate network, either via firewall rules or a VPN tunnel. However, do not commit the “secure internal network fallacy.” Locating the services within the perimeter does not make them immune from attack. The convenience to users is often worth the additional security risk of having the service Internet facing.

Component 3: Application Security

Cloud application security plays a critical role in protecting digital assets. Effective application security requires an understanding of threats, deployment of the right security measures, and continual vulnerability management. Ineffective practices can circumvent infrastructure and data security controls and expose systems to hackers.

This section identifies the threats, security measures, and vulnerability considerations when engineering and deploying cloud application services.

Threats

Establishing a collective understanding of threats to application services is vital. Threat modeling offers insight to the development team on actions, behaviors, or conditions that affect cloud applications and potentially lead to security incidents. The development team can plan to avoid identified threats by means of secure design, coding, configuration, integration, and security testing practices.

Threat models can be simple or complex. A simple threat model is a list of common programming errors based on published information such as the *SANS25* [6] and the *OWASP Top 10* [7]. A simple threat model is used as a guide in coding, integration, and testing. Simple threat models may be sufficient when developing applications or solutions that run on secure cloud infrastructure and platforms.

A complex threat model involves:

- Identifying targets of attack (applications, systems and information assets)
- Documenting bad actors and their motivations
- Documenting actions that bad actors might take to cause those risks to be realized
- Assessing operational risks

This information is cross-referenced with known methods of attack and known software weaknesses to create guidelines for design, development, and integration. Complex threat models should be pursued for applications that process or store sensitive information.

The output of the threat modeling process should be applied in technical design, programming, systems integration and security testing. Threat models are particularly important input to operational security control validation and penetration testing.

In addition to threat modeling, monitoring should be continually performed to understand the changing threat landscape. Multiple threat intelligence sources should be used to determine active and relevant threats to information assets. Sources relevant to industry (e.g., financial, retail, legal) and technology (e.g., mobile, IoT) provide detailed insight into current threats and the methods used to compromise environments. This information should be used to ensure the appropriate countermeasures are in place to prevent system breach and the right detection and response processes are implemented to limit the impact should a breach occur.

Security Measures

Cloud applications must be engineered to address the threats identified during the threat modeling process. It's important to build security measures into the application rather than retrofit controls after deployment. Building security into the application proves much more effective in preventing attacks and reduces solution complexity. There are several frameworks such as [8], [9] and [10] that provide

guidance for building security into the software development life cycle.

This section provides guidance on security in designing, coding, and testing cloud applications. Additionally, cloud application security controls needed to protect information assets are described.

Secure Design

Architects and developers must consider security requirements and threats when designing applications and software. The goal is to ensure threats are managed and applications adhere to relevant international, national, governmental, industry, and regional security standards.

- **Analyze attack surface** – The intent is to decrease potential vulnerabilities that can be exploited by reducing the attack surface. The remaining attack surface is the focus for the security controls needed to protect assets.
- **Service and data isolation** – The approach to service and data isolation must be considered during the design phase. This is especially important when designing multi-tenant applications, which must consider isolation of services to maintain confidentiality of tenant data. Integrity of tenant services can be accomplished by proper session management, container isolation, and data segregation.
  Building in security during the design process will pay dividends throughout the life of the solution. Security testing will uncover fewer vulnerabilities and less time will be spent retrofitting solutions to reduce risk.

Secure Coding

Implementing secure coding practices reduces the likelihood of security-related design weaknesses and security defects. Developers perform secure coding by following the guidance from the design recommendations to avoid dangerous programming, software configuration, and integration errors.

The focus of secure coding includes:

- Input validation
- Output encoding
- Session management
- Credential and password handling
- Protection of sensitive data in storage and in motion
- Error handling and logging
- Protection of log information
- Selection and proper use of APIs and network services

There are general secure coding practices guidelines available to help educate developers including *OWASP Secure Coding Practices* [11] and the *CERT Top 10 Secure Coding Practices* [12].

Secure coding requires a focus on static code testing during the development process. Static code testing should be performed by developers throughout development to identify source code vulnerabilities early in the process. Static tools detect weaknesses such as buffer overflows and SQL injection. Static Application Security Testing (SAST) tools help to identify much vulnerability but there can be blind spots to these tools. Manual code reviews must complement static code testing to ensure comprehensive vulnerability analysis.

Security Testing

Effective security testing is essential to identify cloud application vulnerabilities. The testing process must focus on the relevant threats (e.g., OWASP ASVS) to ensure effective security measures are in place. Cloud application security testing emphasizes:

- **Attack surface review** – A comparison of the original (identified during design) and final (after code development process is completed) attack surface is performed to identify variances. The threat model is updated to reflect any changes in the attack surface. This information is taken into account during the testing process.
- **Fuzz Testing** – This method tests code that processes input across trusted boundaries such as web service interfaces, network sockets, and file exchange. Fuzz testing injects invalid or malformed data to discover exploitable privilege level elevation vulnerabilities. Several fuzz testing methods are used to expose buffer overruns, unhandled exceptions, and other weaknesses.
- **Web Application Scanning & Penetration Testing** – A form of dynamic application testing, this entails an external application scan (authenticated and unauthenticated) to discover vulnerabilities. This may include vulnerabilities such as cross-site scripting, insecure certificates, insecure ciphers, and remote code execution that must be addressed prior to production release. Organizations should perform web application scans periodically to identify vulnerabilities that may surface in the application environment over time.

  Periodic penetration testing is performed to verify vulnerabilities can be exploited and identify the extent of exposure. To ensure cloud applications are secure, penetration testing should include, at a minimum, OWASP Top 10 vulnerabilities.

Cloud Application Security Controls

Application security controls are applied during the development process. These controls address:

- **Cryptography** – should be used to protect sensitive data in use, in transit, and at rest. The decision to encrypt must be based on requirements and focused on maintaining the confidentiality and integrity of sensitive data. Cryptographic solutions can be rendered insecure by poor cloud application implementation. Improper key management, integrating secure ciphers, key re-use, and enabling fallback to insecure ciphers represent a few common mistakes when implementing cryptography. Reference the *Data Security* section in this document for detailed data encryption guidance.

- **Identity and Access Management** – is critical to application security. Managing logical access to applications and data to control who has access to what is essential to reducing risk. The *Identity and Access Management* section in this document provides guidance on this topic.
- **Web Application Firewall (WAF)** – protect web-based applications from known vulnerabilities such as SQL injections, buffer overflow, parameter tampering (e.g., path manipulation), and information leakage. Both inbound and outbound web server traffic is filtered and traffic is inspected to prevent attacks. WAFs can be virtual or physical and are placed on the perimeter network in addition to the network firewall. Research must be performed to ensure the application is compatible with WAF solutions.
  It is preferable to build the countermeasures into the software code rather than depend on external devices such as WAFs. Web application firewalls are most valuable when protecting legacy applications that have become a black box – limited internal knowledge or source code is unavailable.
- **API Security** – APIs are pervasive in cloud environments. It is imperative that secure development methods are adopted and validation of API security is performed during the testing process. At a minimum, the following approach should be followed.
  - Understand how the API acquires information
  - Map the API flows. Understand all methods and functionality at the start of assessment.
  - Capture the runtime traffic
  - Use automated scanners to test the API

  The common API vulnerabilities developers must be aware of include:

  - No or weak authentication
  - Weak password complexity
  - Use of non-federated access
  - Session tokens with no expiration
  - Lack of logout/session expiration

  Additional guidance on securing APIs is provided in the CSCC *Cloud Customer Architecture for API Management* [13].

- **Container Security** – Containers have gained popularity along with the deployment of microservice architectures. Applications isolated using containers share access to the operating system kernel. An understanding of container threats and available security controls is needed to manage risk associated with container deployments.

  IT organizations must begin with understanding the basics of container isolation: namespaces, control groups (cgroups), and network configuration. Namespaces define the virtual boundary for processes executed within containers while cgroups define the resources that can be consumed by the container (prevent resource starvation). Network bridging within container

environments increase risk; proper configuration is needed to reduce exposure of container network traffic.

After attending to the basics of secure container configuration, the administrator must focus on addressing specific threats, such as kernel exploits, container escapes, and cross container attacks. Identifying the right Mandatory Access Control (MAC) tool to apply security policies to containers (e.g., SMACK), performing kernel hardening, and limiting application calls to the kernel (Seccomp, "SECure COMPuting") is essential to addressing specific threats. The NCC Group whitepaper *Understanding and Hardening Linux Containers* [14] provides insight into container security.

Component 4: Data Security

Data Classification

Data security and data protection starts with clearly understanding the data and the organizational, industry and regulatory requirements for protecting it.

Data must first be classified:

1. *Public* - and thus does not need a large protection focus
2. *Proprietary* or *confidential* to either the organization or its customers - which can have related organizational rules or contractual commitments
3. *Regulated* data, such as personally identifiable information (PII), which may have special regulatory requirements

Regulated data such as PII requires a more granular understanding, as the applicable rules and regulations can vary widely based on the specifics of the type of PII as well as the jurisdiction applying to the data. Examples of private data include:

- *Personally identifiable information* (PII), such as name, address, phone number, email, etc.
- *Technically identifiable personal information*, such as geolocation data, device IDs, usage based identifiers and static IP address, when linked to an individual
- *Employment related identifiable information*, such as job history and performance review information
- *Personality related identifiable information*, such as personality insights or sentiment analysis
- *Sensitive personally identifiable information* (SPI), such as government ID, racial/ethnic origins, marital status, sexual orientation, trade union memberships and political views.
- *Financial information*, such as credit card, bank account, financial holdings and salary information.
- *Healthcare information* such as patient records, health insurance details, diagnostic or treatment information and genetic information.

- *Law enforcement information,* such as security clearances, criminal history and background check information.

Beyond PII data, there can be other types of regulatory controlled data as well, such as trade and customs regulated information, or industry and government regulated information.

The protection applied to data should reflect its nature and the impact of unauthorized disclosure, modification or loss. An insufficient level of protection can imply legal and reputational risk. Too much protection can have a negative impact on the cost and performance of the system.

Data De-Identification

One approach for handling sensitive data is to anonymize or de-identify the data. This removes all personal or other identifiers from the data, such that it cannot be reversed and re-identified - even by the original data processors or controllers. Such techniques allow the data to be used for activities such as statistical analysis or research and development activities, without the need for other special data protection or regulatory controls.

Specific details on de-identification techniques are covered by the ISO/IEC 20889 standard [15]. However, some studies on de-identified data have shown the ability to re-identify it is easier than expected, with the result that data is not as protected as expected. Given this, any decisions to use de-identification should be carefully considered.

Data Encryption

Encryption of data is a common technique used to protect data where required based on the sensitivity of the data or on the applicable organizational or regulatory requirements.

Encryption in transit

Encrypting data in transit is usually achieved via techniques such as Transport Layer Security (TLS). When using TLS, certificate checking is required, to avoid "man in the middle" attacks. When TLS is not feasible, there are typically secure options for a given protocol, such as SFTP in place of FTP, or point to point IPSEC tunnels.

Given the ease of enabling transport encryption, as well as low performance overheads, encryption in transit has become the industry norm for cloud services regardless of the sensitivity of the data. Encryption in transit when on public networks is absolutely essential. Encryption on private networks is also highly recommended, and will typically be expected by most cloud customers.

Encryption at rest

Encryption of data at rest requires consideration of where to apply the encryption, and its scope.

Where to apply encryption at rest

Encryption at rest can be applied at the operating system/storage level, the middleware level, or the

application level. OS/storage encryption typically involves using file system encryption, such as Linux Unified Key Setup. Middleware encryption involves encryption capabilities built into the middleware used by the application, such as Transparent Data Encryption in Microsoft SQL Server or DB2 Native Encryption in IBM DB2. Application encryption is encryption applied in the application code itself, versus depending on underlying middleware, operating system, or storage.

Storage level encryption has the advantage of ease of implementation, but the disadvantage of the least amount of protection. Fundamentally it only protects from physical theft of the storage hardware, as once the OS is running, the file system is mounted and accessible, unencrypted, from anything in control of the OS. Middleware level encryption can be convenient and easy to implement, but can widely vary depending on the capabilities of the underlying middleware. For example, newer NoSQL database technologies do not always support middleware level encryption. Application level encryption has the advantage of more fine-grained protection of data. Application level protection can typically be provided more selectively, as the application developer knows exactly which data is sensitive. The application level also knows about users, roles, and entitlements, and can provide access accordingly.

The disadvantage of application level protection is that debug development tools can be used as an angle of attack, to attempt to turn off encryption or gain access to encryption keys via code modifications. Additionally, developers may over engineer encryption by implementing the most complex algorithm they can find, and end up introducing security holes or flaws. When implementing application level encryption it is best to use pre-approved and industry provided cryptographic implementations. A guide on strong cryptographic methods to use is the US National Institute of Standards and Technology (NIST) Special Publication 800-175B. [16]

Scope of encryption

When encrypting data, the scope of encryption must be determined. This includes deciding what data to encrypt. This might be all data, some data, or no data. Another option is to leave the choice of what data to encrypt to a configuration decision by end users/customer. This is especially beneficial when the cloud solution is multi-tenant, and different customers may have widely varying organizational rules for encryption. With encryption as a configuration option, a default encryption baseline is needed so that less educated customers cannot accidentally leave themselves exposed.

Another scope question for encryption relates to the coverage of encryption keys. Should one encryption key be used for the whole service, different encryption keys for different data types, or even different encryptions keys based on the customer of the cloud service? Most cloud customers desire customer specific encryption keys and the option to hold or control the encryption key as well. By controlling the encryption key the customer can prevent cloud provider access to their data by removing the cloud provider's access to the encryption key. Supporting external encryption keys must be architected into any solution early on in design stages, as adding such external key support after the fact can require much reengineering in the solution.

A final scope question is if the architecture of the application requires consideration of encryption at rest outside of the datacenter, such as on the end user device. For example, if the application involves a mobile app, or other form of a thick client app, which may cache or store data locally on the client side. When client side encryption is necessary to adequately protect application data it will typically need to involve an approach of application level encryption, and thus need to be architected in very early on in the development cycle.

Key Management

The management of encryption keys is one of the most typically overlooked aspects of data protection. Key management involves the creation and deletion, secure storage, access control and auditing, and rotation of keys. If keys are not properly protected, malicious access to keys undermines the entire effort of data encryption. Key protection is now becoming a regulatory requirement, with proof of key protection required.

Many key management solutions are available, both open source and proprietary. Example open source options include Openstack's Barbican, or Hashicorp's Vault. Example proprietary options include Amazon Web Service Key Management Service or Microsoft Azure Key Vault. Essential to any key management solution is understanding the underlying storage protections, and whether a Hardware Security Module (HSM) is used. HSMs contain secure crypto processor chips and have additional protections to provide tamper evidence such as logging and alerting, and tamper resistance such as deleting keys upon tamper detection.

Key management is not just a technological decision - it also involves security policy, end user and admin training, and organizational workflows and interactions.

Data Activity Monitoring

Most regulatory rules, as well as organizational policies, require close activity logging and auditing of all data activity. This involves logging not only activity associated with access to the data, but also logging all changes or events that occur on the data. Logging must be at a granular level, with visibility to all events associated with individual data elements. Data activity monitoring techniques involve the ability to define thresholds and rules for what constitutes normal activity, and alerting if data activity exceeds the norms. Implementation of data activity logging should take into account multi-tenancy, to allow individual users and tenants to see only the activity associated with their data.

Many technologies exist to aid in the implementation of data activity monitoring, particularly for data stored in databases. Example solutions include IBM Guardium Data Activity Monitoring, and Imperva SecureSphere.

Data Access & Control

Data access and control is both a fundamental part of transparency and ownership of data, as well as a regulatory item. The majority of data privacy regulations, including the GDPR [3], require that data subjects have full access to and control of their PII at all times. They should be able to export their PII,

and request that it is deleted from the cloud service (e.g. the "right to be forgotten"). While the regulations focus on requirements for PII, many enterprise customers have the same requirements for full control of their organizational data.

When designing access to customer data, requests to delete data need to include all copies of data - even backup copies. Many customers require an industry standard method of secure deletion or destruction of data, beyond a simple delete. Guides to secure data deletion are the NIST Special Publication 800-88 [17] and the ISO/IEC 27040 standard [18].

Component 5: Secure DevOps

DevOps brings development and operations processes together with the aim of speeding the delivery of applications into production. It enables businesses to continuously develop and deploy applications and solutions using cloud computing, incorporating customer feedback and new requirements as they arise. Security must be incorporated into this approach from the first stages of development to ensure applications run on a safe platform, the code is free from vulnerabilities, and the operational risks are clearly understood.

DevOps extends the application software development lifecycle (SDLC) by taking the approach that increasing automation of promoting applications to production can mitigate risks due to disconnects between developers, quality assurance (QA), testers, and operations. Native cloud deployments and DevOps together provide a rapid response mechanism for application changes.

Embedding security in a DevOps operational framework takes advantage of the increased agility and standardization of application deployment. Secure DevOps can be viewed as an extension of application security that is designed to utilize cloud infrastructure, platform virtualization and automation. While scripted deployment of code, configuration and toolsets provides faster and more consistent software lifecycles, all the principles of application security are even more relevant, particularly determination of what must be manually tested to avoid unintentional results from increasing automation.

Security testing

Security testing for DevOps covers automated testing and automated deployment as well as manual testing. DevOps for cloud applications implies distributed resources, multiple endpoints and environments, and infrastructure with a range of visibility, depending on the public/private/hybrid network and cloud infrastructure architecture. The main change is that servers and services are virtualized across multiple compute, storage and network devices, increasing the surface area that is vulnerable to risk.

The main categories for DevOps security testing include:

- Functional testing for authentication, authorization and identity management
- Non-functional testing for known weaknesses, particularly for web exposed services
- Application and infrastructure security scanning

- Testing application logic for vulnerabilities

Security testing activities can happen at various points in the development and operations processes to ensure secure coding, configuration and deployment:

- **Source code scan**: aim to identify security issues statically.
- **Dynamic security scan:** software tools which exercise the solution in a test mode.
- **Manual code review :** Peer and formal code review of PaaS design
- **Validation testing of integrated services**: both automated and manual tests are important in DevOps

Secure deployment management

Secure deployment ensures that the application delivery processes, pipeline and the deliverables are protected from known vulnerabilities. There are an increasing number of automated DevOps specific security non-functional tests and scanning tools to ensure that code, components and packages are deployed securely and protected from attacks and common threats.

Clearly defined roles are important, as is the ability to delegate responsibilities across team members to ensure the timeliness of delivery, such as cycle times for updates, become shorter. The simplification of development pipelines, environments and the increasing automation of code updates mean that there are fewer opportunities for human-induced errors, capacity for fast rollbacks, providing that appropriate DevOps standards, processes, and frameworks are adopted.

DevOps processes are being further developed to support the adoption of continuous integration (CI), a real-time window into the actual state of the software system and associated quality measurements, allowing for immediate responses from all team members, including developers, testers, operations and QA, who are increasingly cross-skilled.

Availability and business continuity management

DevOps security includes availability and business continuity management of infrastructure, runtime components, and management components. Lack of availability of a business solution in an era of accelerated deployment cycles is a serious security problem that must be addressed by appropriate design and operation of the solution.

Some cloud services can provide out-of-the-box capabilities to enable a cloud solution to be highly available. For example, it may be possible to deploy a cloud solution in multiple physically separated data centers, with load balancing between them, thus mitigating a single point of failure. Similarly, some database services support replication and redundancy of the data they hold, again acting to ensure resilience against point failures.

In other cases, the cloud services do not automatically provide for resilience and high availability, but enable the cloud service customer to create a resilient solution with appropriate configuration and use of inherent capabilities. For example, a cloud service may be offered in multiple data centers, but it may

be the job of the customer to deploy a solution into multiple instances of the cloud service and to organize load balancing and failover between those instances.

Security evaluation and learning

Security evaluation and learning are evolving in DevOps to take advantage of the increasing automation of security testing that helps to ensure and speed up continuous integration of code. Container security is a new practice, and vulnerabilities in container engines provide a new surface for exploitation. Care is required to ensure that the security functions and properties in the delivered code and services are maintained as threats continue to evolve. Automated security tools may allow developers to feel free to concentrate on the application, however new security issues require the same focus on secure coding practices as traditional application development.

Component 6: Security Monitoring & Vulnerability Management

There are many aspects of security operations which can be covered as part of cloud. Both security monitoring and vulnerability management are notably different in the cloud environment as described in more details herein.

Security Monitoring

Security monitoring and intelligence enables an organization to proactively monitor, track, and react to security incidents. It is necessary to have end-to-end visibility and integration of security processes and tooling throughout the organization. Security monitoring and intelligence creates a complete audit history for incident management and compliance purposes, and provides reports and APIs for external consumption and integration.

It is important to note that security monitoring varies depending on the cloud deployment model and also varies with the cloud service category.

Use of cloud services increases the surface area of security risk exposure. Advanced, multi-stage attacks exist that evade detection by signature based tools. Meanwhile, DevOps and related agile initiatives have introduced faster infrastructure changes and shrinking threat detection opportunities. These trends combine to offer a substantial challenge to security monitoring and intelligence for cloud applications.

Monitoring Challenges

There are two main security challenges that we see more of with the growing adoption of cloud services:

1. When business users use cloud services without the knowledge of the IT department, they create a parallel technology stack unknown to the company called “Shadow IT.” As a result, IT teams cannot enforce corporate security policies or identify and respond to security incidents when they occur.

2. IT organizations are required to support cloud services requested by the business. IT must apply controls to extend corporate security policies to cloud services. Specifically, they need to protect data from breaches and blind subpoenas, comply with regulatory requirements, enforce governance policies, and detect compromised accounts and insider threats where the cloud operations at the infrastructure and the platform level is out of their control.

Applying traditional security monitoring to cloud services is not effective, since cloud services are fluid, ever changing and moving fast. The recommended approach tightly integrates security monitoring and security analytics. The combination of monitoring and analytics with universal data ingestion enables early detection, rapid investigation and intelligent remediation of threats across heterogeneous on-premises and cloud infrastructure. It provides better assurance that the organization is ready to detect threats and anomalies related to cloud services.

Monitoring Implementation and Technologies Considerations

There are three implementations of cloud security monitoring:

1. Implementations that modify End User Computing (EUC) devices such as tablets, smartphones, and laptops can be accomplished through use of a proxy, through installing software, or through a mobile device manager (MDM) or mobile content manager (MCM).
2. Implementations that require use of a gateway device intercept all traffic and either analyzes it to determine which cloud services are in use or go a step further and also track how those services are used.
3. Implementations that use transparent gateways require no code within the datacenter or on EUC devices. Instead, changes are made either within the cloud service or by the cloud service provider.

The technologies in use for cloud security monitoring range from those that are limited to web content to those that monitor all content using a hook into the cloud service. Technologies that are implemented using proxies or reverse proxies are limited in scope to those applications for which the proxy works. Writing a general reverse proxy is very difficult, so they are usually application specific. However, using a transparent gateway requires either that the cloud service is modified in some fashion, or that it already has the necessary hooks. The most common hook to use is single sign-on which allows an organization to use its own IdAM system. This is useful for restricting identity to known individuals and having this identity completely contained. Single sign-on can also be used to redirect access through a transparent gateway.

If the feature does not exist, all data is made to go through the transparent gateway either by modifying the network DNS or by using an implementation that modifies the EUC device. Once the data traverses the transparent gateway, data is collected for later analytics.

Security Analytics

Many of the existing monitoring tools and the transparent gateways collecting data look at the data in new ways. They aim to identify what is normal across logins versus what is abnormal and could be a security issue. Some of the issues currently tracked include, “Did the user suddenly start using a new function within a SaaS?” “Is data being exfiltrated?” and “Are they logging in from wildly different locations?” This data can be sent to security information and event management (SIEM) and reports run there. The goal is to know who did what, when, where, how, and hopefully why.

Security intelligence uses analytics to detect deviations from regular patterns, uncover changes in network traffic and find activities that exceed defined levels. Within a security intelligence infrastructure, analytics are applied to massive amounts of information in an effort to understand company data within context and prioritize day-to-day activities. By determining which deviations are meaningful, security intelligence can help detect compromises faster and also reduce false positives to save time and resources.

Cognitive Security

The latest trend for security monitoring is cognitive security, which is the implementation of two related capabilities:

1. The use of cognitive systems to analyze security trends and distill enormous volumes of structured and unstructured data into information, and then into actionable insight to enable continuous security and business improvement.
2. The use of automated, data-driven security technologies, techniques and processes that support cognitive systems having the highest level of context and accuracy.

Vulnerability Management

One of the most important aspects of cloud security is vulnerability management. Cloud is filled with security risks, which are constantly evolving as cloud adoption is expanding. Cloud empowers end-users and developers more than ever, allowing them to continuously integrate and deploy applications to /from a cloud by using multiple APIs (both private and public).

Effective vulnerability management requires a focus on the following items:

- Subscribe to Common Vulnerability Exposure (CVE) lists
- Analyze CVE data to identify and prioritize relevant vulnerabilities
- Develop a plan to remediate vulnerabilities in a timely manner
- Test to verify vulnerabilities have been remediated

Vulnerability management is a multi-phase process of identifying, classifying, remediating, and mitigating vulnerabilities in the cloud environment. But since clouds can be multi-tenant, more than one application is shared across a single cloud, some of which are outside an organization’s control. In this

case, performing a hosting environment scan is not enough. Multi-level scans are recommended to be performed such as network, agent, and network traffic.

Phase 1 - Establish Policies

First a specific cloud usage policy must be established. The easiest starting point is to assume cloud is like any other hosting environment; these policies should define what assessment techniques or methods will be used for identifying vulnerabilities.

Once these policies are defined, ensure they are relevant to your organization, industry, and external requirements and compliance obligation.

Phase 2 - Scan to Identify Vulnerabilities

Once the policies are in place and enforced, the next phase is to identify the vulnerabilities by using scanning techniques starting with the environment and then move onto other aspects like devices, databases, applications, users, etc. The ideal scanning technique is to use multiple and different types of scanners:

1. *Network Scanner*: Similar to what's done traditionally; you scan your network for vulnerabilities.
2. *Agent Scanner*: Use this scanner for devices that do not allow the traditional network based scanning for vulnerability assessment.
3. *Traffic Analyzer*: Use tools to sniff packet content and identify vulnerabilities in traffic.
4. *Code Security Scanner*: Examine source code to detect and report weaknesses that can lead to security vulnerabilities. These tools are one of the last lines of defense to eliminate software vulnerabilities during development or after deployment. These tools abilities and vulnerability libraries are specific to the development environment used.

Cloud service providers should perform the following activities based on their cloud delivery model:

1. *SaaS Model*: Scan operating systems, hardware, network infrastructure, access management applications, instance resources, upgrades, patches, and code.
2. *PaaS Model*: Scan operating systems, hardware, network infrastructure, and instance resources.
3. *IaaS Model*: Scan the entire infrastructure including operating systems, hardware, network, and virtual machines.

The most common vulnerabilities found in cloud are related to insecure cryptography, virtual machine escape, data protection and portability, session riding and hijacking, vendor lock-in, and internet dependency.

Phase 3 - Prioritize the Vulnerabilities

Once vulnerabilities have been identified the next logical step is to prioritize them; first assess the seriousness of each threat and then assign priorities. Some threats can affect core business operations

while others will have minimal impact (in cases where PCI information can be exploited, this vulnerability should have highest priority).

Phase 4 - Mitigate Vulnerabilities

Phase 4 addresses the vulnerability mitigation in accordance to the priority set in phase 3. Take up vulnerabilities with highest priority first.

Phase 5 - Maintain & Monitor Vulnerabilities

This is an ongoing process and not a "set and forget" activity. Even if the vulnerability is patched, it is still needed to ensure that it will not reemerge. Cloud customers need to constantly ensure their scans are up to date with new regulatory and compliance concerns.

Continual monitoring of application and infrastructure vulnerabilities is critical to maintaining security. The scope ranges from monitoring vulnerabilities associated with open source libraries included in the code base to operating system and database weaknesses to security flaws in virtual infrastructure. Vulnerability scanning should be performed periodically to identify weaknesses introduced as a result of changes in the environment. This process provides additional assurance that elevated application and infrastructure risk does not exist.

Component 7: Security Governance, Risk, and Compliance

When using cloud services, it is vital to apply appropriate governance to the use of those cloud services, to identify and mitigate risk, and to ensure compliance both with external laws and regulations and with organizational security policy.

Security Policy and Governance

An organization's security policy plays a pivotal role in determining how the organization's IT systems achieve security goals. When adopting cloud services, the security policy must extend to include cloud service security policies, that take into account the different environment involved in using cloud services.

Cloud service security policy needs to take account of the three major categories of cloud service – IaaS, PaaS and SaaS. This is because the level of responsibility varies with each of these categories of cloud service. In an IaaS model, the cloud service only handles physical security and basic aspects of core infrastructure services, while the customer is responsible for everything else (operating system security, middleware security, application security, etc.). On the opposite side of things in a SaaS model, the cloud service is responsible for security all the way up through the application layers, but the customer still has responsibilities in terms of data security and access control.

Cloud security policy is derived from the organization's overall IT security policy and it is worth considering how the IT security policy fits into the organization's security framework.

Security framework and IT security policy

A typical security framework for an organization is shown in Figure 3.

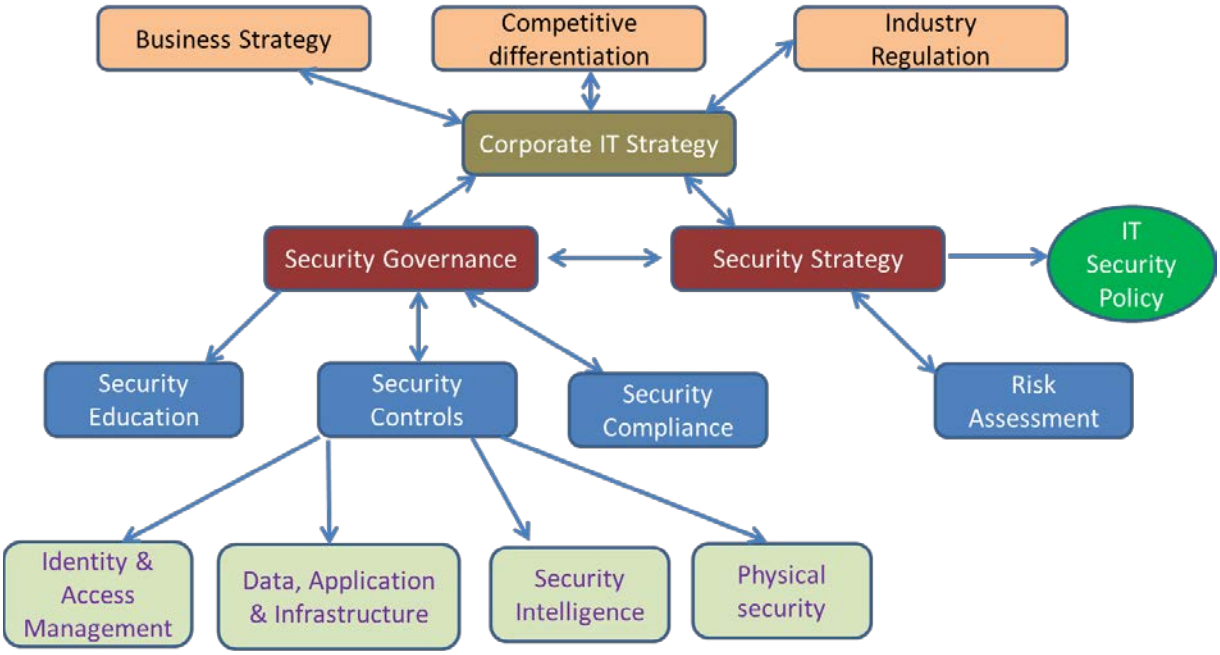

Figure 3: Organizational security framework

The business strategy, competitive differentiation, and industry regulation guidelines are prominent factors that shape a corporate IT strategy. The security strategy drives security governance.

Security governance ensures that the company:

- Enforces the IT security policy through security controls.
- Educates employees and end users about security guidelines.
- Meets industry and compliance regulations.
- Achieves operational efficiency across security controls.
- Continually assesses risks and addresses them through security controls.
- Ensures that security capabilities of suppliers and sub-processors involved in the solution is also taken into consideration.

The security controls are split across various layers of security, including identity and access management, data, applications, network and server infrastructure, physical security, and security intelligence.

When an organization adopts cloud services, the cloud security policy has to be updated to define the required security controls for extending the IT security policy onto cloud-based systems.

Aligning the Security Policies of Customer and Provider

Typically, the cloud service customer and the cloud service provider belong to different organizations. Therefore, each entity adheres to a different IT or cloud security policy that aligns with their organization's security strategy.

The cloud service should adhere to a corporate security policy and also expose security functionality that cloud service customers can use to cater to their organizational security requirements.

It is important to cloud service customers that the cloud service provider provides sufficient information about their cloud security policy and the security controls which are applied to the cloud services. Only with this information can the customers ensure that their use of the cloud services is able to meet the requirements of the customer's cloud security policy.

Risk and Risk Assessment

A key element of security governance is appropriate treatment of risk. This requires that a risk assessment is performed on the cloud service customer's use of cloud services. This in turn requires adequate information from the cloud service provider about their risk management policies and the security controls which are applied to cloud services.

Compliance

Compliance is another key factor in security governance. There is a need to ensure that both the cloud service customer and the cloud service provider comply with laws, regulations and organizational policies. For the customer, this may be achieved by appropriate periodic audits. For the provider, this is often handled through certifications to relevant standards, which take place regularly and which result in public certification notices that can be used by cloud service customers as assurance that the provider complies.

Keys to Success when Implementing Security Architecture

Key to Success	Summary
Manage access to cloud applications and resources	• Id and Access Management for use of cloud services and for the applications and resources within those cloud services • Seamless IdAM systems covering cloud services and on-premises systems
Protect and secure cloud applications, data and infrastructure	• Ensure boundary controls are in place for all assets related to cloud services, such as Firewalls, VPNs • Encryption of sensitive data • Encrypted communications
Gain visibility into all resources on cloud services	• Ensure monitoring of cloud services and the applications and data that are located on cloud services • Integrate cloud service monitoring with monitoring of on-premises resources
Incorporate security into DevOps for cloud services	• Include “Secure by Design” and “Data Protection by Design” principles into all applications destined to run on cloud services • Include security elements into DevOps processes and test security elements before and during production deployment
Strong security policy and governance	• Build a comprehensive security policy for all cloud services • Ensure compliance with all corporate, industry and government requirements and regulations • Enforce security policy through measurable security controls • Check cloud service provider compliance through certifications
Automation of security services	• High levels of automation of security services into centrally reusable services best support standardization and consistency of security

Works Cited

[1] Cloud Standards Customer Council 2015, *Security for Cloud Computing: 10 Steps to Ensure Success V2.0.* http://www.cloud-council.org/deliverables/security-for-cloud-computing-10-steps-to-ensure-success.htm

[2] Cloud Standards Customer Council 2016. *Cloud Security Standards: What to Expect & What to Negotiate V2.0.* http://www.cloud-council.org/deliverables/cloud-security-standards-what-to-expect-and-what-to-negotiate.htm

[3] General Data Protection Regulation (GDPR). http://www.eugdpr.org/

[4] Google Project Zero. *Exploiting the DRAM rowhammer bug to gain kernel privileges.* https://googleprojectzero.blogspot.com/2015/03/exploiting-dram-rowhammer-bug-to-gain.html

[5] CVE Details. *Race condition in Xen allows guest OS administrators to gain privileges.* https://www.cvedetails.com/cve/CVE-2016-9381/

[6] SANS Institute: *CWE/SANS Top 25 Most Dangerous Software Errors*: https://www.sans.org/top25 software-errors/

[7] The Open Web Application Security Project ("OWASP"). *OWASP Top 10 Application Vulnerabilities*: https://www.owasp.org/index.php/Top_10_2013-Top_10

[8] Microsoft Security Development Lifecycle. https://www.microsoft.com/en-us/sdl/default.aspx

[9] OWASP Secure Software Development Lifecycle. https://www.owasp.org/index.php/Secure_SDLC_Cheat_Sheet

[10] NIST Security Considerations in the System Development Life Cycle. http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-64r2.pdf

[11] The Open Web Application Security Project ("OWASP"). *OWASP Secure Coding Practices*: https://www.owasp.org/images/0/08/OWASP_SCP_Quick_Reference_Guide_v2.pdf

[12] Software Engineering Institute, Carnegie Mellon University. *SEI CERT Top 10 Secure Coding Practices*: https://www.securecoding.cert.org/confluence/display/seccode/Top+10+Secure+Coding+Practices

[13] Cloud Standards Customer Council 2017, *Cloud Customer Architecture for API Management.* http://www.cloud-council.org/deliverables/cloud-customer-architecture-for-api-management.htm

[14] NCC Group whitepaper *Understanding and Hardening Linux Containers.* https://www.nccgroup.trust/us/our-research/understanding-and-hardening-linux-containers

[15] ISO/IEC 20889. https://www.iso.org/standard/69373.html

[16] NIST Special Publication 800-175B. http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-175B.pdf

[17] NIST Special Publication 800-88. http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-88r1.pdf

[18] ISO/IEC 27040. https://www.iso.org/standard/44404.html

Additional References

National Institute for Standards and Technology (2011): *NIST Cloud Computing Standards Roadmap.* http://www.nist.gov/itl/cloud/upload/NIST_SP-500-291_Version-2_2013_June18_FINAL.pdf

