Cloud Application Security Architecture Overview

Author: Nya Murray July 2017

Content

Overview

Purpose

The purpose is to provide an overview of the configuration of cloud application security components across cloud infrastructure, comprising software, hosting and network.

This report is intended to provide a comprehensive end-to-end view of cloud application security configuration comprising web applications for mobile devices and PCs to application services and microservices deployed to public clouds.

The report considers cloud applications in the context of public cloud SaaS (Software as a Service), PaaS (Platform as a Service) and IaaS (Infrastructure as a Service) software, hosting and network security measures.

This assessment takes a view of cloud application security in the context of

> **External Connections:** Application IP network connections to external systems, e.g. identity verification systems, cloud services
>
> **Enterprise Connections:** Application IP network connections to enterprise data center deployments via public and private IPSec and SSL VPN connections, as well as the Internet via application gateways.

Microsoft Azure, Amazon AWS, and Google Cloud services are considered in this review, although the information pertains equally to IBM Bluemix, Salesforce and Oracle Cloud.

This overview is taken within the security context that enterprise technology infrastructure is increasingly targeted towards hybrid cloud deployments. (Hybrid cloud uses a mix of on-premises, private cloud and third-party, public cloud services with orchestration between the environments).

Objectives

Security technology is changing in response to the fast pace of attacks.

The primary objectives for this report are

- Provide a view of application end-to-end security architecture configuration
- Provide a reference for security configuration.

At runtime, the most critical parts of the overall component security architecture, are:

- Web and mobile device user interaction
- Cloud application services
- Data services
- SaaS services such as identity management and application containers
- PaaS and IaaS services and how the configuration of these components can be demonstrated to meet enterprise security requirements.

In development, application security configuration comprises

- User interaction design and deployment
- Application services and microservices design and deployment

- Container design and deployment
- Cloud Provider SaaS security design
- Cloud Provider PaaS security design
- Cloud Provider IaaS security design

Configuration information for cloud applications is expected to be updated on an ongoing basis because of architecture decisions, detailed design changes and updates to security configuration resulting from technology upgrades.

Approach

The use of a reference security architecture such as the Cloud Customer Architecture For Securing Workloads On Cloud Services can provide an outline for a security analysis framework, and effectively provides a categoric approach to information technology security.

This approach provides a view of security classified by capability:

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

Architecture and Design Resources

To successfully track application security, artefacts are required to trace security to business requirements, standards, regulations and legislation.

Resource characteristics required for successful systems deployment are essentially to be change enabled, so that technology decisions and program changes are reflected in the application security artefacts, such as documents, code tracking, models and workflows.

Cloud hosted applications, as for enterprise hosted applications, requires definition of the following essential enterprise artefacts to manage enterprise security architecture:

1. **Architecture and Design** (comprehensive solution specifications, configuration information as well as implementation and deployment checklists. Logical information and design models such as entity relationships, UML structural and behavioral models are required to support high and low level design of solution components)

2. **Services** (well-reported, schedule-automated, consolidation processes that are easily understood, repeated and tested) have to be clearly documented with a traceability matrix to requirements.

Scope

The scope of this report is to provide an overview of current security capability for user interfaces, cloud applications, and containerized microservices in the context of security of services offered by public cloud providers software platform, managed hosting and network services.

In Scope

Hosting and network infrastructure comprise most cloud application security defenses. The configuration of SaaS and PaaS services is critical, particularly in the context of DevOps, and containerized microservices.

This security architecture covers the following components:

- Cloud hosted native mobile, web browser and server applications
- Infrastructure architecture and design, hosting and network services
- Transport layer connections to internal and external services
- Containers and container orchestration
- SaaS (e.g. Identity Management, identity data and key store services)
- PaaS (API Management, Notification, Cache, and Data Services)
- IaaS (traffic managers, API gateways, virtual networks, virtual hosting, and OSI Network, Transport and Application Layer security)

Out of Scope

Specific cloud application design, code development and deployment, and CI/CD script configuration are not covered in this report. It is presumed that specific cloud application deployments, as well as code and script developments are documented elsewhere, in accord with best development practice and guidelines such as those pertaining to the appropriate specific topic as provided by OWASP.

Governance

Managing security capabilities helps to mitigate the risks involved in interpretation of corporate security requirements, and industry standards such as the ISO security standards.

It is important to have an enterprise governance framework in place in addition to enterprise information security requirements. Aligning the security of delivered

functionality with enterprise security requirements and SLAs can be subject to interpretation. In a changing security landscape, with a clear security architecture framework, objectivity is more readily available

A framework makes it easier to apply knowledge of industry security best practice, which is currently a large and complex field.

An information security architecture reference aligned with enterprise business requirements can be based on an industry standard framework, to provide a compliance mechanism for external suppliers.

Risk Management

There are a number of security risk analysis frameworks, such as the National Institute of Standards and Technology's (NIST) Risk Management Framework (RMF). This framework is useful as it refers to associated standards and practices in context.

Because of the changes to the security breaches, care has to be exercised in adopting a threat based framework, as the threats are changing. Vulnerability analysis is the first pass for determining where to apply security resources, methods, tests, and event management measures most effectively. (For example, currently most endpoint protection software looks for vulnerabilities are to be found in the use of HTTP REST methods. With the adoption of containers, the focus extends to the use of Linux commands, and the emerging information leakage channels that can compromise physical and virtual machines which share a host Linux OS kernel).

Application risk management has to encompass any interacting systems, including enterprise applications, internal and external cloud services, identity management, data services and databases. A risk matrix can provide the basis for determining where security is most needed and most effective in addressing known vulnerabilities, for example application layer DDOS, interception of HTTP services and container leakage.

GDPR Compliance

GDPR EU legislation is applicable to organizations either processing personal data in the EU, or relating to EU citizens. While prosecutions of organizations outside the EU jurisdiction are unlikely, non-compliant organizations may find it more difficult to do business in Europe.

It is advisable to implement a specific enterprise wide GDPR review process. This process would also be of benefit to external suppliers, as GDPR implementation date of 25th May 2018 approaches.

The GDPR regime includes more stringent definition for clear affirmative consent to use of personal data and encodes 'the right to be forgotten'.

Personally Identifiable Information

The following definition of PII is taken from the recent Customer Cloud Council cloud services security publication 'Cloud Customer Architecture for Securing Workloads on Cloud Services'.

- 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.

Steps to ensure GDPR compliance

1. Review all data collections holding PII information covered by GDPR, including cross border data transfer and international storage.
2. Ensure all data collection activities inform end users about data usage, data privacy, and update consent to collect data appropriately
3. Plan for making individuals' data accessible on request
4. Ensure functionality is available to accede to personal data requests for private data held, and to purge data on request.
5. Review currency of compliance with EU Article 29 Working Party Opinions and Recommendations
6. Implement security logging, monitoring and incident management processes, technologies and workflows for suspected data breaches

Development and deployment of comprehensive audit trails for tracking data access, including privileged user, developer user, and application user data access by role, service and accounting information is essential.

As the legislation is non-technical, it is prudent to be able to demonstrate the measures put in place for compliance. While failsafe security is currently not possible, being able to demonstrate best practice and strong security measures is a prudent way to show that GDPR is taken very seriously.

Identity Management Configuration

Identity Management manages identity and access for administrators, application developers and application users.

Basic roles are

- Privileged access users
- Developer users
- Application users

Application Identity Data

The design analysis and rationale for the identity management solution applies to a single identity domain for shared user definitions and authentication methods.

The main identity management data to be stored for application user authentication and authorization include

- Roles and permissions
- User accounts and profiles
- Short-term and Long-Term Identity tokens
- Secrets and keys
- Identity domain directories
- Access control lists (ACLs)

All major public clouds have the same concepts, however the implementation details are different from provider to provider.

Role Based Access Control

Cloud resources are defined to allow access to users, applications, or services.

Azure

Azure Role-Based Access Control (RBAC) enables fine-grained access management of user roles to access Azure resources. An Azure resource can be any service, infrastructure, platform, software application, database or user account.

Access is granted or denied based on specific user configuration information. Azure Role-Based Access Control (RBAC) is natively integrated into the management platform.

Azure logging is integrated via Azure Operational Management System (OMS) dashboard and forwarding services.

AWS

Role-Based Access Control in AWS refers to managing user identities, and identity claims put forward in an AWS identity token. Identity claims can be either internally managed roles or externally issued identity tokens such as OpenID, GoogleID and public unauthenticated tokens.

Authenticated users are given temporary, limited privilege credentials to access AWS resources, which are defined as infrastructure and platform services such as a compute instance, storage, fine-grained database access, and specific software stacks.

Roles are defined around users, groups and permissions. Access is defined on an API/services basis.

Logging is via integration with AWS Cloud Trail.

Google

Like AWS, Google defines resources as infrastructure and platform services, and uses identity management roles to define access permissions through a resource hierarchy based around organization, project and resource.

Google, at least for its container engine, defines Role-Based Access Control (RBAC) as fine-grained control over how users access the API resources running, for example on a container cluster. RBAC is used to dynamically configure permissions for cluster users and resources available to that cluster.

Comparison

While Azure, AWS and Google all use a mapping of resources to users, Azure uses a directory mapping structure that works for all types of resources, including users and databases as well as network, hosting and software. (Access control concepts may have developed from directory structures defined when Active Directory was re-architected for the cloud.)

AWS and Google have developed access control mechanisms and applied them to infrastructure and platform resources, that they integrate with user identity claims and tokens.

It will be interesting to see how these different types of RBAC implementation adapt and scale to the development of more specialised cloud computing services over time. It may be that Azure has a design edge for IAM.

Authentication and Access Permissions

There are two different scenarios for managing access permissions. One is for developer access, and the other, access to the runtime API for users.

Azure AD

Azure AD manages permissions to all Azure resources and resource groups, which can be network, infrastructure, software or users.

Azure APIM (API Manager) provides endpoint direction and response to internal and external authentication through header analysis of synchronous and asynchronous REST services called by the application. APIM provides default policies and customization capabilities

Azure B2C Identity Management product is a lightweight SaaS solution using the OAuth 2.0 Authorization Code Grant flow to mediate between the client and resource owner, managing authentication and authorization. Profile information is stored in Azure AD (cloud implementation evolved from Microsoft Active Directory). Azure AD provides for an initial user authentication on login. As part of initial identity management, there is a call to the Azure AD B2C token endpoint by passing client id, client secret, and other credentials. Azure AD uses internal cryptographic services and stores encrypted ‘secrets’ on behalf of each application.

Amazon AWS

AWS grants access permissions either as resource based policies or user based policies. Access control lists (ACLs) are one of the resource-based access policy options.

AWS IAM policies can be generated from default templates, or custom created. In effect they are programmatic steps for allowing or denying permissions to resources.

All policies are configured and deployed via the AWS API Gateway. To allow an API caller to invoke the API, the developer creates IAM policies that permit a specified API caller to invoke the API method for which the AWS IAM user authentication is enabled.

Google Cloud

Google Cloud IAM manages access control by defining who (members) has what access (role) for which resource.

The basic concepts are access policies applied to roles. Cloud IAM role primitives are owner, viewer and editor, while Google has established a large number of pre-defined roles, relating to cloud activities, such as biller, administrator, publisher and encrypter, further custom roles can be developed.

Cloud IAM has a wide concept for the accompanying concept of Identity, which is defined as a characteristic of infrastructure, platform, software and service accounts (which may be related to VMs or applications, but not users.) Each service account is identified by its email address, which is unique to the account.

The lifecycle of roles is managed by HTTP methods as REST calls, and access permissions are stored and queried as data tables.

Google Cloud Endpoints provide the API management for Google Accounts for authentication and access management. The framework provides tools, libraries and capabilities to generate APIs and client libraries from an App Engine application

Comparison

All three public cloud providers cover the same ground, historically from different concepts, which results in different deployment mechanisms.

It may be that the strong concepts behind LDAP and identity historically developed by Microsoft provide an edge with the ubiquitous identity being an endpoint URL given to a resource.

AWS has two different concepts for users and resources, which is used by the AWS API Gateway. Temporary security credentials are generated by an AWS global service with a single endpoint that can be configured to refer to any supported region, which makes it simple to deploy globally with minimal latency. However it may be complex to manage access definitions when reworking access security.

Google has a wider concept of roles and identities, however the unique identifier is currently an email address, which may prove problematic in view of the evolution of cloud services generally.

Google Cloud IAM also integrates with very customizable API management services, using Google App Engine as a back end.

In the short term, AWS has a simple approach by separating usage and infrastructure. In the long run, this may not sit as easily with technology change, particularly with hybrid cloud computing. Google and Microsoft are probably better poised to take advantage of ubiquitous public cloud services that can be delivered globally in expanded LTE networks.

As cloud services usage proliferates, and identity management is deployed globally it may be that the storage models for identity tokens, credentials and user profile data management become critical in view of location governance regimens.

Data Security Configuration

Data in Transit

Data arrives at the point of cloud provider traffic management for Domain Name System (DNS) mappings which directs client requests from the public internet when an end user accesses a cloud application endpoint.

All major providers offer automated configuration of failover, geolocation and latency for traffic routing. This includes health checks of automated requests to verify resources are reachable, available, and functional. Internet traffic can be routed away from unhealthy resources. Network layer load balancing provides for cross region performance.

All providers offer network layer DDOS protection for DNS servers. (Application Layer DDOS has to be built into the application design itself, as well as in the cloud platform. Currently only AWS Shield offers Layer 7 DDOS protection out of the box, although Azure and Google can link to third party cloud service providers.)

Transport layer symmetric encryption provides for an initial handshake and the generation of a shared secret key for application communications between multiple network devices. Certificate termination and protocol configuration is provided at

- Azure Application Gateway offers certificate termination
- AWS certificate termination combined with Elastic Load Balancer
- Google Cloud Load Balancing combined with certificate termination

The current view is that TLS 1.2 is not sufficient protection for private data (there are ongoing protocol vulnerabilities), and message level encryption may be required to provide additional protection to meet the new stringent requirements of the EU GDPR.

Internal Protection of Data

Cloud application internal communications use network routing directions for both public and private IP addresses in the context of cloud software-defined networking (SDN) that provides virtual networks made up of multiple subnets.

- Traffic between VMs in the same subnet.
- Between VMs in different subnets in the same virtual network.

- Data flow from VMs via the Internet.
- Allowing VMs to communicate with each other via IP-Sec VPN
- Enabling virtual machines to route to enterprise network via IP-Sec VPN or private fiber

Within the cloud perimeter, the following mechanisms protect data transport

- Public IP forwarding (uses whitelisting)
- Internal private IP forwarding (using regional cloud backbone)
- Access Control Lists with the ability for conditional allow and deny rules for private IP address ranges for particular ports

It is presumed that TLS can be configured as required, as communications are configured as part of the initial infrastructure architecture for all major cloud platform provider services.

Inbound Traffic

Mobile and desktop application sign up and sign in, when personal details are supplied are particularly vulnerable to interception.

End user devices are connected through public IP networks, traffic managed to the certificate termination point where pinned certificates are checked. Requests are forwarded using private IP addressing over encrypted links to the cloud API Management capability, where the appropriate policies are applied.

While PKI encryption is essential for high risk communications, for example, private data and banking details, two way PKI may be overkill for some applications. Mobile device native applications for Android and iOS are known to be particularly vulnerable to interception of messages and capture of keystrokes. For devices where 'jailbreak' attacks have been successful, both operating systems have known 'data leakage' vulnerabilities where user data has been found in clear text.

One method of reducing risk of compromising cloud applications by data leaks, particularly on mobile devices, is to maintain encryption algorithms on the server side. Responses from API management capabilities to end user devices can be encrypted with a short-lived public/private keypair, with the public key forwarded to the device, and the private key stored and referenced.

By using short lived public/private keypairs, information can be exchanged, such as personal information prior to the issue of short and long lived identity tokens.

Short lived application tokens can be issued and exchanged based on OAuth2.0 grant flow types. (Long-lived tokens are particularly vulnerable).

The OAuth 2.0 protocol has been customized for the web application sign up processes to address these vulnerabilities. Like most forms of application URI-based communication, the web application sends data over insecure local channels. Eavesdropping and interception of the authorization response is a risk for native apps. The PKCE RFC7636 standard was created specifically to mitigate against these vulnerabilities. It is a Proof of Possession extension to OAuth 2.0 that protects the code grant from being used if it is intercepted. It achieves this by having the client generate a secret verifier which it passes in the initial authorization request, and which it must present later when redeeming the authorization code grant.

This principle can be adopted by obtaining and validating the device ID for each communication prior to the granting of the authorization token. This does not change the fact that no identity token mechanism is currently invulnerable to the sophisticated types of attack currently carried out by well-funded state sponsored incursions.

Outbound Traffic Flows

External systems:
Network routing directs application traffic sent from the API management capability to external internet services.

- Connections via public IP address can be secured using encrypted links
- Cloud Application access to internal enterprise connections can be secured through public and private IPSec VPN routes.

Data at Rest

Data at rest defines the storage mechanisms for the cloud application data elements.

Data Store Security

Personal and technically identifiable information is encrypted in cloud application custom databases using database encryption mechanisms.

To select a best practice cloud provider, evaluation of database offerings can focus on cost effectiveness of addressing any gaps between the data transport mechanisms from the application to the database, and the encryption of data at rest, for all types of data storage mechanism.

Relational database security mechanisms for all providers offer request authentication of credentials, with appropriate permissions defined to access a cloud database instance.

Access control makes use of user roles and login credential validation integration with Identity Management components.

Providers offer various options for encryption, both at database level and field level. Key management options range from provider to customer managed keys. Keys can be optionally HSM generated by a hardware appliance. Providers offer FIPS 140-2 Level 2 HSM key services.

Key management services are able to be used and configured by end users. The level of encryption required depends entirely on the use case.

Common use cases for using provider key management services are:

- Geographically distributed applications
- In house cryptography capabilities insufficient
- SaaS providers do not want to manage customer keys
- Cloud key management access is monitored by cloud provider
- Compliance with stringent security regulations

Database Administration and Activity Monitoring

Database administration such as hardware provisioning, database setup, patching and backups is a specialised field, and may be performed either by the customer or by the cloud service provider, depending on the level of skill available.

Database management logging and monitoring views and dashboards are critical for audit of database activity, particularly for sensitive data.

Cloud providers log and monitor a variety of database management information, including performance data from database execution and transactions, as well as security information about user access to data.

There are a variety of third party cloud services offered to complement the logging and monitoring collected by cloud providers, and this capability forms a clear selection criteria for database services.

Cryptography

Enterprise cloud applications generally have traffic flow of inbound user traffic from both the internet and customer internal networks. Cryptographic algorithms protect inbound and outbound traffic to the web application using TLS at the transport layer for applications, and an IPSec VPN at the network layer to secure site to site communications.

Mobile and web applications make use of cryptographic functions to ensure the following security aspects

1. **Privacy/confidentiality:** Ensuring that no one can read the message except the intended receiver.
2. **Authentication:** The process of proving the identity of the sender.
3. **Integrity**: Assuring the receiver that the received message has not been altered in any way from the original.
4. **Non-repudiation:** A mechanism to prove that the sender really sent this message.
5. **Key exchange:** The method by which cryptographic keys are shared between sender and receiver.

Use of the following cryptographic architecture patterns ensures compliance with regulations for confidential information at rest and in motion.

There are three types of cryptographic functions currently in use:

1. **Secret Key Cryptography (SKC):** Use of a single key for both encryption and decryption; also, called symmetric encryption. Primarily used for privacy and confidentiality.
2. **Public Key Cryptography (PKC):** Use of one key for encryption and another for decryption; also, called asymmetric encryption. Primarily used for authentication, non-repudiation, and key exchange.
3. **Hash Functions:** Use of a mathematical transformation to irreversibly "encrypt" information, providing a digital fingerprint. Primarily used for message integrity.

Web application choice of functions are to meet current industry best practice. The following table provides an overview of web application cryptography.

Function	Cryptography	Code Library References
Authentication of mobile device and user during sign up and sign in to the web application	Public Key Cryptography. RSA (The minimum suggested RSA key is 1024 bits; 2048 and 3072 bits are even better)	NIST recommended JCrypto Libraries August 2011 RSA resources for Crypto-J NIST on compliant SHA-3 libraries March 2017
Encryption of password/pins	Hash Function (The minimum suggested is SHA-2. SHA-3 libraries are good for future proofing the web application	

Secret Keys in Common Usage

While there are many, the main secret key cryptography algorithms in use today:

Data Encryption Standard (DES): Two important variants that strengthen DES are: Triple-DES (3DES): A variant of DES that employs up to three 56-bit keys and makes three encryption/decryption passes over the block.

Advanced Encryption Standard (AES): The algorithm can use a variable block length and key length; the latest specification allowed any combination of keys lengths of 128, 192, or 256 bits and blocks of length 128, 192, or 256 bits.

Current Practice Public Key Cryptography

Public key cryptography algorithms that are in use today for key exchange or digital signatures include:

RSA: RSA uses a variable size encryption block and a variable size key. The key-pair is derived from a very large number, n, that is the product of two prime numbers chosen according to special rules; these primes may be 100 or more digits in length each, yielding an n with roughly twice as many digits as the prime factors.

Elliptic Curve Cryptography (ECC): Examples of this type of encryption are Diffie-Hellman elliptic curve and NSA Suite B. This is A PKC algorithm based upon elliptic curves. ECC can offer levels of security with small keys comparable to RSA and other PKC methods. It was designed for devices with limited compute power and/or memory, such as smartcards and PDAs.

Note: Public Key Cryptography Standards (PKCS) is a set of interoperable standards and guidelines for public key cryptography, designed by RSA Data Security Inc. It covers topics such as passwords, certificates, tokens and keys. It is not an official standard.

Current Hash Functions

Hash algorithms that are in common use today include:

Message Digest (MD) algorithms: A series of byte-oriented algorithms that produce a 128-bit hash value from an arbitrary-length message.

> **MD5** (RFC 1321): Developed by Rivest after potential weaknesses were reported in MD4; this scheme is similar to MD4 but is slower because more manipulation is made to the original data. MD5 has been implemented in a large number of products although several weaknesses in the algorithm were demonstrated by German cryptographer Hans Dobbertin in 1996 ("Cryptanalysis of MD5 Compress").

Secure Hash Algorithm (SHA): Algorithm for NIST's Secure Hash Standard (SHS), described in FIPS 180-4.

> **SHA-1** produces a 160-bit hash value and was originally published as FIPS PUB 180-1 and RFC 3174. It was deprecated by NIST as of the end of 2013.
>
> **SHA-2**, originally described in FIPS PUB 180-2 and eventually replaced by FIPS PUB 180-3 (and FIPS PUB 180-4), comprises five algorithms in the SHS: SHA-1 plus SHA-224, SHA-256, SHA-384, and SHA-512 which can produce hash values that are 224, 256, 384, or 512 bits in length, respectively.
>
> **SHA-3** is the current SHS algorithm. Although there had not been any successful attacks on SHA-2, NIST decided that having an alternative to SHA-2 using a different algorithm would be prudent.

Transport Layer TLS/SSL.

TLS/SSL encryption provides guarantees of integrity during transmission. The private key used to generate the cipher key must be sufficiently strong for the anticipated lifetime of the private key and corresponding certificate. The current best practice is to select a key size of at least 2048 bits. Additional security is provided by Key Management services

After the initial certificate handshake, a common practice is to use an ephemeral (temporary) shared secret key exchange such as the Elliptic Curve Diffie-Hellman (ECDHE) Diffie-Hellman for forward secrecy with an appropriate length for the generated temporary key

Strong TLS 1.2 is currently using the ECC with SHA 2 (e.g. SHA-256 and SHA-384 algorithms) as a signature to prove ownership of a private key. (ECDSA with SHA-1 is now deprecated).

Application inbound traffic from mobile and web users is routed through a WAF protected application gateway, encrypted using TLS/SSL to protect data during transmission.

TLS 1.2 provides authentication of the server certificates to the client application.

Public certificate provider certificates can be used for internal certificate requirements. Cloud key management services often provide renewal functionality.

Network Layer IPSec VPN

IPSec VPN encrypted communications can be used for inbound traffic from existing enterprise applications to public cloud providers.

Outbound traffic can use IPSec VPN link from internal services to enterprise VPN Gateway endpoints.

Certificate Pinning

Certificate Pinning is a method of enhancing security characteristics of device communications. It is the process of associating a host or device with the expected X509 certificate or public key. Once a certificate or public key is known or inspected, the certificate or public key is associated or 'pinned' to the host.

Applications generate a public/private key pair, and the public key and identifying information for the customer is sent to the Certificate Authority who subsequently issues the certificate. The public key and identifying information is stored in the certificate. The private key is kept secret by the customer.

Certificate pinning is the process of associating the app public key or certificate (the public key is the basis for the certificate) to the mobile application by 'pinning' the public key/certificate to the mobile device. This means that only a specific web application certificate is recognized by the mobile operating system (or browser).

A copy of the certificate or the public key is placed either in the application code for installation onto the mobile device, or installed on the mobile device on first usage. It is expected that the certificates are to be renewed on a periodic basis, and the certificate can be updated. The current the web application already has a pinned certificate. A new certificate can be copied to the mobile device to replace the old one when an existing certificate expires.

During the SSL handshake (first request to the server), the application verifies that the public key of the server certificate matches the public key of the certificate that is stored on the mobile device. If the certificates do not match, the mobile device discontinues the connection.

Cloud Application Data Models

As for enterprise applications, cloud applications require logical data models managed by functional domains. This makes it far easier to track, monitor and manage security incidents

ETL: Detailed models of the ETL of the principal domain entities and entity relationships, providing information about data elements, data inputs and output flows, data transformations and applied business logic.

Interface Models: Detailed interface models providing descriptions of interface attributes, fields, types, methods and applied business logic.

Entity Relationship Models: There are some detailed domain data models providing information about the physical representation within the respective application databases.

UML Models: End-to-end standard UML domain models, that provide a combination of behavior (e.g. sequence) and structure (e.g. class) elements, both at the conceptual, implementation and deployment level.

PaaS Security Configuration

The cloud provider physical and perimeter security provides the context for the security of the cloud provider platform services.

PaaS Security

The cloud provider PaaS services are protected by the following cloud provider perimeter security measures at Layer 3 (Network), Layer 4 (Transport) and Layer 7 (Application)

The cloud provider perimeter comprises:

- Network Layer DDOS prior to internet traffic reaching cloud provider virtual networks, VMs and subnets. This is a layer of the cloud provider physical network that protects the cloud provider platform itself from large-scale internet-based attacks. This is generally not user configurable. It is applied prior to public IP NAT address translation.
- Small scale DDOS attacks that target an application weakness specifically would not be detected by the cloud provider network DDOS. Customers must elect to use application layer DDOS protection.
- The perimeter network subnet communicates to the internet directly, and routing tables provide communication to and from the back end and on-premises networks.

Cloud application traffic goes through the cloud provider application layer WAF. However not all WAFs are created equal, and care can be taken to evaluate the level of protection offered.

Fine grained access management is enabled by cloud provider Role Based Access Control (RBAC) to cloud service offerings (see Identity Management).

Load Balancers and Application Gateways

Comparative analysis of load balancers and application gateways can include the level of configuration available to customers, and the information able to be accessed from operational dashboards. Otherwise they all form similar functions, though each provider has their own set of terms to describe them.

Logging and Monitoring

Monitoring and incident security event management operates by analyzing both the packet header, the TCP header and the data payload contained in network packets. This means that the application layer contents are used to allow or deny connections. This has become the means to counteract the growing number, type and nature of threats and attacks from malware.

The cloud provider WAF can identify network packets from sources via blacklisting (negative), and whitelisting (positive) and a mixture of both.

Negative patterns

These are patterns of threat that WAF vendors have addressed correspond to well-known attacks such as the OWASP Top 10 security threats.

- A1 Injection
- A2 Broken Authentication and Session Management
- A3 Cross-Site Scripting (XSS)
- A4 Insecure Direct Object References
- A5 Security Misconfiguration
- A6 Sensitive Data Exposure
- A7 Missing Function Level Access Control
- A8 Cross-Site Request Forgery (CSRF)
- A9 Using Components with Known Vulnerabilities
- A10 Unvalidated Redirects and Forwards

Positive patterns

These are application specific and ensure that content is as expected, for example the datatypes expected in fields.

API Management

An API Management capability allows functionality to be exposed as APIs and published on a self-service portal that can be used by application developers who want to consume those APIs.

The cloud provider API management provides communication routing from cloud applications to registered endpoints on the cloud provider hosted applications, software and platform services. Part of an API management service is the inspection of tokens, such as JWT tokens identity claims, that pass parameters from the cloud application to the cloud provider services.

API management policy types generally include:

- Access restriction e.g. token validation, APIM call rate limits,
- Request forwarding, conditional evaluation and execution, call logging
- Authentication
- Message transformation
- Configuration of cross domain calls

Container security

Cloud applications can be deployed as microservices, a development model that is well suited to container SaaS such as Docker.

Containers are usually isolated by the software defined network. This arrangement provides some preliminary perimeter network security for inbound and outbound traffic.

Pipeline endpoints can be configured to use cloud provider file storage, and access control can be configured for code repository access.

Container orchestration provides for the co-ordination of the management, control and data planes for containers.

Linux Control Groups and isolation of procs provides the basis for container operations, and vulnerabilities that include access to Linux filesystems.

Agents provide container health data, which can be used to monitor container performance and co-ordination, aggregate data for preventive management, as well as raising alerts and alarms.

Fine grained access control can be configured and managed by cloud provider identity management services.

Container services provide schedule management, resource management and service management within a distributed Linux nodes context for the deployed containers on a managed container cluster.

Security features for containers include:
Identity Management

- Control of access to management interfaces, using IDM access controls
- Enforces authentication on connections to cluster
- Communication routing can be access controlled.

Network Security

- Designed to work with cloud provider standard network templates, and configured within software defined networking.

IaaS Security Configuration

Software defined networking provides for an on-demand configurable pool of shared computing resources allocated within the public cloud environment, providing a certain level of isolation between the different tenants. The isolation between one tenant and another as well as other public cloud users is achieved through allocation of private IP subnets and encrypted virtual communication channels.

The physical locations for the cloud provider deployment can also provide for regional and cross regional disaster recovery.

Security Elements

The following cloud infrastructure (network and hosting) applies to all the major cloud providers

- Perimeter firewalls filter packets coming into the network.
- Operating system firewalls can allow or deny packets in the application services.

- Intrusion detection/prevention systems drop suspicious packets, and disconnect unauthorized connections.
- Partitioned virtual networks control traffic passing between different IP subnets, enabling separate address spaces for cloud workload.
- VMS communications remain private within a virtual network.
- Site to site connectivity using VPN gateways or virtual appliances, allows communications to and from the cloud provider
- Network Access Control Lists allow or deny traffic to network interfaces, individual VMs, or virtual subnets, within virtual networks, enterprise to cloud, and public internet communications.
- Fine grained RBAC to cloud resources can be managed from a user interface.
- Automated logging and monitoring systems provide data for forwarding to endpoint protection technologies.

Application Security Testing

The security configuration of the application services comprises a number of key areas.

User Interaction

Review of application code installation, configuration and deployment is required.

Application Testing has to cover the following areas:

- Application analysis - logic, data flow and platform
- Vulnerability to client attacks - application code, data leakage, data storage, file system, binaries, runtime and XSS vulnerabilities
- Vulnerability to network attacks – traffic analysis, encryption

OWASP provides a good starting point for application penetration testing as well as mobile top 10 vulnerabilities

Defensive Coding Practices

The OWASP Top 9 coding security flaws are listed in this section the most critical security flaws to find during a code review. These coding practices may require extension to address container micro services security.

Input Validation
Source Code Design
Information leakage and improper error handling
Direct object reference
Resource usage
API usage
Best practices violation
Weak Session Management
Using HTTP GET query strings

DevOps Security

It is clear that DevOps require risk analysis of cloud application security practices, as there is a clear increase in exposed vulnerabilities because of the automated

development code deployment and the reliance on security testing that may be less than rigorous.

The cloud application DevOps configuration information has to be documented clearly, and reviewed for secure coding prior to deployment to the cloud.

The speed of deployment with the Continuous Integration/Continuous Deployment (CI/CD) style of application development provides both the strength of DevOps and the weakness.

The speed of automated deployment that is not sufficiently tested provides for coding weaknesses that may not be detected until an ingress has occurred.

While this is a good innovation, from a security perspective there is a considerable increase in risk while the practices are early days, and most technologies have not yet been adapted to automate rigorous security testing frameworks prior to deployment as part of code delivery.