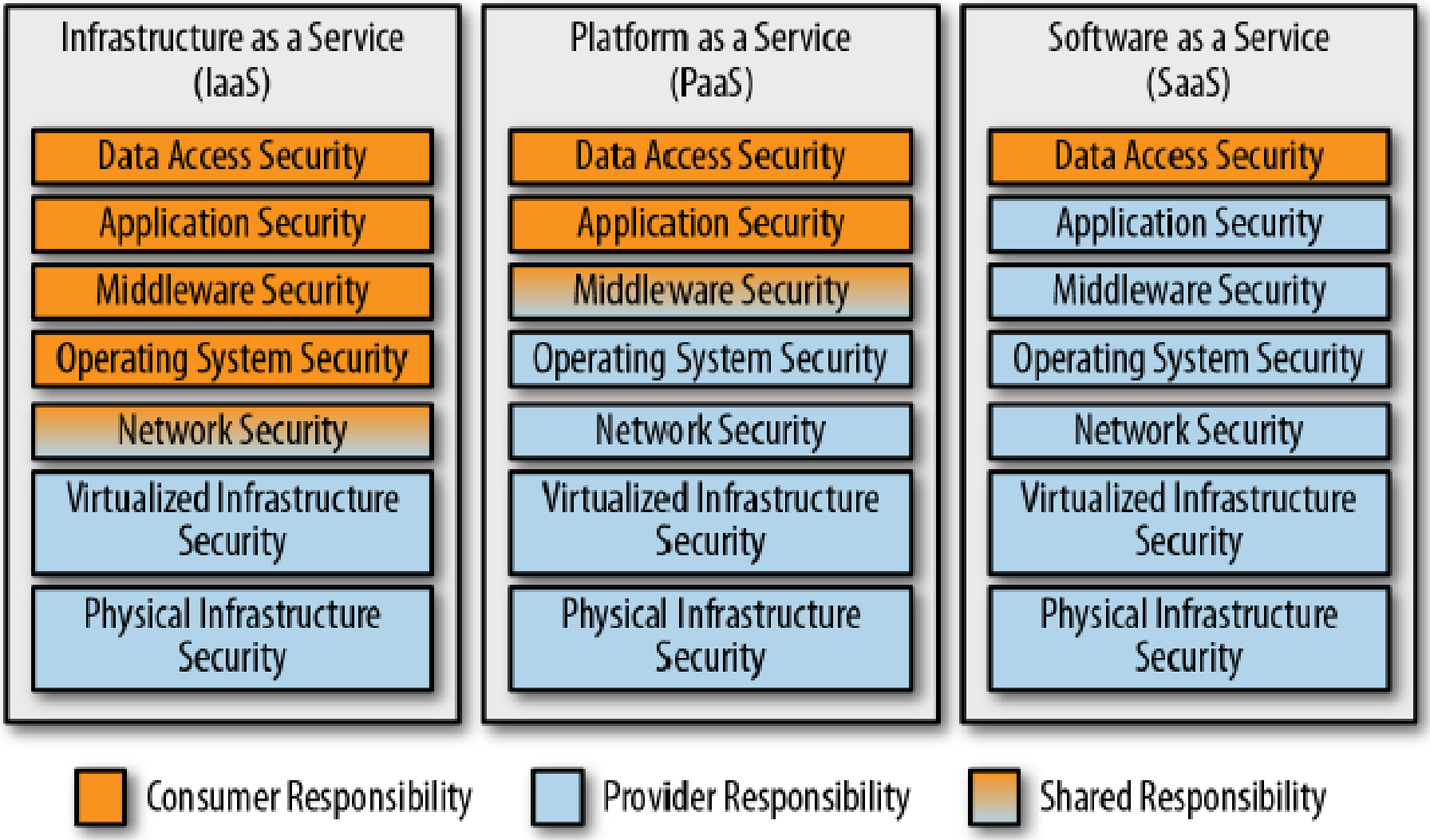
Security in Cloud Computing

Dr. Amit Praseed

Whose Responsibility is it?



Whose Responsibility is it?

- If the provider offers virtualized environments, the virtualized infrastructure security controls keeping your virtual environment separate from other virtual environments are the provider's responsibility.
  - Spectre and Meltdown vulnerabilities (2018)
- Operating system security is usually straightforward:
  - Your responsibility if you're using IaaS
  - Provider's responsibility if you're purchasing platform or software
- **If you have the ability to break it, you usually have the responsibility for securing it!**

Whose Responsibility is it?

- Root cause of Cloud Security issues is an assumption that the cloud provider is handling something, when it turns out *nobody* was handling it.
- AWS S3 storage is secure and encrypted, but none of that helps if you don't set your access controls properly.
  - Data on 198 million US voters
  - Auto-tracking company records
  - Wireless customer records
  - Over 3 million demographic survey records
  - Over 50,000 Indian citizens' credit reports
- Misunderstandings and Misinformation
  - 77% of IT decision makers believe that public cloud providers were responsible for securing customer data in the cloud
  - 68% said they believed these providers were responsible for securing customer applications as well

Data Asset Management

- Classify your data – low, medium and high security
  - Use tagging to keep track of data
- Understand security regulations and compliance
  - EU GDPR
  - US FISMA
  - Global PCI DSS

Cloud Data Protection

- Tokenization
  - store something that functions similarly to the data but is useless to an attacker
  - Eg: credit card numbers - replace a piece of sensitive data with a token
  - Token generally has the same characteristics as the original data, so underlying systems that are built to take that data don't need to be modified
  - Only one place (a "token service") knows the actual sensitive data.

Cloud Data Protection

- Encryption
  - Data can be in three states: in motion, use or rest
  - Encrypting Data in Use
    - Relatively new concept
    - requires support in the hardware platform, and it must be exposed by the cloud provider
    - encrypt process memory so that even a privileged cannot read it, and the processor can read it only when that specific process is running
    - Eg: Intel SGX, AMD SME, and IBM Z Pervasive Encryption.

Cloud Data Protection

- Encrypting Data at Rest
  - once you've encrypted the data, you now have an encryption key that can be used to access it
  - Hardware security module (HSM) to hold your encryption keys, usually in the form of an expansion card or a module accessed over the network
  - key management service (KMS), a multitenant service that uses an HSM on the backend to keep keys safe

Issues with KMS

- Simple Approach:
  - Use key management is to generate a key, encrypt the data with that key, stuff the key into the KMS, and then write the encrypted data to disk along with a note indicating which key was used to encrypt it
  - Problems?
    - Load on the KMS – too many keys
    - Erasure of Data
      - Delete the key – have to trust the KMS
      - Overwrite your data – time consuming

Issues with KMS

- Maintaining two keys
  - Data Encryption Key and Key Encryption Key
  - the key encryption key is used to encrypt (or “wrap”) data encryption keys, and the wrapped keys are stored right next to the data.
  - The key encryption key usually stays in the KMS and never comes out, for safety.
  - The wrapped data encryption keys are sent to the HSM for unwrapping when needed, and then the unwrapped keys are used to encrypt or decrypt the data
  - Delete the data? Delete the data encryption key!

Server-side and Client-side encryption

- Server Side Encryption
  - The storage service will automatically create data encryption keys, wrap them using a key encryption key that you can manage in the KMS, and store the wrapped keys along with the data.
  - Multitenant storage service does have the ability to decrypt your data!!!
- Client Side Encryption
  - Encryption and Decryption handled by client
  - No server-side searches, calculation, indexing, malware scans, or other high-value tasks can be performed

Homomorphic Encryption

- Homomorphic encryption is a method of encryption that allows any data to remain encrypted while it's being processed and manipulated.
  - Enables you or a third party (such as a cloud provider) to apply functions on encrypted data without needing to reveal the values of the data.
- Uses a public key to encrypt data and allows only the individual with the matching private key to access its unencrypted data
  - It uses an algebraic system to allow you or others to perform a variety of computations (or operations) on the encrypted data.

Homomorphic Encryption can solve Real World Problems!!!

- **Securing Data Stored in the Cloud**
- **Enabling Data Analytics in Regulated Industries**
- **Improving Election Security and Transparency**

Types of Homomorphic Encryption

- **Partially homomorphic encryption (PHE)**
  - allows select mathematical functions to be performed on encrypted values
  - one operation can be performed an unlimited number of times on the ciphertext.
  - Some examples of PHE include ElGamal encryption (a multiplication scheme) and Paillier encryption (an addition scheme).
- **Somewhat homomorphic encryption (SHE)**
  - supports limited operations (for example, either addition *or* multiplication) up to a certain complexity
  - These operations can only be performed a set number of times.
- **Fully homomorphic encryption (FHE)**
  - still in the development stage
  - capable of using any efficiently computable functions (such as addition *and* multiplication, not just one or the other) any number of times
  - makes secure multi-party computation more efficient.
  - Practically extremely slow