CHAPTER 1

Introduction: Dimensions of Cloud Security

Cloud adoption is a strategic move that enterprises take to optimize cost, mitigate risk, create scalable infrastructures, and build enterprise-ready applications. Different organizations can adopt different depths of cloud security, depending on their best practices.

Organizations that adopt cloud-based technologies have to identify any security risks and initiate controls to keep data in the cloud secure. Organizations should take the following measures to ensure the security of their cloud services:

- Match the organization's security requirements with the cloud's security requirements
- Audit and analyze cloud security policies with a history of transparency and security-related practices
- Understand the compliance requirements and certifications that the cloud service provider requires

This chapter covers the following topics:

- History of security and the public cloud
- Cloud security boundaries and responsibilities
- Pattern-based security
- Azure's defense-in-depth security architecture

S. Lad, *Azure Security For Critical Workloads*, https://doi.org/10.1007/978-1-4842-8936-5_1

History of Public Cloud Security

In the world of cloud computing, cloud solutions are ever-evolving. Organizations should ask themselves the following questions before moving to a cloud-ready environment:

- Does our organization have a security architecture in place?
- Does our organization have a future security roadmap?
- Does our organization have a security process to remediate security vulnerabilities?

If the answer to all three questions is yes, then your organization must have considered security very thoroughly and must belong to the small percentage of companies that focus on security. The bad news is that this approach could need a major overhaul when your organization decides to leverage cloud computing services.

Traditional security in the IT world was implemented from the mindset of limiting access boundaries to applications. In traditional security architectures, security controls resided in the data center and applications lived happily under the protection of the data center. Organizations had complete control of the data center and could operate the security controls the way they wanted in terms of racks, cables, rooms, and access to comply with security policies. This required proper management.

More often than not, reluctance to move to public cloud architectures is due to the fear of giving up data centers rather than any other rational cost benefits. The fear of not being able to trust someone else is understandable. And even with organizations that are brave enough to go ahead, the estimation provided by the engineers to implement new security controls can seem like a gigantic investment, as it also must include the cost of procuring new tools, teaching new skills, and implementing the system.

Risk assessment needs to be based on the organization's risk appetite, so it can evaluate which security controls could live in a public cloud. Normally, when organizations perform this exercise, it is an eye-opener and more often than not the gaps are really wide. Most traditional security controls are not applicable to risk assessment, as the parameterization in the public cloud is not as strong as with traditional or on-premises data centers.

Enterprises are rapidly moving toward cloud technology and cloud services. This trend will only increase in the coming future. Cloud computing is not the future anymore; it is already everywhere, although most end users don't even realize it. In the cloud computing world, security policies need to change their focus from "controlling access" to "protecting data".

In the cloud computing world, security requirements around data include the following:

- Confidential information must be encrypted in a secure manner
- PII (personally identifiable information) data must be protected
- Data integrity must be maintained
- Secure data disposal must be built in a mature manner
- A data lifecycle must be followed to ensure proper data usage

In cloud computing, two security controls become more important than ever before. The first is certificate management. It plays a very important role in all web service security solutions. Organizations need to make sure that they can create, renew, and revoke certificates from a central place in no time. The requirements to use the Public Key Infrastructure (PKI) must be well established, understood, and followed. The other control is Identity and Access Management (IAM), which has become the most important security control in the cloud computing world, where a lot of solutions like OpenID Connect (OIDC), Security Assertion Markup Language (SAML), and Open Authentication (OAuth) exist.

With cloud computing, security breaches can happen easily at multiple levels of technology. Therefore, defense-in-depth and complete mediation are the most important principles to follow. For business leaders, it's hard to know whether a cloud service is secure enough for their IT applications to work accurately. Business leaders have to trust their cloud providers in this regard. But security is a shared responsibility, and no cloud provider will provide a full guarantee for securing the workloads. The shared responsibility is the very reason that security architecture needs to be reconstructed in the view of risk assessments for the cloud computing model. See Figure 1-1.

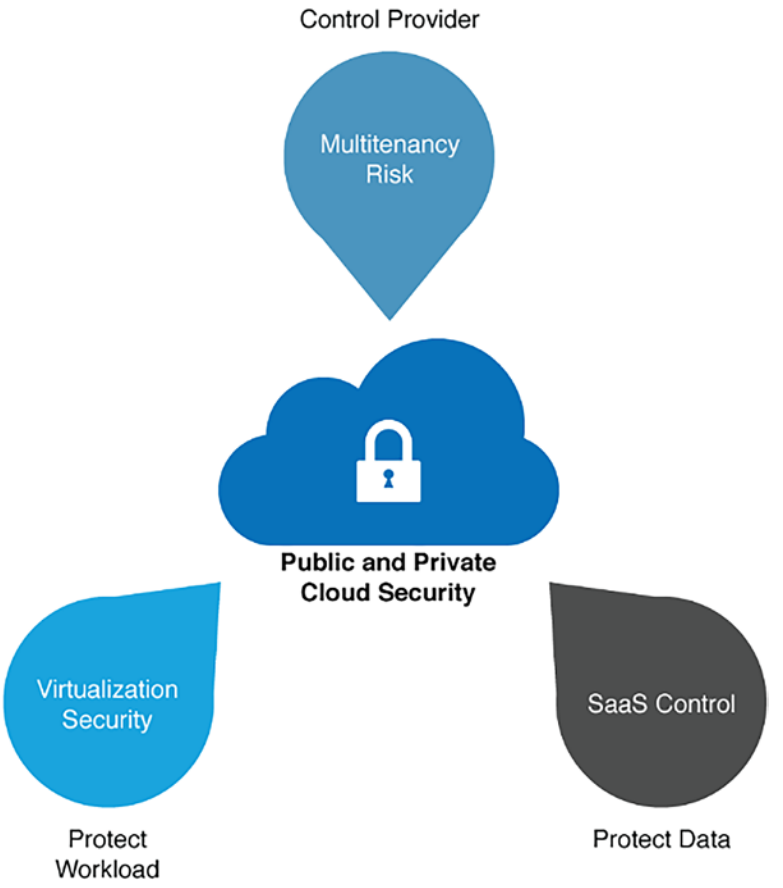

Figure 1-1. *Public cloud security*

In a nutshell, cloud computing growth has been truly astonishing and it will continue to grow in the future. Although cloud computing has many benefits, it doesn't reduce existing network security risks.

- Security risks threaten the data center and network change once the applications move to the cloud. Such security risk remains. For example, data center applications use a wide range of ports that make traditional security considerations ineffective when those applications move to the cloud. See Figure 1-2.

Figure 1-2. *Network security*

- Cloud computing works on the model of shared resources. It is best to separate mission-critical applications and data applications into secure network segments. This separation is also known as *zero trust segmentation*. See Figure 1-3. It is very straightforward to implement zero trust segmentation on a physical network of the enterprise data center using firewalls and virtual LANs based on the application and user identity. However, in a cloud computing environment, direct communication between the virtual machines occurs constantly. This makes the segmentation task difficult since cloud computing works on the concept of shared resources.

Figure 1-3. *Zero trust security*

- In cloud computing, virtual workloads can be modified in seconds/minutes. But in a traditional workload, security and other configurations can take hours. This imbalance causes problems in security policy and workload deployment during cloud migration. See Figure 1-4.

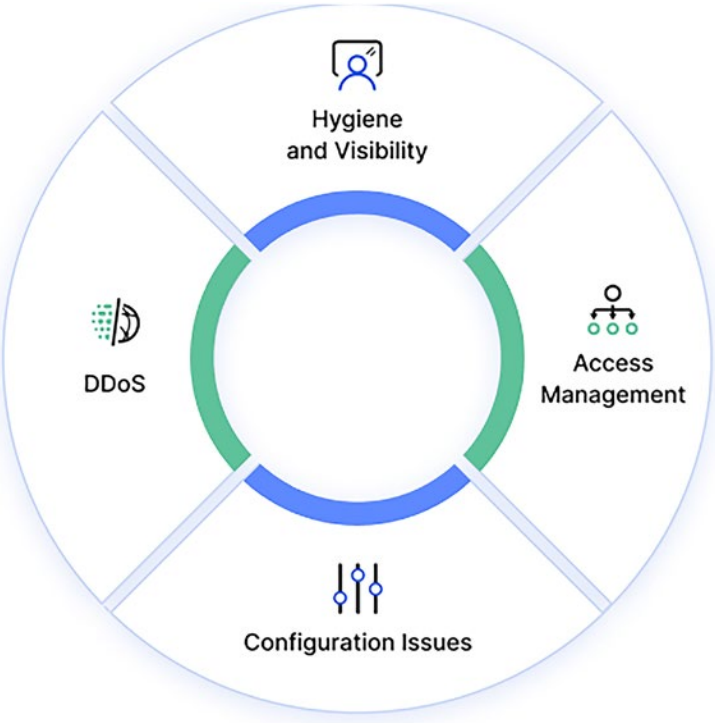

Figure 1-4. *Cloud security configurations*

When you dive deeper into cloud security considerations, you see that it is basically a shared responsibility between the cloud providers and the customers. In a shared responsibility model, there are three main categories:

- Completely the provider's responsibility: Securing the infrastructure as well as patching the compute resources
- Completely the customer's responsibility: Managing users and access privilege, preventing unauthorized access to the data, encrypting data and protecting cloud resources

- Responsibilities that differ based on the cloud service model are as follows (see Figure 1-5):
  - Infrastructure as a Service
  - Software as a Service
  - Platform as a Service

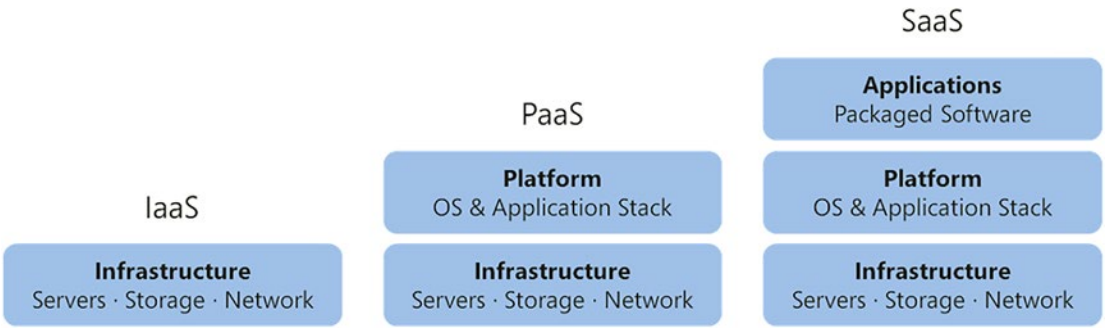

Figure 1-5. *Cloud security shared responsibility models*

Considering the popularity of the public cloud, you should be aware of these common cloud security challenges:

- Increased surface attack: Public cloud environments are more vulnerable to surface attacks by hackers and a poorly designed cloud security solution can disrupt the cloud workload.
- Lack of transparency: Public cloud providers have full control over the complete infrastructure in the IaaS model. This leads to a lack of transparency toward the end customers and is also extended to the PaaS and SaaS models.
- Dynamic cloud workload: Cloud environments are very dynamic and constantly changing at scale and with velocity. Traditional security tools are static in nature and not capable of enforcing security policies in flexible and dynamic environments.
- DevSecOps and automation: Enterprise organizations started adapting the highly automated DevOps CI/CD culture to ensure that all security controls were in place and embedded in the form of code to ensure the development lifecycle. Once the workload is deployed to production, security-related changes can have cause security concerns and increased time to market if the DevOps process is not followed.

- Granular access management: Cloud user roles cannot be configured loosely and grant more privileges than what is required. An example is giving database delete or write permissions to users who don't need such privileges.
- Cloud governance and compliance: All public cloud providers are aligned with well-known governance and compliance programs but customers are also responsible for ensuring that the workload, the data lifecycle, and its processes are compliant.

Now that you've read about these cloud security challenges, it's time to explore the six pillars of creating robust cloud security:

- Granular IAM and authentication controls: It is recommended to work with groups and roles rather than with individual IAM access. You need to grant only the minimal access privileges to assets and APIs for the group. It is best to proactively manage IAM access with strong password policies, permissions, and so on.
- Zero-trust network security control: It is advised to deploy business-critical resources in a logically isolated area of the cloud provider's network, such as virtual private cloud or virtual private network. Use subnet to micromanage the traffic between the workloads.
- Compliance to change-management process and software updates: Cloud security vendors provide robust cloud security posture management to apply governance and compliance rules when provisioning the servers.
- Secure application with a web firewall: A firewall will granularly inspect and control traffic to and from the web application servers and automatically update Web Application Firewall (WAF) rules in response to traffic changes.
- Data protection: Improved data protection with encryption at all transport layers, secure file shares and communications, risk management, maintaining secure storage of data, and so on, should be considered.

Table 1-1. *Data Protection Principles*

Lawfulness, fairness and transparency	Personal data shall be processed lawfully, fairly and in a transparent manner in relation to the data subject
Purpose limitation	Personal data shall be collected for specified, explicit and legitimate purposes and not further processed in a manner that is incompatible with those purposes
Data minimisation	Personal data shall be adequate, relevant and limited to what is necessary in relation to the purposes for which they are processed
Accuracy	Personal data shall be accurate and, where necessary, kept up to date
Storage limitation	Personal data shall be kept in a form which permits identification of data subjects for no longer than is necessary for the purposes for which the personal data are processed
Integrity and confidentiality	Personal data shall be processed in a manner that ensures appropriate security of the personal data, including protection against unauthorised or unlawful processing and against accidental loss, destruction or damage, using appropriate technical or organisational measures
Accountability	The controller shall be responsible for, and be able to demonstrate compliance with the GDPR

- Threat intelligence: Third-party cloud security vendors work by collecting all the cloud-native logs with internal data, such as asset and configuration management systems, and vulnerability detection, as well as external data such as threat intelligence feeds, and so on. They also provide tools to visualize and query the landscape and incident response times. Real-time alerts about policy violations shorten time to remediation. There are various types of intelligence threats, as highlighted in Figure 1-6.

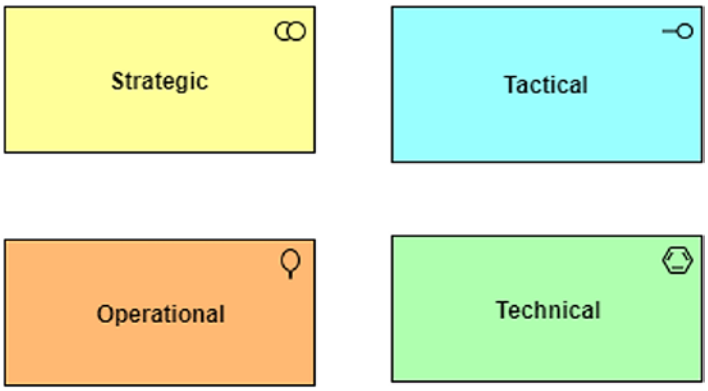

Figure 1-6. *Threat intelligence types*

Cloud Security Boundaries and Responsibilities

Decoding what "cloud-ready application" means is like opening a can of worms. First of all, there are no basic steps, as it's still a new topic and can depend on assumptions and opinionated views. If you Google "cloud-ready" it's sometimes defined as "designing applications for the Internet". Well, frankly that's like saying "a car is designed for the road". Although this is not wrong, such definitions are not helpful, as it is hard to write any requirements with this explanation.

Given the evolution in technology today, being "cloud-ready" could be as simple as containerizing an application and hosting it over a managed infrastructure. While this is partially correct, cross-cutting concerns like security and resiliency are the ones that create havoc in the quest to become cloud-ready. The scope of cloud-ready does look small and manageable at first, but the further down it goes, the bigger it gets.

Cloud-ready applications adhere to the following three principles (see Figure 1-7):

- The infrastructure must no longer be perceived as a bunch of physical boxes of hardware, but rather a bunch of managed services. In the cloud-ready world, an infrastructure platform is an integrated set of equipment, operating software, middleware, databases, and centralized services that help the applications run seamlessly.
- The infrastructure platform includes the provisioning, hardening, and configuration of the platform services and is based on easily-scalable, hyperconverged infrastructure. The infrastructure platform must be easily reproducible and leverage the Infrastructure as Code (IaC).

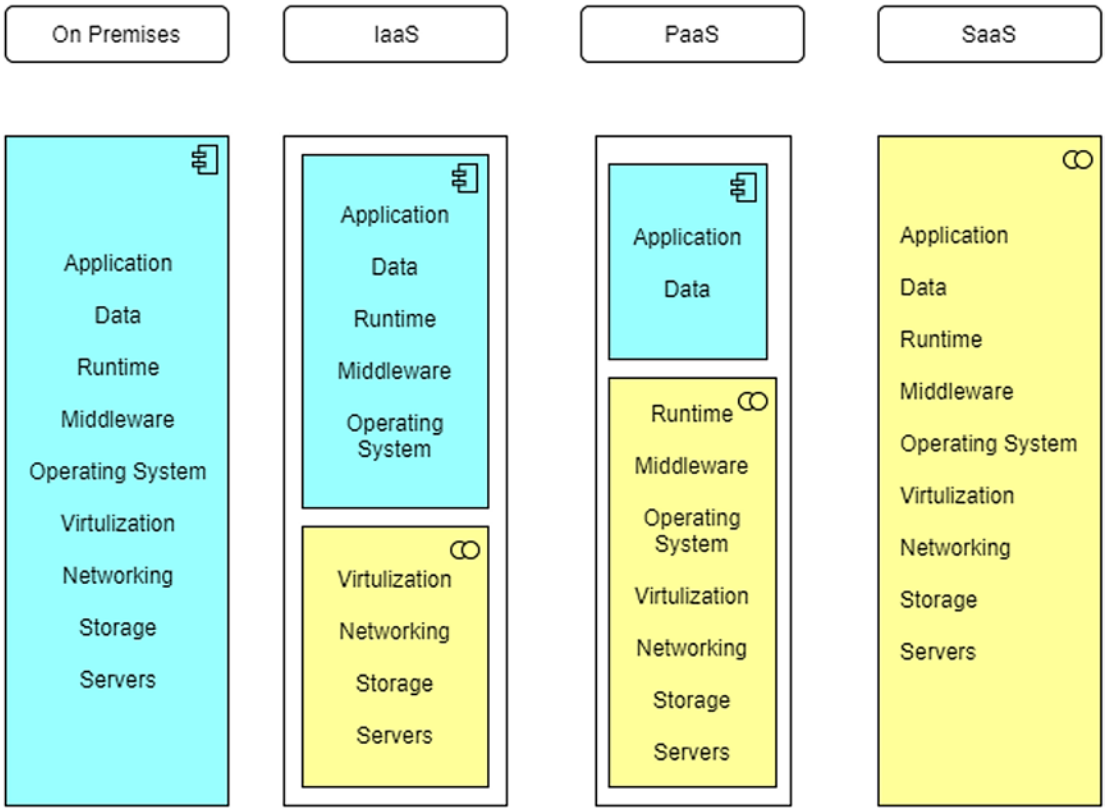

Figure 1-7. *Cloud security boundaries*

- To enhance IT resiliency, applications must be written to allow failures and handle these gracefully without loss of service. Failure on the cloud is inevitable, so applications that are truly cloud-ready must be designed for failures. Such applications are far more capable of self-healing, restarting, and continuing operational services when the worst happens. This is more than just a disaster recovery plan. Applications must have been thoughtful about failure modes like hardware failures, operating system failures, Internet failures, network peering issues, and other aspects that may be outside their direct control. They must still be able to handle these gracefully and automatically.

These are possible by embracing new resilience patterns like retry, circuit breakers, Command and Query Responsibility Segregation (CQRS), sharding, and many more. Technology architecture plays an important role here, by designing systems that are loosely coupled, auditable, and based on proven resilience patterns. It is also common to embrace principles of microservices for application development and event sourcing via immutable events for auditing. In addition, any file sharing or direct database interactions need to be replaced with asynchronous or synchronous communication styles.

Figure 1-8. *Cloud application security best practices*

Traditional security controls are not enough and security architecture needs revamping. Traditional security was implemented from the mindset of limiting access boundaries toward the applications.

In cloud-ready architectures, the perimeters for access boundaries are much looser and new security controls need to be implemented. Two security controls become more important than ever before:

- The first is certificate management and it plays a significant role in all web service security solutions. Organizations need to make sure that they can create, renew, and revoke certificates from a central place in no time. The requirements for use of the PKI must be well established and understood.

- The other control is Identity and Access Management (IAM), which has become the most important security control in the cloud computing world, where a lot of solutions like OIDC, SAML, and OAuth exist.

Figure 1-9. *Traditional application security*

These three pointers highlight where energy and design efforts need to go to create a cloud-ready enterprise application. More importantly, a cloud-ready organization also needs a cultural change, which includes embracing new skills. To begin, your organization needs a good vision and strategy. Such transformations can take time. Just like it takes a whole village to raise a baby, it will take the whole organization to become cloud-ready!

Pattern-Based Security

We often talk about modern principles like infrastructure as code, microservices for applications, service-oriented approaches with enterprise bus, and so on. How often do we underestimate the work that needs to go into these principles?

In the good old days, the simple architecture of the client-server database was centralized. Its simplicity was the key to keeping it running in a confined network space. Access was controlled using simple firewalls and login passwords with basic authentication.

Figure 1-10 shows how to identify the list of threats for each asset and identify the controls that mitigate each threat. Once the assets are identified, the next step is to create an asset-centric access list for each asset.

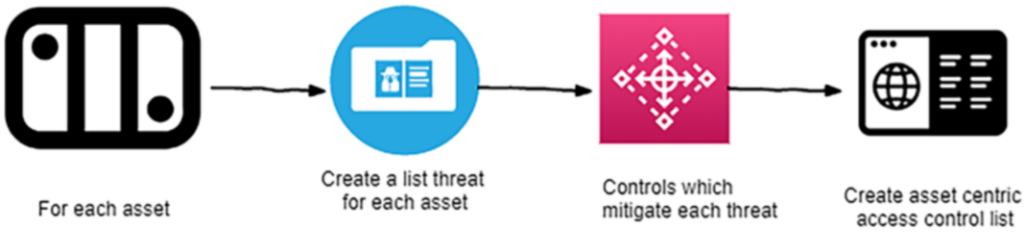

Figure 1-10. *Cloud security pattern*

These worked brilliantly in the 1990s, as connections were small and distribution was not a requirement, but a wish. The challenge in those days was about balancing cost to meet timelines rather than dealing with complexity. Architecture was not in the limelight, as more often than not, the architecture would just be a few flows in a client-server database set.

Times have changed. Distribution is a requirement and delimiting access is not considered a smart thing to do. Today, client-server-databases are considered an anti-pattern to modernization. Speaking about client-server-databases sounds like the 1990s.

Today's architecture needs to be built on high-level principles of scalability, with the core requirements to protect data and allow access in a controlled fashion. The needs of a scalable architecture are moving to a pluggable interface, where ubiquity is the norm. Client-server, microservices, and service-oriented architecture all work together brilliantly well in a pluggable way in hybrid cloud models. But to do this, the architecture roots have to be very strong, and security is one of the biggest pillars of this foundation. See Figure 1-11.

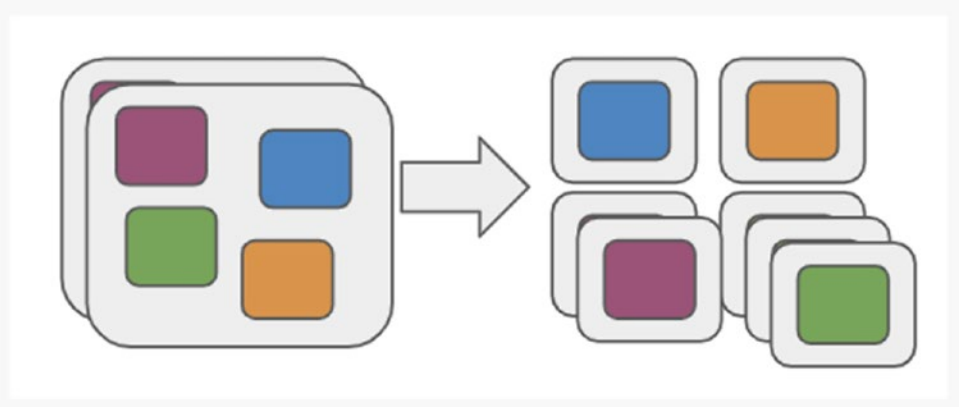

Figure 1-11. *Monolithic vs microservice architecture*

A fundamental part of a scalable, modernized architecture is pattern-based security. This is fundamental for companies looking at a hybrid cloud model, as networking partitions will blur over time and applications are becoming more accessible and vulnerable than in the 1990s.

Pattern-based security is based on the fundamentals of compartmentalization with security zones. The compartmentalization is logical and achieved with security dimensions. As a simple example, assume you have a very simple service-oriented architecture with three layers—a consumer, an API gateway, and a provider. The layers are one security dimension. Then comes the famous CIA (Confidentiality, Integrity, and Availability) ratings, which is another security dimension. Another security dimension is the consumer type, internal or external. Based on your organization, more security dimensions could be added. See Figure 1-12.

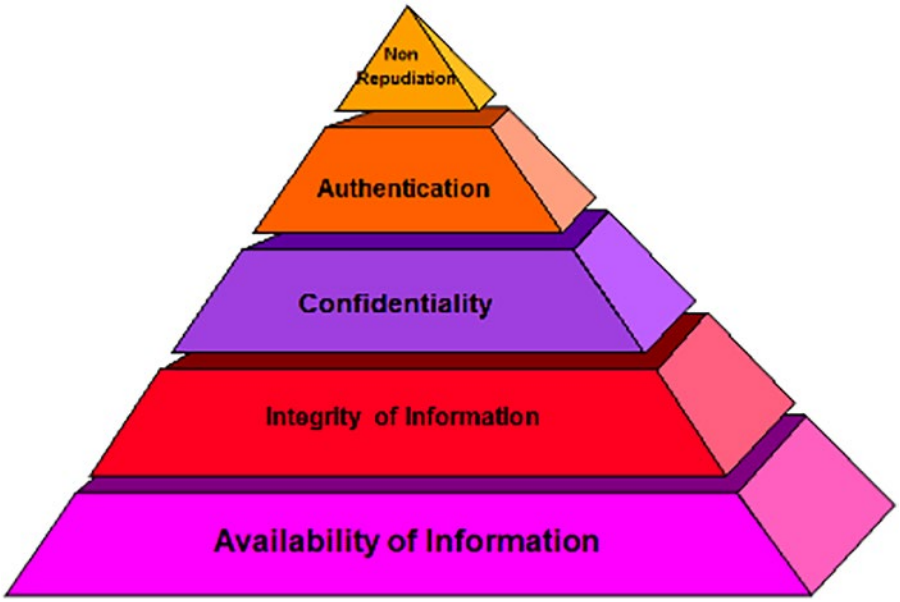

Figure 1-12. *Security dimensions*

Continuing from the security dimensions already discussed (layers, CIA rating, and consumer type), try to map them to security zones:

Layer; CIA; Consumer type; Computed Security zone

Consumer;111; Internal; Extra Small

API gateway;111; Internal; Medium

Provider; 111; Internal; Large

This table can go on, to an enterprise security model. What is important is that it is clear how many security zones you need. As a rule of thumb, there should not be more than ten security zones.

To build on the story further, every security zone must be mapped to security controls. The security controls must be based on the risk-based policies of the organization. (For example, an application with confidential data may need a firewall, encryption, and an authentication and access control list, but another application without data may only need the security control of the firewall.) Continuing from the fictitious example, you could translate this to the following mapping of security zones and security controls:

1. Extra small security zone - Firewall
2. Medium security zone - Firewall + Encrypt
3. Large security zone - Firewall + Encrypt + Authentication + Access Control List

The aforementioned security zones and their security controls are only examples. A real example of any medium-to-large enterprise would include more. (At rare times, it could be more than ten zones. If this is the case, the architecture may be very complex due to other reasons not related to security.)

Each of these security controls must be then governed with security policies and rich sets of standards, guidelines, and acceptance criteria, probably also dash-boarded to show the weak points of the application landscape.

Such a model is a holistic approach toward pattern-based security and has rich scalability. Such a model allows decision-making toward a hybrid cloud strategy, enabling decisions like what to offload into the public cloud and what to retain in private data centers.

Pattern-based security is a prerequisite of any scalable, modernized architecture. It keeps the complexity in control and still leads to a complete mediation. In addition, such a strong foundation with pattern-based security enables higher management to make strategic decisions on the future of data centers and, more importantly, estimate costs better.

In addition to understanding pattern-based security, you need to understand the software layers that create a scalable application security model for moving to a distributed architecture. See Figure 1-13.

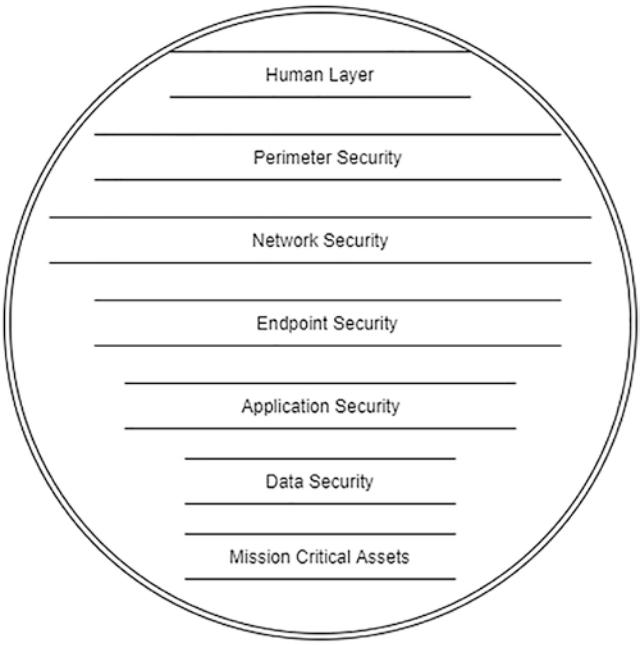

Figure 1-13. *Security layers*

The speed of digitalization creates the urge and need to be ready for the next phase of execution, which consists of heterogeneous integration flows. Decoupling the architecture from static dependencies is becoming the most important characteristic for the survival of the organization. Decoupling creates the flexibility to replace applications or legacy systems, but also to scale quickly and to monitor and guarantee overall security. One aspect that's a beautiful practice is embracing software layering. The number of layers is not important, but the architecture principles per layer is really important. The principles for application layers are as follows:

- A centralized integration layer to decouple the service consumers and service providers. The integration layer must be responsible for message routing, transformation, and mediation.
- Multiple consumer layers that are equivalent to the demand for consumable services and more often than not have a user interface with presentation views. No databases must be hosted in the consumer layer and no two consumers could have a data flow.

- Multiple provider layers are responsible for delivering services that consist of heavy lifting business logic, statefulness, and persistence. Any provider service must be exposed as services in the integration layer.

Consumer Layer

The consumer layer hosts consumer applications that create business value with personalized, engaging, and interactive experiences. These interactions can be for a single screen type use or can involve multiple viewports and devices to present the functionality in fit-for-purpose Human Machine Interface (HMI).

While consumer applications are responsible for service functional needs, from non-functional aspects, the consumer layer also increases availability and resilience by coping with failures to backend systems. An important element of any consumer application is gracefully handling exceptions, while decreasing response times by using in-memory caching and content delivery networks.

The consuming layer application must have a faster time to market. The consumer layer application must not hold any persistence.

Integration Layer

The integration layer is the decoupling layer and must be composed of either one or polyglot technologies that serve as the only gateway for exchange of data flows between the service consumer to the service provider. The integration layer must also provide a minimal set of capabilities to enable service providers and service consumers to interact.

When the service provider is outside the organization, the integration layer can also hold data in persistence storage.

The integration layer provides standardized ways of message routing, transformation, and mediation in order to expose enterprise services.

The integration layer can safeguard quality, assure the overall architecture, and apply security controls.

An integration layer must be mapped to a separate security zone and controls. As an example, an integration layer must use client certificates to safeguard the integrity and confidentiality of the provider applications.

The interface layer is only responsible for interfaces and hence can be a place to standardize the standards. In today's world, the SOAP (XML), REST, and JSON standards are the most common.

As integration layer is not an implementation layer, so no heavy business logic should reside in it.

In order to protect the unintended flow of information to consumers, the integration layer is also responsible for security controls like the access control list and encryption in addition to other basic requirements, like routing and throttling (rate limiting). These security controls should be formulated based on the business needs.

The integration layer can also act as a stitching layer for legacy-consuming applications that require any form of integration or transformation.

Provider Layer

Provider layer applications provide data and business functionality.

A provider layer application must shield the data and business logic, as these are the core of any business. The provider layer should not provide the integration layer with direct access to the databases, rather the databases must be wrapped by reusable services.

Based on the needs of the business, the provider layer component could host a database for persistence. In today's DevOps world, established, open source database solutions prevent a potentially costly vendor lock-in and allow companies to benefit from the input of the open source community. Many open source databases are available in the market, with SQL and NoSQL flavors like MongoDB and Cassandra.

The provider application must be based on the principle of reusability and must be atomic for the functionality it provides.

Provider services, which then depend on other provider services, can introduce static dependencies, which need to be avoided and will otherwise increase the complexity of the overall architecture.

Linking Software Layers to Application Security

The previous introduction to software layers can make it much easier to apply pattern-based security, following principles like defense-in-depth and complete mediation, yet still keeping security easy and scalable.

For any organization that's taking the leap to distributed architecture, putting software layers in place and leveraging pattern-based security will reap a lot of benefits and improve the lifetime and agility of the organization.

Azure's Defense-in-Depth Security Architecture

Defense-in-depth is a security risk management approach that defines multiple security layers in an IT environment, so that if a security attack is not caught by one layer, it will more likely be caught by another layer. Multiple security layers increase the overall security score of an environment and reduce the probability of a security breach.

With the rapid expansion of cloud computing, requirements for a defense-in depth strategy have increased dramatically. Cloud security is a shared responsibility between the customer and the cloud provider. Let's consider each security layer with respect to Azure:

- Physical security: Microsoft has its own Azure data centers and manages physical security at all locations. Only authorized persons should have access to different areas of the data centers. See Figure 1-14.

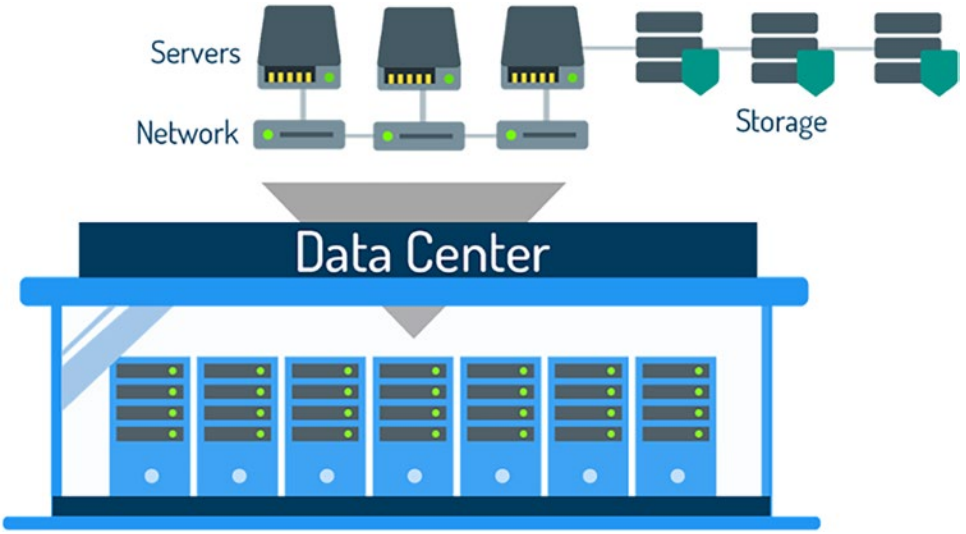

Figure 1-14. *Physical data centers*

- Identity and Access: All Azure resources are governed and controlled through the Azure Active Directory, which is the central place to manage all identities and related access. In addition to this, you can also manage access using role-based access controls. Certain users can also be assigned just-in time access to certain services to make it more secure using the Azure Privileged Identity management. See Figure 1-15.

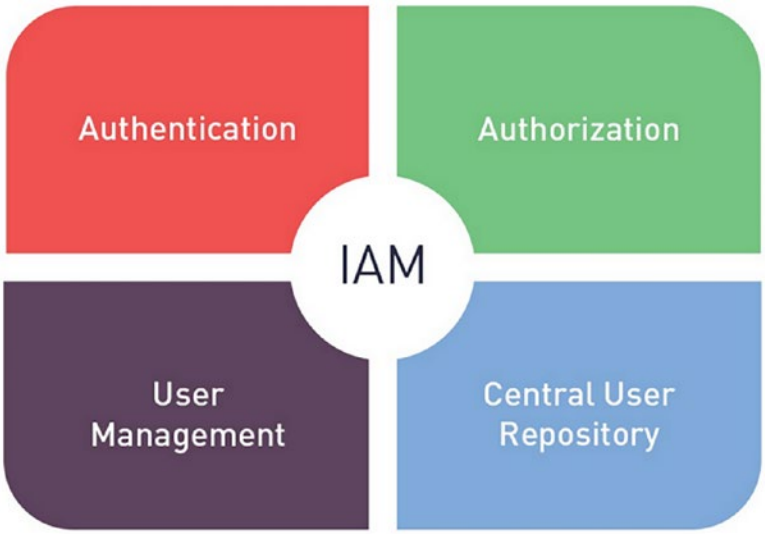

Figure 1-15. *Identity and access management with Azure AD*

- Perimeter: By default, Azure enables basic distributed denial of service, which comes with traffic monitoring and real-time mitigation of the network attacks. The standard tier of the DDoS provides additional capabilities to protect volumetric attacks.
- Network: You can filter network traffic from the Azure resources in a virtual network through the Network Security Groups (NSGs), which contain security rules allowing or denying traffic.

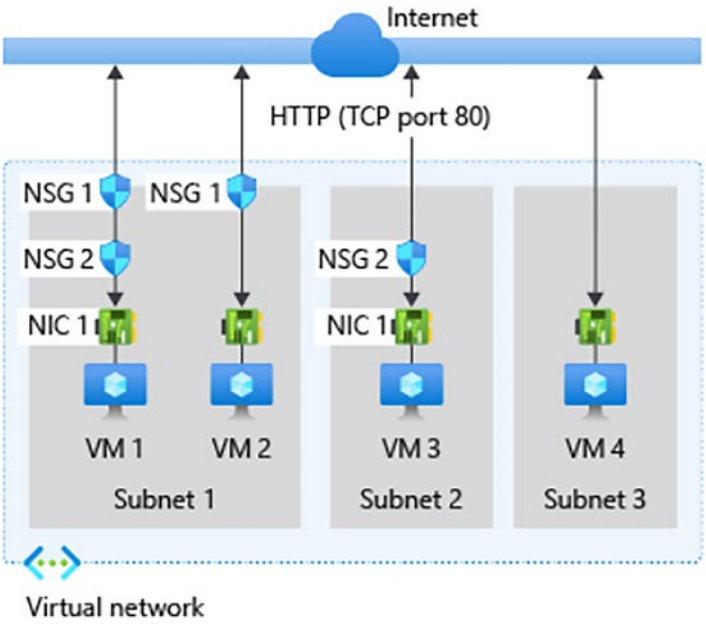

Figure 1-16. *Virtual network*

- Application Gateway: Application gateway and the web application firewall provide the centralized protection of the web application from vulnerabilities. Managed Identity from the Azure Active Directory allows the app to easily access the Azure Active Directory resources, such as Azure Key Vault.
- Data: Data is encrypted-at-rest for both structured and unstructured data. Using a combination of Azure Active Directory and Azure Key Vault, data can be encrypted and stored securely.

Looking back at the Open Systems Interconnection (OSI) model, there are seven layers. Possible threats and security measures can be applied to mitigate/control the threats.

OSI Layer	Protocols*	Used for	Possible threats	Security measures
Application Layer	Telnet, FTP, SMTP, DHCP, HTTP, SNMP, SMPP	Interaction at the user end with applications.	Backdoor attacks, static passwords, SNMP Private Community Strings	· Authentication/Access Control · Virus scanners · TLS encryption · Cryptographic Algorithm · Input Validation · Session Management
Presentation Layer	XDR, TLS, SSL , MIME	Encryption and Decryption	Virus, worms	
Session Layer	PPTP, RPC, SAP, L2TP, NetBIOS	Create a session between 2 nodes for efficient exchange of data.	Personal information retrieval, root privilege access, Net Bios attacks, RPC Attacks	
Transport Layer	TCP, UDP, IPX/SPX, DCCP, SCTP	This layer transmits data from source to destination node.	Endpoint identification, unauthorized Internet access , SYN flood, Ping of death.	· Private IP addressing via Network Address Translation (NAT) · Firewalls · Router Access Control Lists · Demilitarized zone (DMZ) · Proxy/ Application gateway
Network Layer	IPv4, IPv6, IPX, AppleTalk, ICMP, IPSec, IGMP	Real time processing and transfers data from nodes to nodes		
Data Link Layer	ARP, CSLIP, HDLC, IEEE.802.3, PPP, X-25, SLIP, ATM, SDLS, PLIP	Transforms the digital signals into frames.	ARP spoofing, MAC flooding, Spanning tree attack, 802.1Q and ISL tagging attack	· Tunneling · VPN connections
Physical Layer	Bluetooth, PON, OTN, DSL, IEEE.802.11, IEEE.802.3, L431, TIA 449	Hardware of networks such as cabling to transmit the digital signals.	Power loss or power spikes	· Uninterruptible power supply (UPS)

Figure 1-17. *Security layers, protocols, and possible threats*

Implementing security at every OSI layer will never protect against all cyberattacks. But the more the layers that are protected, the more hardened your application and system are from unintentional mishaps. Security is a continuous beast and it needs to be looked on with discipline and knowledge in order to close up any new open holes.

Although it is rather hard to project cost to implementing security measures, it should be done from a business mindset. This implies that mission-critical applications and data need to be better protected, as they can put an enterprise out of business; this risk should be addressed.

Conclusion

This chapter explored the various dimensions of cloud security. It explored in detail the history of security and the public cloud. It also took a deep dive into the boundaries and responsibilities to be considered when adopting cloud security. You also learned about the need for pattern-based security and about Azure's defense-in-depth security architecture.

This next chapter explains how to configure Identity and Access Management using Azure Active Directory with various features.