INFO 7350 - Systems and Cybersecurity Fundamentals Spring 2021

CLOUD INFRASTRUCTURE NETWORK SECURITY

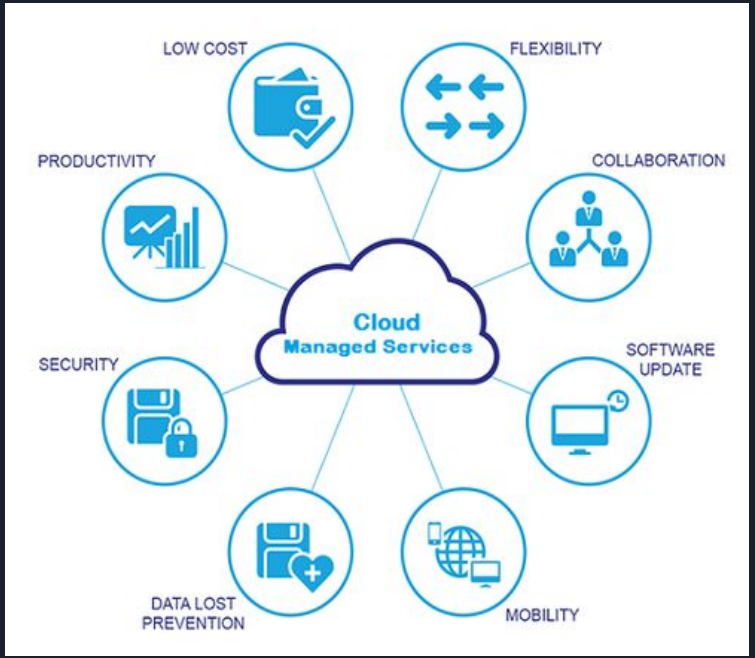
Guided By:
Prof. Angel L. Hueca
Northeastern University

TEAM - 1

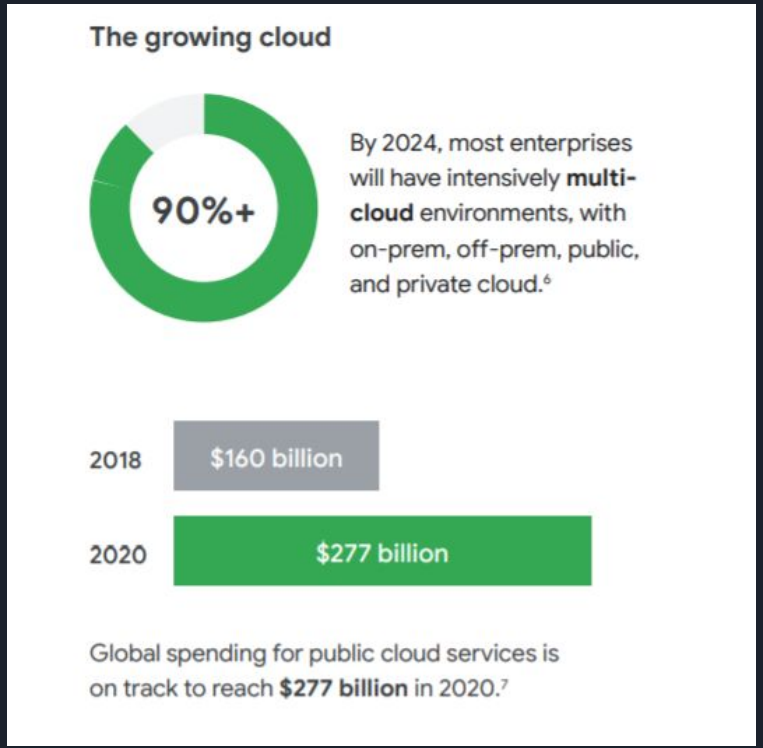
Hemant Jain
Mahesh Vatalu Renukaprasad
Sapna Patel
Krutarth Vanesa
Vivek Shakya

Table of Contents

Cloud Computing

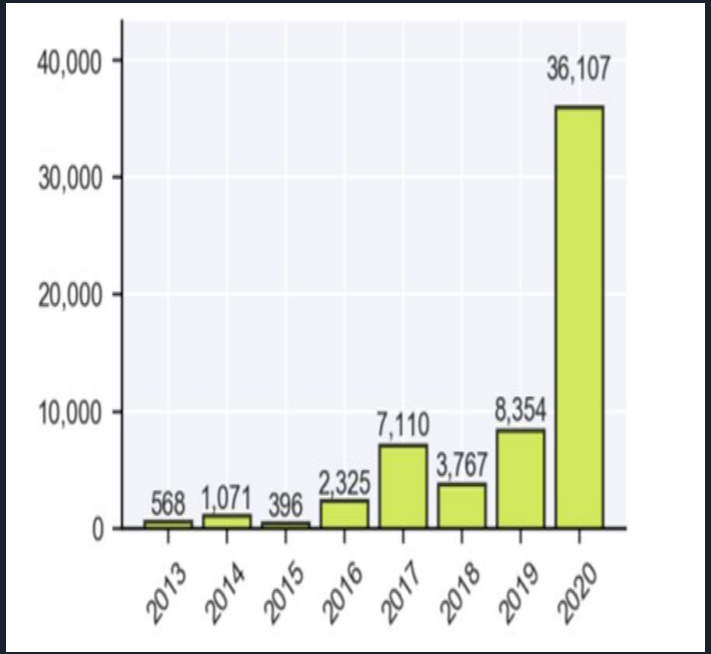


Cloud Security - Why?



How pandemic change Cyber Security

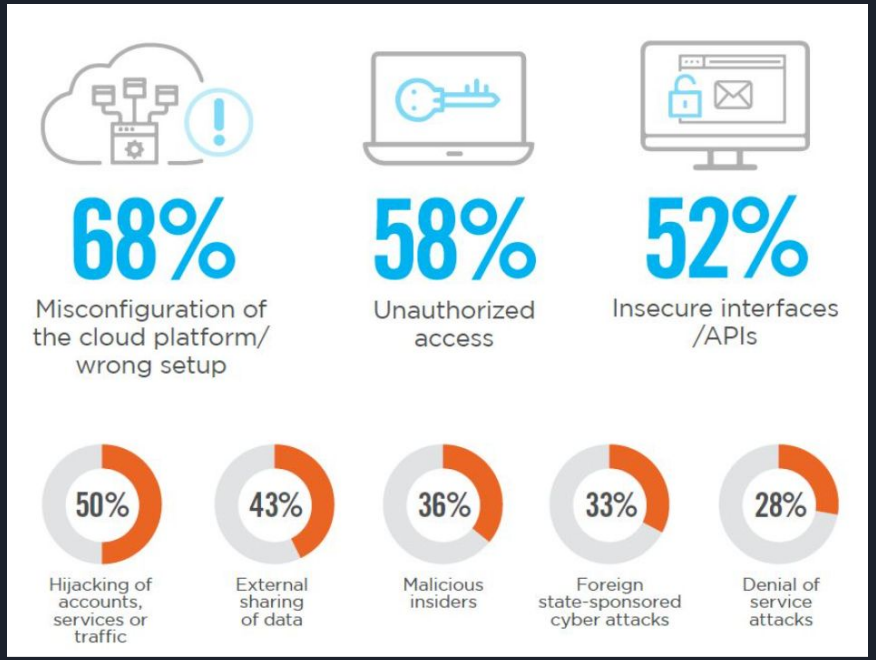
- In the first three quarter of 2020, 21% of reported breaches involved the use of ransomware.
- These ransomware-related events contributed to the unusually high number of unknown (11.2%) and miscellaneous (10.4%) data types exposed.
- Following well established trends, the Healthcare sector had the most reported breaches, accounting for 11.5% of the events that could be attributed to a specific economic sector.



Understanding the problem

It inherits the security issues which persist in those individual platforms involving

1. Secure access
2. Authentication
3. Authorization
4. Access management
5. Security controls and services
6. Misconfigurations
7. Access Control Vulnerabilities
8. Data-theft and cross-control access



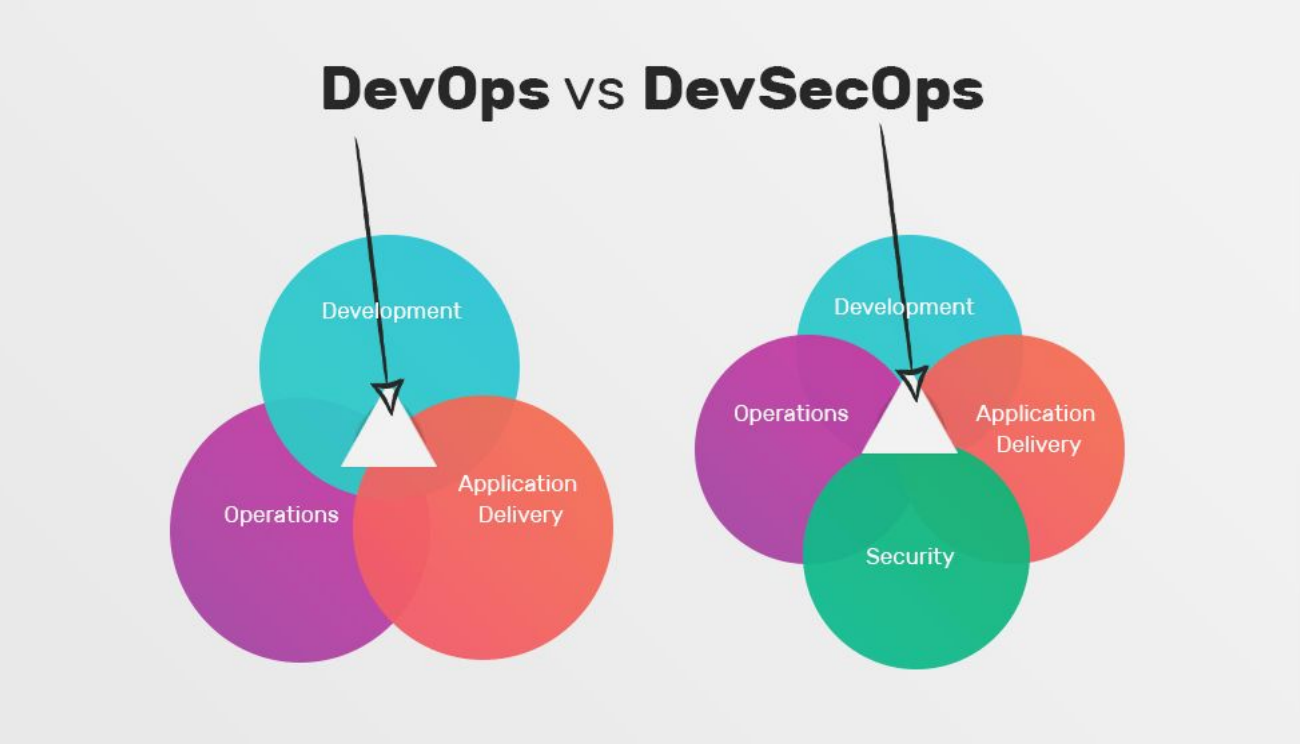
Project objective

The main objective of the research is to identify the most common issues that organizations consider when migrating their infrastructure from on premise to cloud and explore the security services and policies that can make the cloud infrastructure more secure and reduce the vulnerabilities at the architecture level.

Target audience

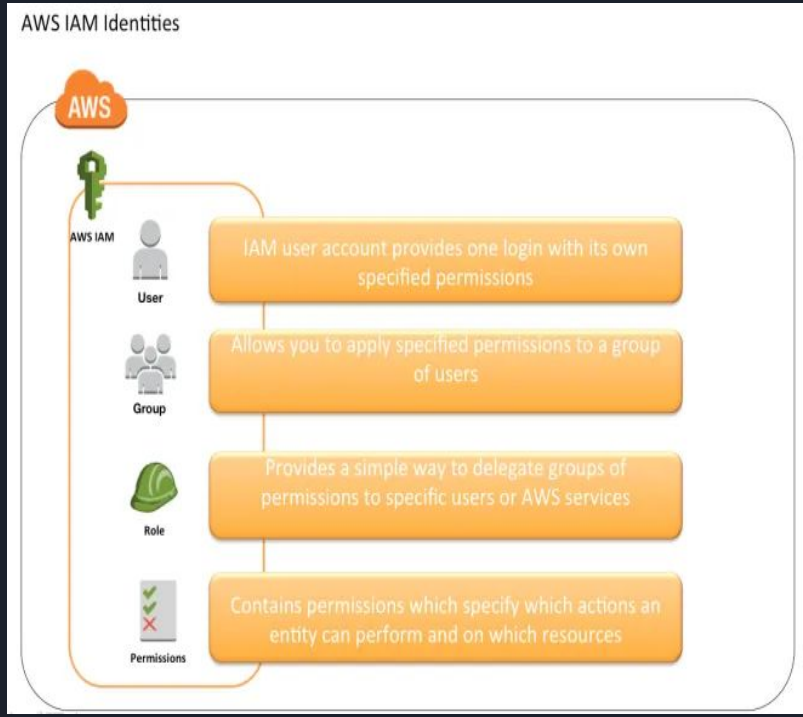
- Organizations looking to migrate their infrastructure on cloud
- Organizations looking to start their infrastructure on cloud
- Developers interested in hosting applications on cloud

DevOps and DevSecOps



Identity, Access and Management (IAM)

- Manage users, groups and roles
- Create suitable roles for different cloud resources
- Manage user permission by designing relevant IAM policies
- Follow the principle of least privilege while assigning permissions or creating IAM policies
- Can be integrated with your own user repository

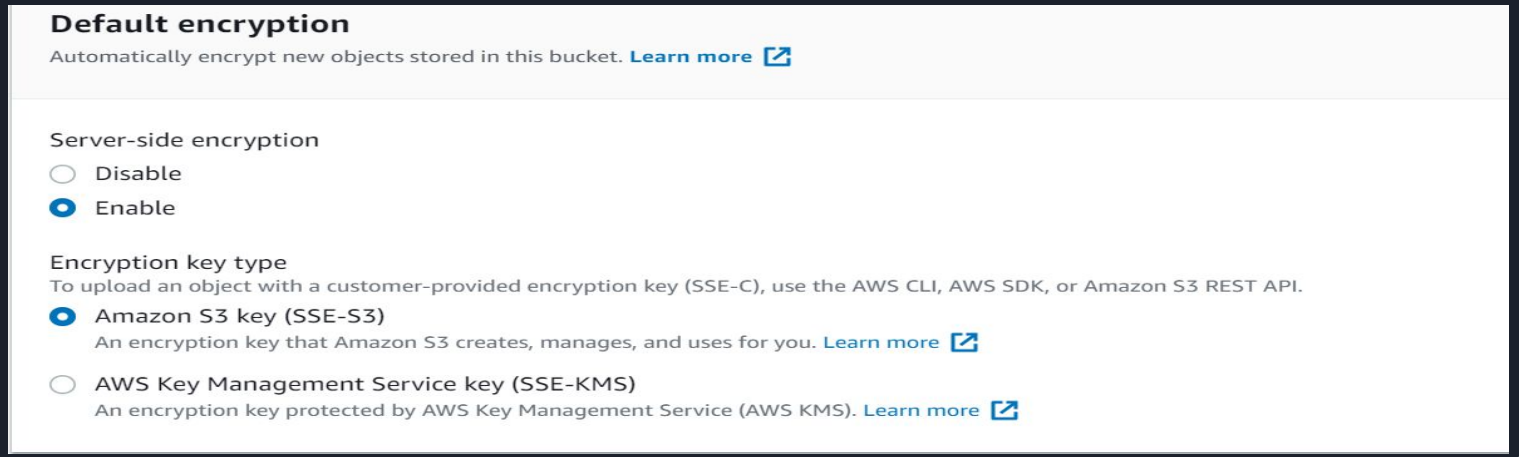


Best practices to be followed for IAM credentials from security perspective

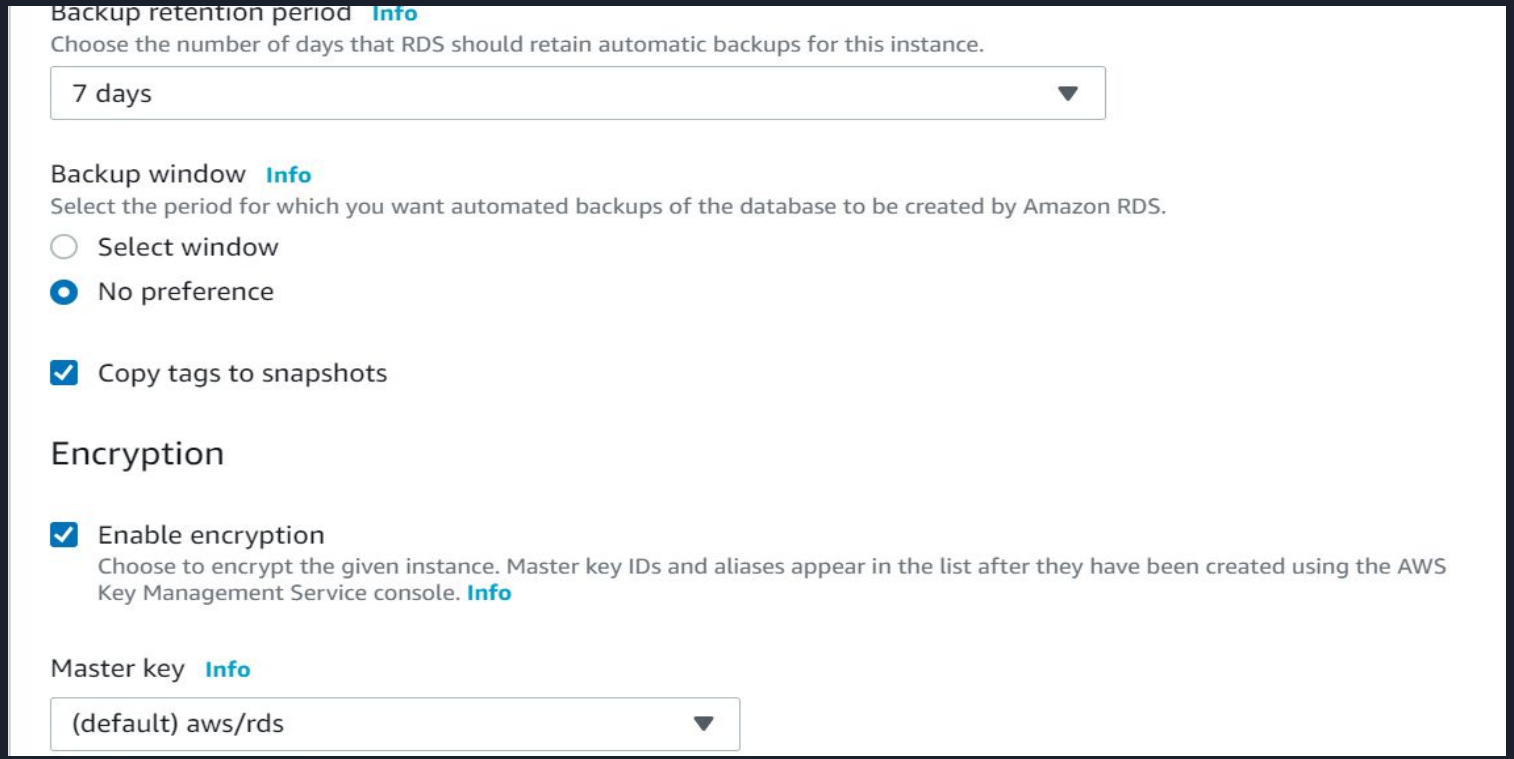
- One IAM role per person
- Never commit or share IAM credentials
- Never use root credentials

Encrypting Data at rest and in transit

- Make use of the key management service to encrypt your data
- Cloud providers offer key management which is completely managed by them or you can also use custom keys which are managed solely by the customer
- The data keys used to encrypt your data are also encrypted and stored alongside the data that they protect



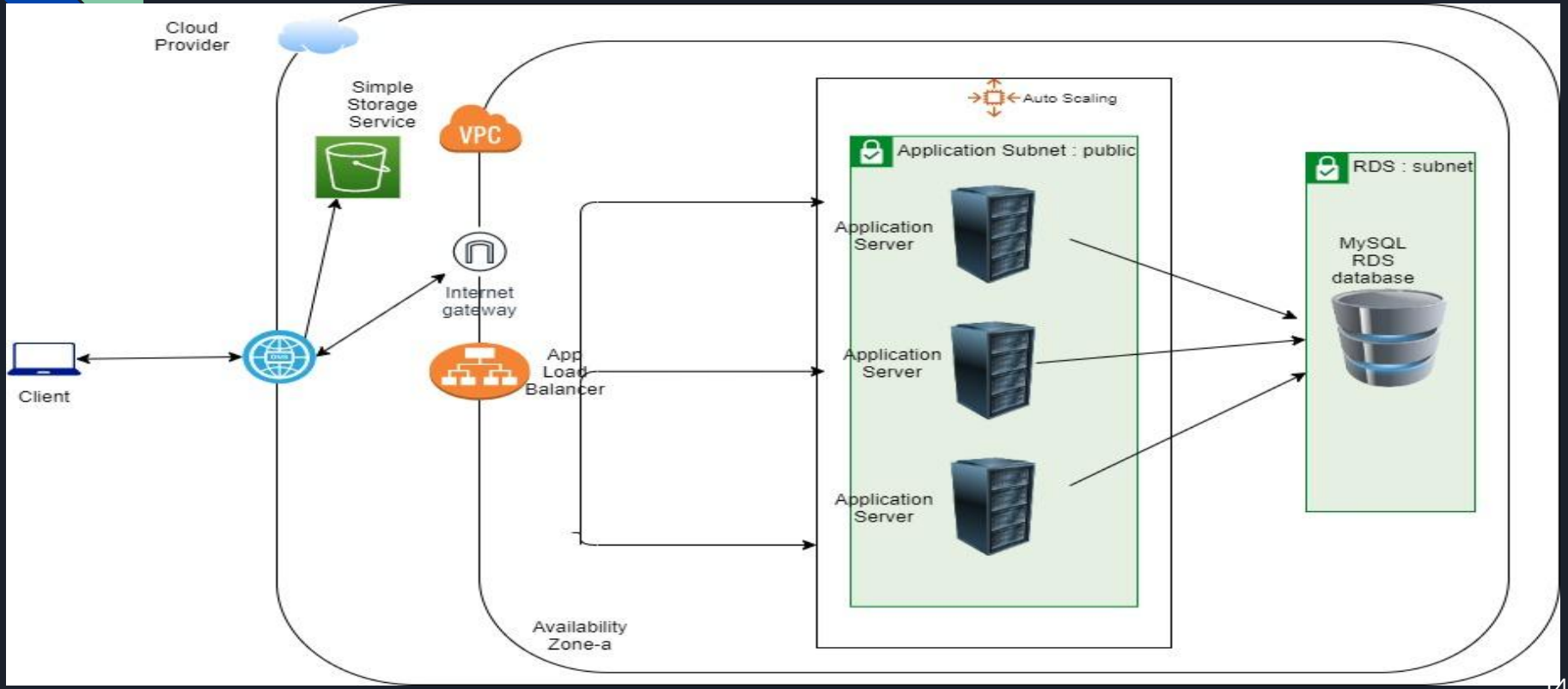
Encrypting data residing in a Relational database storage



Methodology / Security Approach

- **Understand the design shift from on premise to cloud**
- **Consider security as an architectural strategy**
- **Control access privileges**
- **Use a risk based approach**
- **Focus on monitoring and logging**

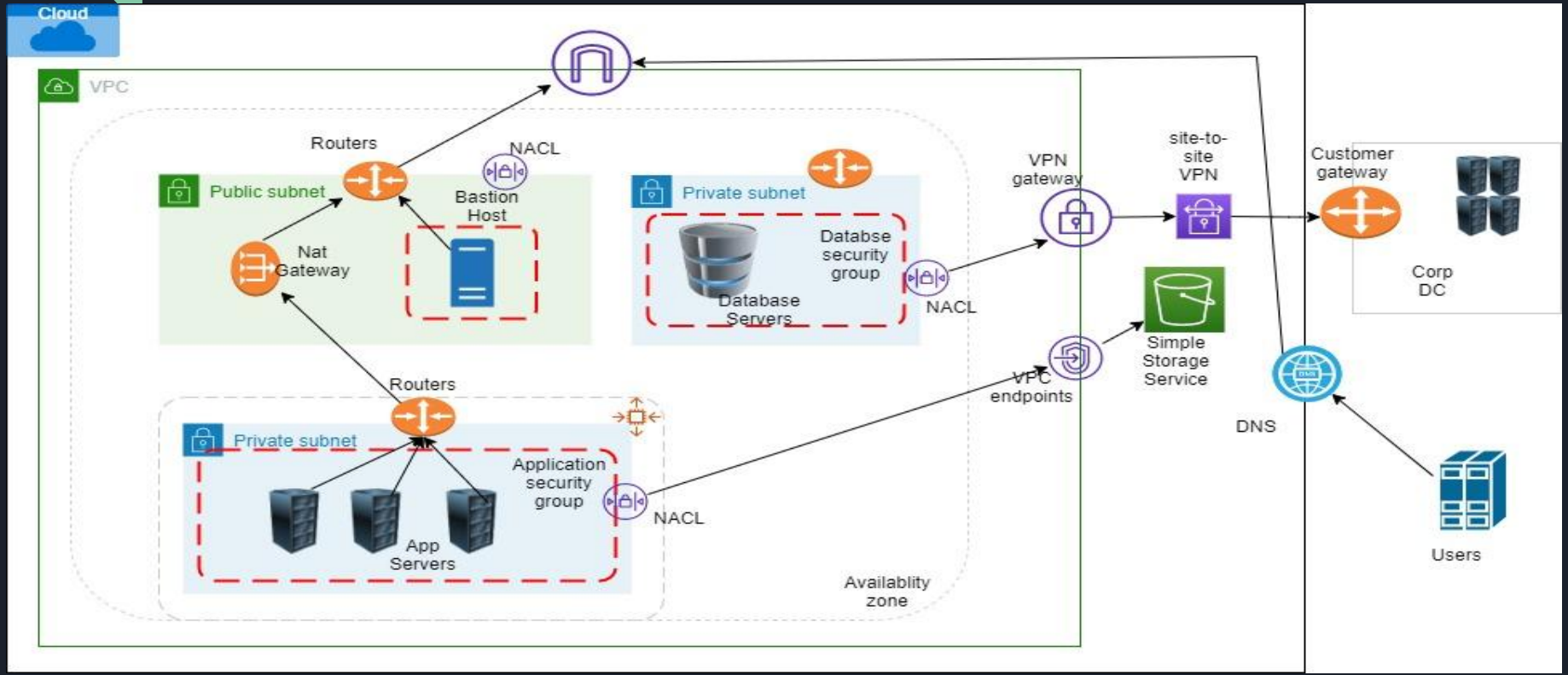
Model 1 - weak architecture design



Cloud resources leveraged for model 1

- VPC - isolate your cloud resources, need to assign a CIDR range
- Public subnets - falls inside the CIDR range of the VPC CIDR
- Web Servers - linux instance which host the web application
- Relational Database server which serves the requests made from web servers
- Default security groups which allows access to linux instances
- Routing tables - stores route from web and database servers to the internet gateway
- Internet gateway- allows access to public internet via DNS
- DNS - domain name server which resolves the host names

Model -2 showcasing security at architecture level



Cloud resources leveraged for model 2

- **VPC** - virtual network to isolate your cloud resources
- **Public subnets** - host public instance and NAT gateways which in turn allow access to private instances
- **Private subnets** - helps to enable a layered security approach by assigning private ip's to your resources
- **NAT gateway** - allows access to compute instances inside private subnets , must be placed in a public subnet
- **Security groups** - enable access rules for specific protocols and their associated ports, stateful components
- **Network Access list** - implements another layer of security placed before security groups, stateless components(need to specify inbound and outbound rules separately)

- **VPC/service endpoints** - provides secure and direct connectivity to cloud resources, allows private IPs to reach the cloud resources
- **Site-to-site VPN** - creates a secure connection between corp. data center or remote location to your cloud resources using IP Security (IPSec) tunnels
- **VPN gateway** - sends encrypted traffic from cloud network to the on premise data center
- **Customer gateway** - physical device that forms the part of the IPSec tunnel with VPN gateway when utilizing the site-to-site VPN connection

Why Infrastructure as Code and why we need to secure it?

- These Infrastructure as Code scripts are can be identified beneficial with the example of Fortune 500 company Intercontinental Exchange (ICE).
- ICE uses IaC scripts to maintain 75% of its 20,000 servers
- These results in a time reduction of environment provisioning from one or two days to 21 minutes.
- Defects in IaC scripts can have serious consequences, as these scripts are associated with setting up and managing cloud-system infrastructure and ensuring the availability of client side services.
- For example, in early 2017, execution of a defective IaC script removed the home directories of around 270 customers in cloud instances maintained by Wikimedia.

Best Practices to Secure IaC

- Manual Security Assessment:

  This step involves reviewing the architecture/templates before they are deployed and inspecting the live infrastructure after deployments.

- Codify everything:

  All of the infrastructure specifications should be coded in configuration files, such as AWS CloudFormation templates. Infrastructure should be deployed seamlessly, and no one should log into a server to manually make changes.

Best Practices to Secure IaC (Continued)

Continuously test system, integrate and deploy: When changes are made automated tests verify that there were no security and compliance regressions before deploying the changes to a production server. All of this is managed by an automated CI/CD pipeline. Unit tests for configuration code should include security checks such as the following:

- Disable unnecessary services.
- Close unused ports.
- Look for hardcoded credentials and secrets.
- Check and review permissions on files and directories.
- Ensure that development tools are not installed in production servers.
- Check auditing and logging policies and configurations.

Best Practices to Secure IaC (Continued)

- Version everything:

  The config files should be managed in a version control system. Because all configuration data are written in code, any modifications to the code can be controlled and tracked. Any changes in infrastructure will be performed by changes committed to the VCS. In addition, VCS is important for IaC because it provides the following functions:

- Traceability: Record all changes that have been made.
- Rollback: Restore things back in case of any failures.
- Correlation: Useful for tracing and fixing complex problems when these occur.
- Visibility: All team members can see when changes are committed to VCS.
- Actionability: VCS can trigger actions automatically when a change is committed.

Best Practices to Secure IaC (Continued)

- Modular code (small changes rather than big batches):

  Smaller changes make errors easier to detect and allow the team to fix them.

  - Easier and less effort in case of testing the changes and evaluation.
  - Faster to find the cause of the bugs and errors, then easy to fix them.

- Immutable infrastructure:

  IT infrastructure elements are replaced for each deployment, instead of making changes to current infrastructure. This process provides consistency, avoids bugs in previous infrastructure and restricts the impact of being breached. It improves security and makes fixes easier.

Best Practices to Secure IaC (Continued)

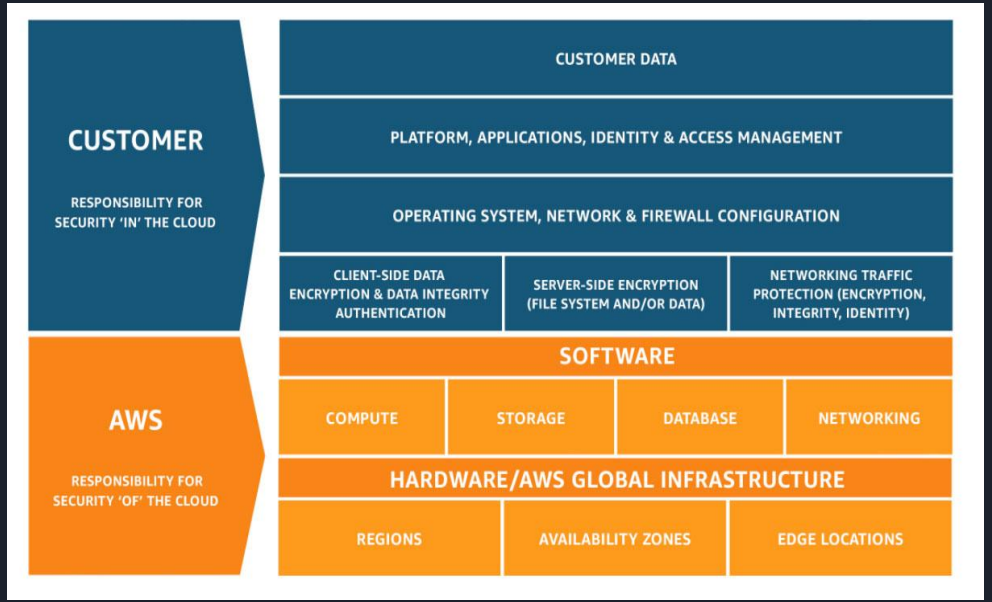
Continuous security and service availability:

- It's important to secure not only the application and its runtime environment but also the CI/CD pipeline and test environment.
- It is important to protect the pipeline from attacks by ensuring that all changes are fully transparent and traceable from end to end.

Shared Responsibility Model

Clients should use some of features present within cloud infrastructure to protect their resources like:

- Security Groups
- Private Subnets
- CORS (Cross-Origin Resource Sharing)
- Web ACLs
- Firewalls
- Encryption
- Setting up roles, policies and users etc

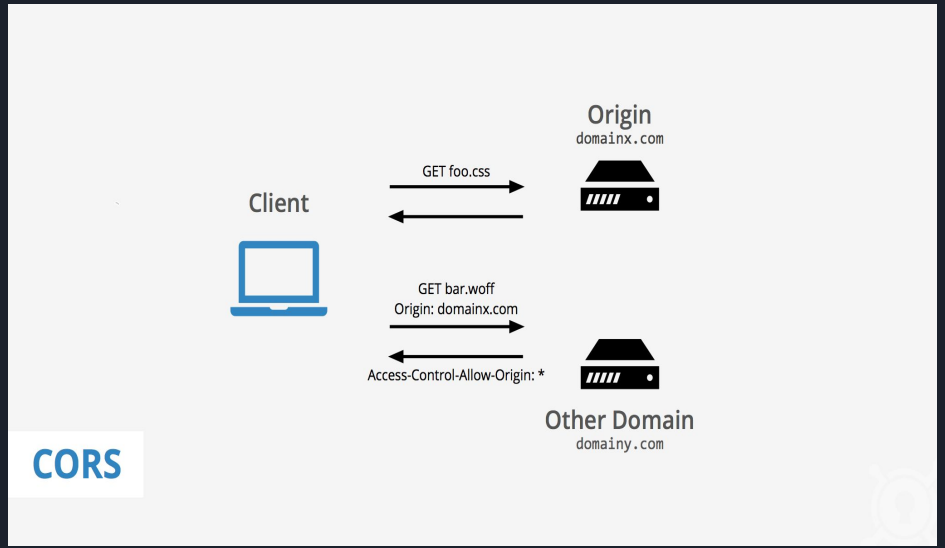


Securing your APIs

CORS (Cross-Origin Resource Sharing)

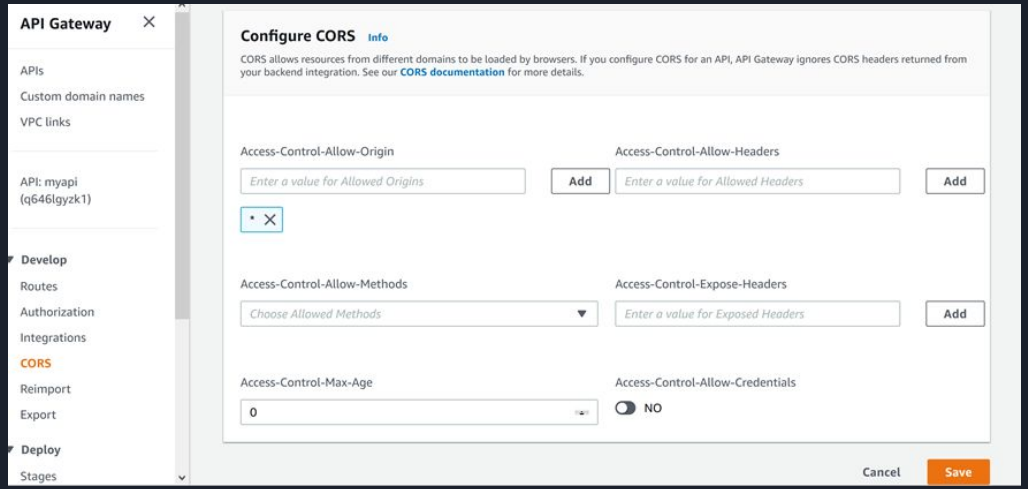
CORS is a mechanism that allows restricted resources on a web page to be requested from another domain outside the domain from which the first resource was served.

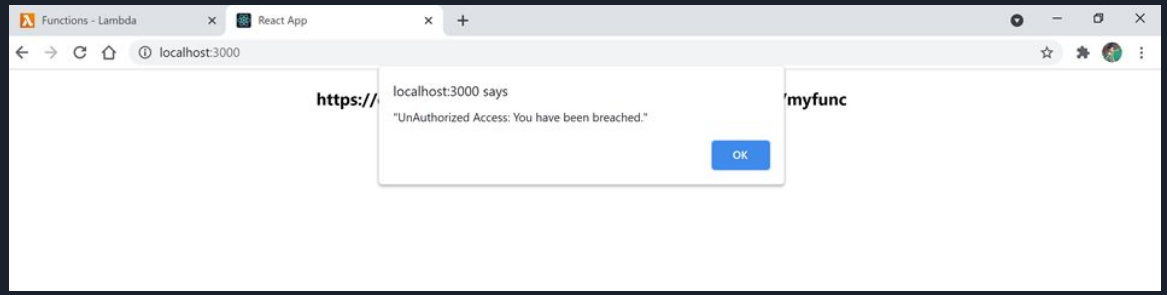
The benefit of CORS is that it allows your domain to allow reads from another trusted domain.



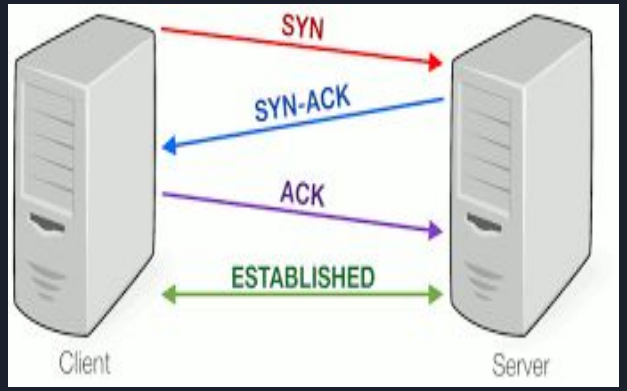
Configuring CORS in API Gateway in AWS

Misconfiguring CORS with “*” Value

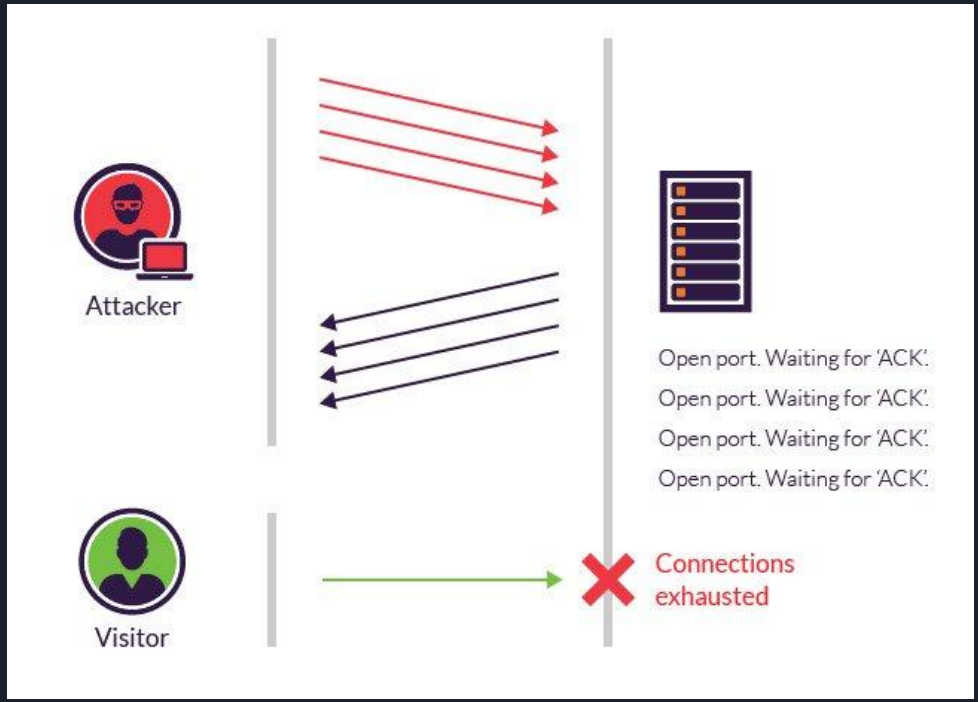




SYN Flood Attack

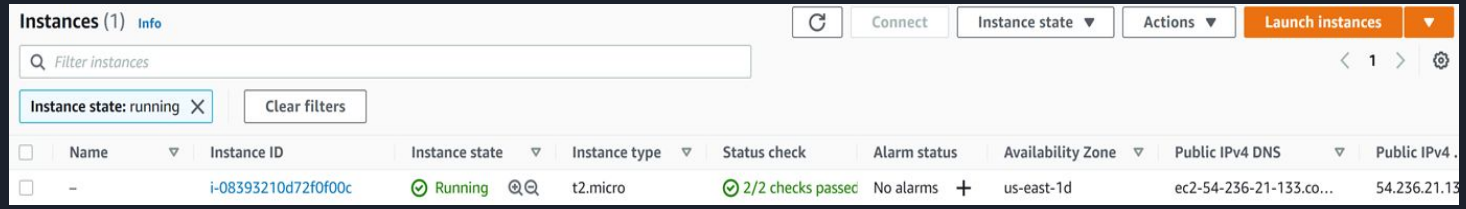


TCP HandShake



SYN Flood Attack

SYN Flood Attack using Metasploit



Outbound rules (1)

Type	Protocol	Port range	Destination
All traffic	All	All	0.0.0.0/0

Inbound rules | Outbound rules | Tags

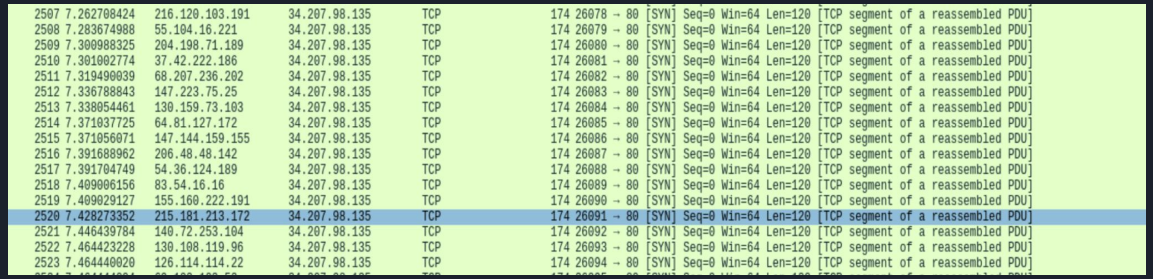
Inbound rules (3)

Type	Protocol	Port range	Source
All TCP	TCP	0 - 65535	0.0.0.0/0
SSH	TCP	22	0.0.0.0/0
All ICMP - IPv4	ICMP	All	0.0.0.0/0

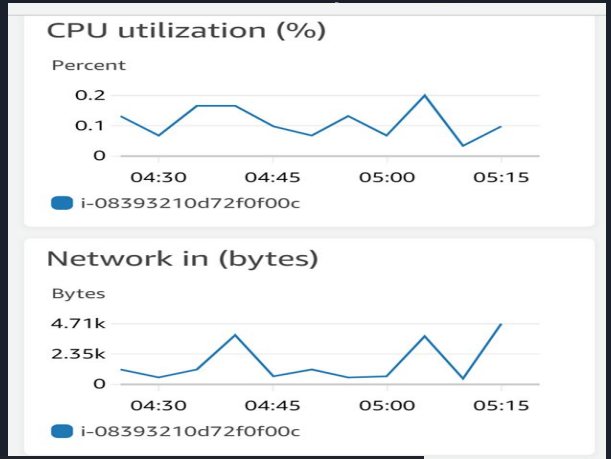
```
root@kali:~# ssh -i "myssh.pem" ec2-user@ec2-54-236-21-133.compute-1.amazonaws.com
Last login: Mon Apr 26 02:13:53 2021 from c-67-189-243-101.hsd1.ma.comcast.net

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
1 package(s) needed for security, out of 15 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-30-157 ~]$ iostat
Linux 4.14.225-169.362.amzn2.x86_64 (ip-172-31-30-157.ec2.internal)     04/26/2021     _x86_64_     (1 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.17    0.03    0.11    0.10    0.11   99.48

Device:            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
xvda              2.15        19.07        26.55     236020     328592

[ec2-user@ip-172-31-30-157 ~]$
```

SYN Flood Attack (Continued...)

SYN Flood Attack Result



```
2507 7.262708424   216.120.103.191   34.207.98.135   TCP   174 26078 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled PDU]
2508 7.283674988   55.104.16.221     34.207.98.135   TCP   174 26079 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled PDU]
2509 7.300988325   204.198.71.189    34.207.98.135   TCP   174 26080 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled PDU]
2510 7.301002774   37.42.222.186     34.207.98.135   TCP   174 26081 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled PDU]
2511 7.319490039   68.207.236.202    34.207.98.135   TCP   174 26082 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled PDU]
2512 7.336788843   147.223.75.25     34.207.98.135   TCP   174 26083 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled PDU]
2513 7.338054461   130.159.73.103    34.207.98.135   TCP   174 26084 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled PDU]
2514 7.371037725   64.81.127.172     34.207.98.135   TCP   174 26085 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled PDU]
2515 7.371056071   147.144.159.155   34.207.98.135   TCP   174 26086 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled PDU]
2516 7.391688962   206.48.48.142     34.207.98.135   TCP   174 26087 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled PDU]
2517 7.391704749   54.36.124.189     34.207.98.135   TCP   174 26088 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled PDU]
2518 7.409006156   83.54.16.16       34.207.98.135   TCP   174 26089 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled PDU]
2519 7.409029127   155.160.222.191   34.207.98.135   TCP   174 26090 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled PDU]
2520 7.428273352   215.181.213.172   34.207.98.135   TCP   174 26091 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled PDU]
2521 7.446439784   140.72.253.104    34.207.98.135   TCP   174 26092 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled PDU]
2522 7.464423228   130.108.119.96    34.207.98.135   TCP   174 26093 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled PDU]
2523 7.464440020   126.114.114.22    34.207.98.135   TCP   174 26094 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled PDU]
```




```
root@kali:~# ping 34.207.98.135
PING 34.207.98.135 (34.207.98.135) 56(84) bytes of data.
From 10.0.2.15 icmp_seq=1 Destination Host Unreachable
From 10.0.2.15 icmp_seq=2 Destination Host Unreachable
From 10.0.2.15 icmp_seq=3 Destination Host Unreachable
From 10.0.2.15 icmp_seq=4 Destination Host Unreachable
From 10.0.2.15 icmp_seq=5 Destination Host Unreachable
From 10.0.2.15 icmp_seq=6 Destination Host Unreachable
From 10.0.2.15 icmp_seq=21 Destination Host Unreachable
From 10.0.2.15 icmp_seq=22 Destination Host Unreachable
From 10.0.2.15 icmp_seq=23 Destination Host Unreachable
From 10.0.2.15 icmp_seq=24 Destination Host Unreachable
From 10.0.2.15 icmp_seq=25 Destination Host Unreachable
From 10.0.2.15 icmp_seq=26 Destination Host Unreachable
```

```
File  Actions  Edit  View  Help
root@kali:~# ssh -i "myssh.pem" ec2-user@ec2-34-207-98-135.compute-1.amazonaws.com
```

Materials and Methods

Cloud security is a set of policies, controls, procedures and technologies that should work together to protect your cloud-based applications and systems.

Data Integrity and Protection

- One of the biggest concerns which disinclined the trust and authenticity on the Cloud Infrastructures
- One of the best ways to mitigate this threat is to secure your data using in-transit and at-rest data security.
- This would include the use of encryption both for email server and for the messages themselves.
- This would include the use of digital certificates such as SSL/TLS website certificates and S/MIME

Materials and Methods (Continued…)

Lack of cloud security architecture and strategy

- Keep threat models up to date.
- Deploy continuous monitoring capability

Insufficient identity, credential, access and key management

- Inadequately protected credentials
- Lack of automated rotation of cryptographic keys, passwords and certificates
- Lack of scalability
- Failure to use multi-factor authentication
- Failure to use strong passwords

Materials and Methods (Continued...)

Account hijacking

- Don't just do a password reset when account credentials are stolen. Address the root causes.
- A defense-in-depth approach and strong IAM controls are the best defense.

Insider threats

- Conduct employee training and education on proper practices to protect data and systems. Make education an ongoing process.
- Regularly audit and fix misconfigured cloud servers.
- Restrict access to critical systems

Insecure interfaces and APIs

- Employ good API practices such as oversight of items like inventory, testing, auditing and abnormal activity protections.
- Protect API keys and avoid reuse.
- Consider an open API framework such as the Open Cloud Computing Interface (OCCI)

Results of Cloud Infrastructure Network and the Infrastructure Security

- Protection against DDoS
  This entails monitoring, absorbing and dispersing DDoS attacks to minimize risk.

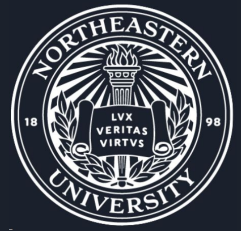
- Data security
  This prevents a third party from eavesdropping or tampering with data being transmitted.

- Regulatory compliance
  Helps in managing and maintaining enhanced infrastructures for compliance and to protect personal and financial data.

- Flexibility
  When the high traffic is over, you can scale back down to reduce costs.

- High availability and support
  Live monitoring 24 hours a day, 7 days a week, and every day of the year.

Things to do to prevent hacks and breaches



- Companies should embrace automation and use technologies that continuously scan for misconfigured resources and remediate problems in real time.
- Develop and implement a security architecture framework.
- Ensure that the threat model is kept up to date.
- Bring continuous visibility into the actual security posture.
- Take measures to minimize insider negligence to mitigate the consequences of insider threats
- Practice good API hygiene. This includes the diligent oversight of items such as inventory, testing, auditing, and abnormal activity protections.
- All cloud providers should conduct penetration testing and provide findings to customers.

Recommended Configurations to follow while using Infrastructure as Code:

Permissions to IaC User:

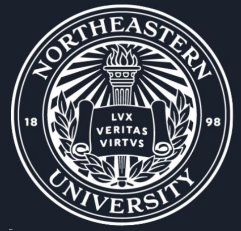
- The deployment user/role should only have write-only access to cloud servers during the deployment for the application.
- This will restrict the intruder from getting the secrets or user's data from the server if it got access to the cloud infrastructure via IaC.

Path configuration:

- it's important to save all paths in the config file and encrypt that file.
- This will protect the file as well as saves time and effort in the case of any path issues.
- So during the breach attempt, it will save time to reconfigure paths and make deployment available back again in less amount of time.

Conclusion

- Cloud Security is a layered architectural approach
- Misconfigurations should be avoided at all cost.
- Identifying the risk points in the architecture helps to identify the security solution
- Security approach can be same as on premise but keep the infrastructure shift in mind while designing the cloud architecture



Thank you!

Any Questions