

# ASSESSMENT REPORT FOR HEALTH CARE INFRASTRUCTURE

## Table of contents

### VULNERABILITY ASSESSMENT REPORT

1. Executive Summary
2. Scan results
3. Findings
4. Current State analysis, Risks and Remediations
5. Conclusion
6. References

### DATA SECURITY ASSESSMENT REPORT

1. Executive Summary
2. Scan results
3. Findings
4. Assessment Report
5. Vulnerabilities, Remediation and Recommended Configuration
6. Conclusion
7. References

### VIRTUAL MACHINE VULNERABILITY ASSESSMENT REPORT

1. Executive Summary
2. Scan results
3. Findings
4. Assessment Report
5. Remediations, Security Policy and Recommended Configuration
6. Conclusion
7. References

## NETWORK SECURITY ASSESSMENT REPORT

1. Executive Summary
2. Scan results
3. Findings
4. Current State analysis, Risks and Remediations
5. Conclusion
6. References

## DISASTER RECOVERY ASSESSMENT REPORT

1. Executive Summary
2. Scan results
3. Findings
4. Current State analysis, Risks and Remediations
5. Configuration best practices for current infrastructure
6. Conclusion
7. References

# VULNERABILITY ASSESSMENT REPORT

## Table of Contents

1. Executive Summary
2. Scan results
3. Findings
4. Current State analysis, Risks and Remediations
5. Conclusion
6. References

## 1. Executive Summary

The project entails evaluating the cloud infrastructure security of a healthcare organization, which is essential for handling and storing sensitive patient data. This contains billing information, personal information, and medical records. Maintaining the application's confidentiality, integrity, and availability as well as detecting and mitigating any security issues are the primary goals. The evaluation will address several cloud security topics, including controlling user access, encrypting information, protecting the network, and handling vulnerabilities.

The healthcare organization uses the cloud extensively to host vital healthcare apps, store patient data, and do telemedicine consultations. Regrettably, because of setup errors and security lapses, the infrastructure is extremely vulnerable to possible cyberattacks. The objective is to identify and take advantage of these vulnerabilities to highlight the dangers connected to the unsecure cloud configuration and offer suggestions for fixing it. This project is essential for protecting private patient data and guaranteeing the quality of the medical care given.

## 2. Scan Results

We have included the Vulnerability assessment report that is consolidated from the scans of AWS with an open-source tool “prowler”

**Scan Results** - The scan results provide granular detail of each vulnerability, which are categorized by their severity: critical, high, medium, and low. An expanded definition of the known threat and solutions for remediating the vulnerability are also available.

## 3. Findings

Severity	Count
Critical	11
High	3
Medium	3

## 4. Current State analysis, Risks and Remediations:

The current infrastructure found to be very vulnerable with no mandatory best practices implemented. It is observed that the infrastructure is vulnerable due to misconfigurations in accounts, users, organizations, key handling, rotation, encryption and more. A detailed information of those vulnerabilities is enclosed in the below table.

Critical vulnerabilities in the current infrastructure:

S.NO	Vulnerability	Risks
1	MFA Not enabled for root account	Account compromise, Data exfiltration, Exploitation of services, reputational damage

2	Hardware MFA not implemented	Account compromise, Data exfiltration, Exploitation of services, reputational damage
3	Secrets are present in cloud formation outputs in one of the stacks	Lateral Movement, privilege escalation, Data exfiltration, further compromise to services
4	MFA not configured for IAM users	Data exfiltration, Exploitation of services, account prone for brute forcing
5	Overly Permissive IAM roles	Data Breaches, Insider threats, unnecessary misconfigurations, and downtime
6	A custom policy named "Developer role" is implemented with administrator access	Account compromise, Privilege escalation, lateral access to other services, Insider threats
7	Guest role has access to the S3 buckets with public access	Data Exfiltration, sensitive data exposure, lateral access to other services
8	Default credentials for databases stored in cloud formation template	Hardcoded credentials might be used to gain lateral access to other services in the healthcare infrastructure.
9	IMDSv2 is disabled on the instance/IMDSv1 enabled on the instance	Enable IMDSv2 and disable IMDSv1
10	Security Groups are open to internet 0.0.0.0/0 on port 22	Restrict the access to port 22 from only specific IP prefixes
11	Encryption not implemented	Enable encryption on data wherever needed

#### MFA Not enabled for root account and Hardware MFA not implemented:

**Vulnerability:** Root account in HealthCare infrastructure has the privileged access for every service. This functionality makes this account critical and must be configured as per best practices.

**Risks:** Misconfiguration in root's MFA would result in complete compromise of the HealthCare infrastructure.

**Mitigation:** Disable root account/Enable MFA preferably using hardware keys for root account

#### Secrets are present in cloud formation outputs in one of the stacks:

**Vulnerability:** CFT stack exposes the secret keys of an S3 User

**Risks:** Attackers might abuse this misconfiguration and gain access not only to S3 services but also to the other AWS services using those exposed credentials, this would result in sensitive data leakage from the health care infrastructure.

**Mitigation:** Enable automated detective controls to scan accounts for passwords and secrets. Use secret manager service to store and retrieve passwords and secrets

#### MFA not configured for IAM users:

**Vulnerability:** IAM roles are assigned based on the duties of an individual/group in the organization. IAM roles play a major role in ensuring confidentiality and integrity of the patient data in a health care organization. Compromise of an admin user account might result in data alteration and exposure.

**Risks:** IAM users without MFA prone for password attacks and a compromise to the user account might compromise the entire infrastructure. This might include data loss, sensitive data exposure, unplanned downtime, breach to data integrity.

**Mitigation:** Enable MFA for IAM users (Preferably, hardware MFA).

#### Overly Permissive IAM roles:

**Vulnerability:** IAM groups/users are configured with AWS managed IAM policy “Administrative Access” that permits access to all the services in the HealthCare Infrastructure’s account.

**Risks:** IAM roles are designed to segregate the access to an individual/group based on their duties in an organization, Configuring Overly permissive policies to the groups might result in privilege escalation and compromise. Failure to follow principle of least privilege might result in configuration changes, downtime, privilege escalation and insider threats.

**Mitigation:** Always follow principle of least privilege while configuring IAM policies, use custom policies, start by granting least access permissions and scale out the permissions based on the need.

#### Misconfiguration in IAM policies:

**Vulnerability:** It is observed that custom IAM roles are overly permissive due to misconfiguration in the policy. It is identified that developer roles are configured with access to all the services in the infrastructure, and guest role is allowed to access S3 Buckets that has public access.

**Risks:** Overly permissive policies in the health care infrastructure leads to privilege escalation, if a guest account gets compromised in the infrastructure, that provides an access to the public S3 buckets would leak sensitive data. Similarly, if a developer account gets compromised, it provides access to EC2 instances, network configurations which might lead to misconfigurations and downtime.

**Mitigation:** Use IAM access analyzer and IAM policy auditor to validate the IAM policies

#### Default Credentials:

**Vulnerability:** Default credentials are always the weakest link for a security compromise. It is observed that the RDS instances are configured with default credentials with weak usernames and passwords.

**Risks:** Using Default credentials in the health care infrastructure leads to data exfiltration, compromise of the service on which the default credentials are configured. Lateral gain of access on other services if the same credentials are used on other services.

**Mitigation:** Change Default Credentials, disable default accounts, update passwords regularly, use 2FA for the credentials.

IMDSV2 Disabled:

**Vulnerability:** IMDSv2 is disabled on the instance/IMDSv1 enabled on the instance

**Risks:** IMDSv1 work on request/response mechanism that does not need any security tokens for authentication, Attackers can perform server-side request forgery to manipulate the requests and retrieve the metadata of the instance for further exploitation.

**Mitigation:** Always Disable IMDSv1 and Enable IMDSv2

Security Groups are open to internet 0.0.0.0/0 on port 22

**Vulnerability:** security groups allow access to all the IP addresses in the internet on port 22.

**Risks:** Attackers might look for vulnerabilities in the remote SSH service and may try to bypass the password authentication and gain access to the instance

**Mitigation:** Restrict the ingress traffic on port 22 to specific prefix lists.

Encryption not implemented

**Vulnerability:** Data on RDS instances and EBS volumes are not encrypted

**Risks:** If not enabled, sensitive information at rest is not protected.

**Mitigation:** Enable encryption on the data wherever needed

High Vulnerabilities in the current infrastructure:

S.No	Vulnerability	Risks
1	Web server uses port 80 for communication	Data unencrypted in transit, Data manipulation and exfiltration, vulnerable to server-side request forgery attacks
2	Cloud Trail logging Disabled	API Actions / Calls are not recorded, critical for incident response
3	Improper Configuration of Backup Policies	Data Loss, Extended Downtime, and non-compliance

Web server uses port 80 for communication

**Vulnerability:** Communication over Plain text might lead to interception of data. The web server is configured to listen on port 80 In the current infrastructure.

**Risks:** Personal Identifiable Information of the patient might get exposed when the request/response is intercepted in the network, Credentials used to access the healthcare portal will get leaked, Possible DOS attacks, Exfiltration of sensitive data.

**Mitigations:** Always encrypt the data at transit, Enable HTTPS communication and disable port 80 on the webserver, the communication over the network must be always encrypted.

#### Cloud Trail logging Disabled

**Vulnerability:** Logging is the critical service in any sensitive and critical infrastructure. It is observed that logging is not enabled

**Risks:** AWS infrastructure is flexible in such a way that every service can be accessed through API calls. It is mandatory to enable cloud trail logging to monitor any API call events on the services to look out for any malicious calls to the services. Failure to configure logging would make the incident response complex

**Mitigations:** Enable Cloud Trail on all the regions and integrate them with cloud watch for alerts regarding the suspicious events.

#### Improper Configuration of Backup Policies

**Vulnerability:** Lack of versioning on S3 buckets will leads to zero recovery of data after a overwrite

**Risks:** This might expose the data leads to data exfiltration

**Mitigations:** Follow Best practices while configuring S3 buckets

#### Medium Vulnerabilities in the current infrastructure:

S.NO	Vulnerability	Risks
1	KMS service not implemented	Data leaks
2	Cloud Watch not enabled	Incident response get affected
3	IAM password policy not implemented	Compromise of the system, vulnerable to password attacks

#### KMS service not implemented

**Vulnerability:** All the keys used for encryption must be rotated periodically and must be stored in secure vaults, Storing the keys on the resources makes it vulnerable, Absence of KMS forces the users to store the keys in insecure vaults or on the instances which might result in compromise.

**Risks:** The encryption keys stored on the resources are prone to exposure when the resource gets compromised. This would lead to data leaks. The patient data are usually encrypted in databases. When an encryption is leaked from the instance, then the entire data is exposed to internet



**Remediations:** Follow a robust plan to rotate the keys and store the keys in KMS which is a key management service in AWS that provides tamper protection.

#### Cloud Watch not enabled

**Vulnerability:** Alerts and notifications about an event/ incident in the healthcare infrastructure is essentials to maintain 100% uptime of the environment and react to the events proactively. Failure to configure cloud watch leads to failure in event monitoring and breach to availability may occur

**Risks:** Unexpected Downtimes, poor incident response, downtime, increase in security incidents, No containment measures. Misconfigurations.

**Remediations:** Enable CloudWatch and configure to raise alerts and notifications on the event of configuration change from unauthorized user accounts, Security events/incidents.

#### IAM password policy not implemented

**Vulnerability:** IAM password policy sets the standards required to protect the passwords from brute force attacks. It is set of rules that helps the users in choosing strong passwords. Furthermore, it provides extended protection to the user accounts by allowing administrators to configure password expiration, reuse protection, authorization to change their passwords. In the current Infrastructure, IAM password policy is not implemented

**Risks:** Misconfiguration/Failure to configure IAM password policy allows users to create weak passwords, no control on handling passwords which makes vulnerable to password-based attacks and result in further exploitation in health care infrastructure

**Remediations:** Enable IAM password policy and perform periodic audits on this policy to eliminate misconfigurations.

## 5.Conclusion

Vulnerabilities due to policy misconfigurations, failure to follow security best practices are two weakest links identified in the current healthcare infrastructure. It is mandatory to perform vulnerability assessments periodically followed by policy auditing to identify the weakest links and remediate them as per the severity and business importance. The report has detailed information about the identified vulnerabilities and its remediations. It is imperative to follow those best remediations and eliminate the vulnerabilities to secure the healthcare data in the infrastructure as well as to adhere to HIPAA guidelines.

## 6. References:

1. [https://docs.aws.amazon.com/IAM/latest/UserGuide/access\\_policies.html](https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies.html)
2. <https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/WhatIsCloudWatchLogs.html>
3. <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSEncryption.html>
4. <https://docs.aws.amazon.com/AmazonS3/latest/userguide/Versioning.html>
5. [https://aws.amazon.com/kms/features/?pg=ln&sec=c/#AWS\\_Service\\_Integration](https://aws.amazon.com/kms/features/?pg=ln&sec=c/#AWS_Service_Integration)
6. <https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Overview.Encryption.html>

# DATA SECURITY ASSESSMENT REPORT

## Table of Contents

1. Executive Summary
2. Scan results
3. Findings
4. Assessment Report
5. Vulnerabilities, Remediation and Recommended Configuration
6. Conclusion
7. References

## 1. Executive Summary

The project involves assessing a healthcare organization's cloud infrastructure security, which is crucial for managing and preserving sensitive patient data. This includes personal data, medical records, application, database server configuration. The main goals are to detect and mitigate any security risks as well as to maintain the confidentiality, integrity, and availability of the applications. A significant amount of test cases related to sensitivity of data; misconfigurations have been tested as part of this data security assessment.

The healthcare company stores patient data, hosts critical healthcare apps, and conducts telemedicine consultations using the cloud extensively. Unfortunately, the system is highly susceptible to potential cyberattacks due to setup faults and security breaches. Finding these vulnerabilities and using them to illustrate the risks associated with the insecure cloud configuration, as well as provide recommendations for repairing it, is the aim of this assessment.

## 2. Scan Results

We have included the Data Security assessment report that is consolidated from the scans of Macie and an open-source tool "prowler."

**Scan Results** - The scan results provide granular detail of each vulnerability, which are categorized by their severity: critical, high, medium, and low. An expanded definition of the known threat and solutions for remediating the vulnerability are also available.

## 3. Findings

Severity	Count
High	1
Medium	10
Low	2

## 4. Data Security Assessment Report

A detailed report is provided about each vulnerability in the current infrastructure and its exploitability in the wild.

[High Severity:](#)

S No.	Service Name	Vulnerability	Risks Involved	Remediation
1	S3	Ensure that Block Public Access is enabled on s3 accounts	S3 buckets and objects may inadvertently become publicly accessible, exposing sensitive data to the internet, thereby letting unauthorized users, including malicious actors, to potentially access, modify, or delete data, leading to data breaches.	Enable Block Public Access on s3 accounts.

--	--	--	--	--

Medium Severity:

s. No.	Service Name	Vulnerability	Risks Involved	Remediation
1	rds	Ensure whether the RDS instances have deletion protection enabled.	Without deletion protection, RDS instances can be accidentally deleted by authorized users, potentially leading to data loss and service disruption.	Enable deletion protection using the AWS Management Console for production DB instances.
2	rds	Check if RDS instances have multi-AZ enabled.	In case of failure, with a single-AZ deployment configuration, should an availability zone specific database failure occur, Amazon RDS cannot automatically fail over to the standby availability zone.	It enhances database resilience and reduces the risk of downtime due to infrastructure or instance failures.
3	rds	Check if RDS instances is integrated with CloudWatch Logs.	If logs are not enabled, monitoring of service use and threat analysis is not possible. Proactive monitoring is essential for identifying potential issues before they impact performance or security. Integrated logs would enable us to take a proactive approach to database management.	Use CloudWatch Logs.
4	rds	Check if RDS instances storage is encrypted.	If not enabled, sensitive information at rest is not protected.	Enable Encryption.
5	S3	Check if S3 bucket MFA Delete is not enabled.	Your security credentials are compromised or unauthorized access is granted.	Add MFA delete to S3 buckets.
6	S3	Check if S3 buckets have KMS encryption enabled.	Amazon S3 KMS encryption provides a way to set the encryption behaviour for an S3 bucket using a managed key. This will	Ensure that S3 buckets have encryption at rest enabled using KMS.

			ensure data-at-rest is encrypted.	
7	S3	Check if S3 buckets have object versioning enabled	With versioning, you can easily recover from both unintended user actions and application failures.	Enable object versioning for S3 buckets.
8	S3	Check if S3 buckets have secure transport policy.	If HTTPS is not enforced on the bucket policy, communication between clients and S3 buckets can use unencrypted HTTP. As a result, sensitive information could be transmitted in clear text over the network or internet.	Ensure that S3 buckets have encryption in transit enabled.
9	S3	Check if S3 buckets have server access logging enabled	Ensure that S3 buckets have Logging enabled. CloudTrail data events can be used in place of S3 bucket logging. If that is the case, this finding can be considered a false positive.	Ensure that S3 buckets have Logging enabled.
10	kms	Ensure rotation for customer created KMS CMKs is enabled.	Failure to rotate keys may leave data encrypted with outdated or vulnerable algorithms, potentially exposing it to security risks. Relying on a single key for an extended period creates a single point of failure. Key rotation ensures that if one key is compromised or lost, the exposure is limited to data encrypted with that key.	Always KMS Customer Master Keys (CMKs) should be used to encrypt the data.

#### Proof of Concepts for the Vulnerabilities:

Vulnerability: Ensure whether the RDS instances have deletion protection enabled.

The below image shows that the RDS instances do not have deletion protection enabled for the given infrastructure.

One of the most critical risks involves the unintended removal of the RDS instance, which can result in the permanent loss of all associated data and configurations if an individual possessing

the appropriate IAM permissions mistakenly deletes it. Deletion protection serves as a protective measure against this potential scenario.

```
(kali㉿kali)-[~]  
$ aws rds describe-db-instances | grep -i "Deletion"  
    "DeletionProtection": false,
```

Vulnerability: Check if RDS instances have multi-AZ enabled

The below image shows that in the given infrastructure, RDS instances do not have multi-AZ enabled.

Without multi-AZ, if the primary instance fails or becomes inaccessible, there is a higher risk of data loss if backups are not configured appropriately. In contrast, multi-AZ automatically maintains a synchronous replica in a different Availability Zone, ensuring data durability and reducing the risk of data loss.

```
(kali㉿kali)-[~]  
$ aws rds describe-db-instances | grep -i "MultiAZ"  
    "MultiAZ": false,
```

Vulnerability: Check if RDS instances is integrated with CloudWatch Logs.

The below image shows that the RDS instances are not integrated with CloudWatch Logs.

The "EnableCloudwatchLogsExports" field usually lists the log types that are being exported to CloudWatch Logs, which indicates that the RDS instance is integrated with CloudWatch Logs. Here, the below image shows that "EnableCloudwatchLogsExports" field is empty or not present in the output, which means that CloudWatch Logs integration is not enabled for the RDS instance.

```
(kali㉿kali)-[~]  
$ aws rds describe-db-instances | grep -i "EnableCloudwatchLogsExports"  
  
(kali㉿kali)-[~]  
$
```

Vulnerability: Check if S3 bucket MFA Delete is not enabled.

The below image shows that the RDS instances storage is not encrypted for the infrastructure.

Storing unencrypted data in your RDS instance exposes it to unauthorized access. If an attacker gains entry to the underlying storage, they can read, copy, or manipulate your sensitive data, resulting in data breaches and breaches of confidentiality. Various regulatory standards and industry requirements (such as GDPR, HIPAA, PCI DSS) enforce the encryption of sensitive data. Neglecting to encrypt data can lead to non-compliance, which carries legal and financial consequences.

```
(kali㉿kali)-[~]  
$ aws rds describe-db-instances | grep -i "StorageEncrypted"  
"StorageEncrypted": false,
```

Vulnerability: MFA Delete is not enabled for s3 buckets.

The Health Insurance Portability and Accountability Act (HIPAA) imposes stringent requirements for securing and maintaining the privacy of patient health information (PHI). Inadequate protection of PHI can lead to violations of HIPAA, which entail significant penalties. The absence of MFA Delete heightens the risk of unauthorized deletions of PHI and other sensitive healthcare data, thereby increasing the potential for non-compliance with HIPAA regulations.

```
(kali㉿kali)-[~]  
$ aws s3api get-bucket-versioning --bucket cf-templates-140eqzibzav8d-us-east-1  
  
(kali㉿kali)-[~]  
$ aws s3api get-bucket-versioning --bucket medcirclepatientdata1-421235857870  
  
(kali㉿kali)-[~]  
$
```

Vulnerability: S3 buckets do not have object versioning enabled.

Without object versioning, there is no automatic version history for objects in the S3 bucket. If a critical file or piece of patient information is accidentally overwritten or deleted, it may be impossible to recover it, leading to data loss.

Healthcare organizations are subject to strict data protection regulations, such as HIPAA. Object versioning can be an important tool for demonstrating data integrity and compliance. Its absence may lead to compliance challenges and potential regulatory violations.

```
(kali㉿kali)-[~]  
$ aws s3api get-bucket-versioning --bucket medcirclepatientdata1-421235857870  
  
(kali㉿kali)-[~]  
$
```

Vulnerability: Check if S3 buckets have secure transport policy.



Without a secure transport policy, data transferred to and from S3 buckets may not be adequately protected during transit. This introduces the risk of eavesdropping and interception by unauthorized individuals, potentially leading to the exposure of sensitive patient health information (PHI) and other healthcare data.

```
(kali㉿kali)-[~]
$ aws s3api get-bucket-policy --bucket medcirclepatientdata1-421235857870
An error occurred (NoSuchBucketPolicy) when calling the GetBucketPolicy operation: The bucket policy does not exist

(kali㉿kali)-[~]
$ aws s3api get-bucket-policy --bucket cf-templates-140eqzibzav8d-us-east-1
An error occurred (NoSuchBucketPolicy) when calling the GetBucketPolicy operation: The bucket policy does not exist
```

Vulnerability: S3 buckets do not have server access logging enabled.

Server access logs have a vital role in the detection of security incidents and unauthorized access to S3 buckets. The absence of these logs can pose challenges in promptly identifying and responding to potential breaches. It becomes more difficult to track and investigate unauthorized activities, hindering timely incident response and mitigation efforts.

```
(kali㉿kali)-[~]
$ aws s3api get-bucket-logging --bucket medcirclepatientdata1-421235857870

(kali㉿kali)-[~]
$ aws s3api get-bucket-logging --bucket cf-templates-140eqzibzav8d-us-east-1
```

Low Severity:

SI No.	Service Name	Vulnerability	Risk Involved	Remediation
1	rds	Ensure that RDS instances has enhanced monitoring enabled.	Without enhanced monitoring, you have reduced visibility into the performance metrics of your RDS instances, making it challenging to identify and address performance bottlenecks and inefficiencies. Enhanced monitoring provides more granular metrics and detailed insights, allowing for quicker detection and resolution of performance issues. Without it, you may experience delayed issue identification, leading to service disruptions.	Enable enhanced Monitoring.

2	S3	Ensure that S3 buckets have object lock enabled	Without Object Lock enabled on S3 buckets, there are risks of accidental or intentional data deletion, modification, and unauthorized access. Compliance and legal retention requirements may not be met, and data integrity and security could be compromised, leading to potential data loss and compliance issues.	Ensuring that Amazon S3 buckets have Object Lock feature enabled helps prevent the objects they store from being deleted.
---	----	---	---	---

#### Proof of Concepts for the Vulnerabilities:

Vulnerability: Ensure that RDS instances has enhanced monitoring enabled

Having enhanced monitoring in place is crucial for early detection of security threats, such as unauthorized access, SQL injection, or other database attacks. Without this monitoring capability, organizations may lack the essential visibility required to promptly identify and respond to such threats. This increases the vulnerability to data breaches and compromises the overall security posture of the organization.

Insufficient implementation of enhanced monitoring for RDS instances can give rise to potential HIPAA compliance concerns, hindered detection of security threats, data breaches, operational disruptions, and a lack of data for optimizing database performance. Enhanced monitoring serves as a valuable tool for protecting patient data, ensuring compliance, and upholding the integrity and security of healthcare systems.

```
(kali@kali)~$ aws rds describe-db-instances --query "DBInstances[+].{DBInstanceIdentifier:DBInstanceIdentifier, EnhancedMonitoringStatus:EnhancedMonitoring}" --output table
```

DescribeDBInstances	
DBInstanceIdentifier	EnhancedMonitoringStatus
midterm-group17-smandal1-dbinstance-zgcrpmd23rb	None

Vulnerability: S3 buckets do not have object lock enabled.

Object Lock contributes to maintaining data integrity and creating detailed audit trails. This is crucial for the healthcare industry to track who accessed patient data, when, and for what purpose. The absence of Object Lock can lead to a lack of accountability and visibility into data access, which is essential for compliance and data security.

## 5. Vulnerabilities, Remediation and Recommended Configuration

- **Vulnerability:** Block Public Access is not enabled on s3 accounts

**Remediation:** Enabling Block Public Access:

1) Helps prevent accidental exposure of sensitive data to the public or unauthorized users, thereby assisting in meeting compliance requirements by reducing the risk of data breaches.

2) It also helps mitigate the risk of data loss by preventing unauthorized changes or deletions of objects. This is especially valuable for preserving critical data, such as audit logs or historical records.

**Recommended Configuration:**

Below is the recommended configuration for enabling a block on Public Access on s3 accounts:

```
(kali@kali)-[~]
$ aws s3api put-public-access-block --bucket medcirclepatientdata1-421235857870 --public-access-block-configuration "BlockPublicAcls=true,IgnorePublicAcls=true,BlockPublicPolicy=true,RestrictPublicBuckets=true"

(kali@kali)-[~]
$ aws s3api get-public-access-block --bucket medcirclepatientdata1-421235857870
{
  "PublicAccessBlockConfiguration": {
    "BlockPublicAcls": true,
    "IgnorePublicAcls": true,
    "BlockPublicPolicy": true,
    "RestrictPublicBuckets": true
  }
}
```

- **Vulnerability:** RDS instances do not have deletion protection enabled.

**Remediation:** Enable deletion protection using the AWS Management Console for production DB instances. Only those instances can be deleted that do not have deletion protection enabled.

- **Vulnerability:** RDS instances don't have multi-AZ enabled.

**Remediation:** It enhances database resilience and reduces the risk of downtime due to infrastructure or instance failures.

- **Vulnerability:** RDS instances integrated with CloudWatch Logs.

**Remediation:** Use CloudWatch Logs to perform real-time analysis of the log data. Create alarms and view metrics.

- **Vulnerability:** RDS instances storage is not encrypted.

**Remediation:** Enable Encryption. Use a CMK where possible. It will provide additional management and privacy benefits.

- **Vulnerability:** MFA Delete is not enabled on s3 buckets.

**Remediation:** Adding MFA delete to an S3 bucket, requires additional authentication when you change the version state of your bucket, or you delete and object version adding another layer of security in the event your security credentials are compromised or unauthorized access is granted.

- **Vulnerability:** KMS encryption is not enabled on s3 buckets.

**Remediation:** Ensure that S3 buckets have encryption at rest enabled using KMS. S3 buckets should be reviewed regularly to ensure that KMS encryption is enabled on any newly created buckets and that encryption remains enabled on existing buckets.

- **Vulnerability:** S3 buckets do not have object versioning enabled.

**Remediation:** Configure versioning using the Amazon console or API for buckets with sensitive information that is changing frequently; and backup may not be enough to capture all the changes.

- **Vulnerability:** S3 buckets do not have secure transport policy.

**Remediation:** Ensure that S3 buckets have encryption in transit enabled.

- **Vulnerability:** S3 buckets do not have server access logging enabled.

**Remediation:** Ensure that S3 buckets have Logging enabled. CloudTrail data events can be used in place of S3 bucket logging. If that is the case, this finding can be considered a false positive.

- **Vulnerability:** Rotation for customer created KMS (Key Management Systems) CMKs (Customer Managed keys) is enabled.

**Remediation:** Always KMS Customer Master Keys (CMKs) should be used to encrypt the data. Usage of symmetric keys should be avoided for encryption. AWS KMS can automatically rotate keys at regular intervals. This enhances security by reducing the risk associated with long-lived keys.

- **Vulnerability:** RDS instances do not have enhanced monitoring enabled.

**Remediation:** To use Enhanced Monitoring, you must create an IAM role; and then enable Enhanced Monitoring.

- **Vulnerability:** S3 buckets do not have object lock enabled.

**Remediation:** Storing objects using a write-once-read-many (WORM) models may help prevent objects from being deleted or overwritten for a fixed amount of time or indefinitely. That helps to prevent ransomware attacks. Ensuring that Amazon S3 buckets have Object Lock feature enabled helps prevent the objects they store from being deleted.

- **Vulnerability:** Key Management Services (KMS) is not enabled for EBS volumes.

**Remediation:** Enable encryption for your EBS volumes to protect data at rest. Use AWS Key Management Service (KMS) to manage encryption keys securely. Healthcare

organizations are subject to strict data protection regulations like HIPAA (Health Insurance Portability and Accountability Act) in the United States. Failure to encrypt patient data may result in non-compliance with these regulations, leading to fines and penalties.

- [Vulnerability](#): Unauthorized access to EBS volume snapshots

[Remediation](#): AWS Identity and Access Management (IAM) policies to control access to your EBS snapshots. Create and assign IAM policies that grant permissions only to authorized users and services. Review and update these policies regularly. The principle of least privilege should be followed, granting users and services only the permissions they need to perform their tasks.

## 6. Conclusion

Data plays an important role in any infrastructure, especially in a Health Care Infrastructure, it is imperative to protect the patient's PII. Hence, encryption is essential wherever PII is present in the infrastructure. Besides that, the current health care infrastructure does not adhere to HIPAA standards, Policy best practices and infrastructure best practices. Vulnerabilities and misconfigurations that are identified in the infrastructure through manual and automated scans are listed and remediations are provided highlighting the risk. It is mandatory to follow those best practices to secure the data from exfiltration.

## 7. References

1. <https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Welcome.html>
2. <https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Overview.Encryption.html>
3. <https://aws.amazon.com/kms/>
4. [https://aws.amazon.com/s3/security/?pg=ln&sec=be#Access\\_management\\_and\\_security](https://aws.amazon.com/s3/security/?pg=ln&sec=be#Access_management_and_security)
5. <https://aws.amazon.com/blogs/database/best-practices-for-securing-sensitive-data-in-aws-data-stores/>

# VIRTUAL MACHINE VULNERABILITY ASSESSMENT REPORT

## Table of Contents

1. Executive Summary
2. Scan results
3. Findings
4. Assessment Report
5. Remediations, Security Policy and Recommended Configuration
6. Conclusion
7. References

## 1. Executive Summary

The project entails evaluating the cloud infrastructure security of a healthcare organization, which is essential for handling and storing sensitive patient data. This contains billing information, personal information, and medical records. The primary goal of the assessment is to identify the weaknesses and misconfigurations in the system and ensure confidentiality, integrity, and availability of the infrastructure. The evaluation will address several cloud security topics, including controlling user access, encrypting information, protecting the network, and handling vulnerabilities.

The healthcare organization uses the cloud extensively to host vital healthcare apps, store patient data, and do telemedicine consultations. Regrettably, because of setup errors and security lapses, the infrastructure is extremely vulnerable to possible cyberattacks. The objective is to identify and take advantage of these vulnerabilities to highlight the dangers connected to the unsecure cloud configuration and offer suggestions for fixing it. This project is essential for protecting private patient data and guaranteeing the quality of the medical care given.

## 2. Scan Results

The Virtual machine Vulnerability assessment report includes the report generated from the scans of AWS config, inspector, and an open-source tool “prowler”

**Scan Results** - The scan results provide granular detail of each vulnerability, which are categorized by their severity: High, medium, and low. An expanded definition of the known threat and solutions for remediating the vulnerability are also available.

## 3. Findings

Severity	Count
High	3
Medium	4
Low	3

## 4. Virtual Machine Vulnerability Assessment:

A detailed report is provided about each vulnerability in the current infrastructure and its exploitability in the wild.

### High Vulnerabilities in the Current Infrastructure:

S.No	Vulnerability	Risks Involved	Remediation
1	Security Groups are open to internet 0.0.0.0/0 on port 22	Misconfiguration in security groups leads to increase in	Restrict the ingress traffic on port 22 to specific prefix lists.

		attack surface for network intrusion	
2	Ensure the default security group of every VPC restricts all traffic.	Default Security groups allow access to 0.0.0.0/0 on all protocols. Overly permissive security groups might allow attackers to do network pivoting and exfiltration.	Follow the best practices to configure the Default security groups. Minimize the number of ports that are allowed and restrict the access to authorized prefix lists

### Detailed Explanation of the vulnerabilities:

Vulnerability: Security Groups are open to internet 0.0.0.0/0 on port 22

```

{
  "FromPort": 22,
  "IpProtocol": "tcp",
  "IpRanges": [
    {
      "CidrIp": "0.0.0.0/0"
    }
  ],
  "Ipv6Ranges": [],
  "PrefixListIds": [],
  "ToPort": 22,
  "UserIdGroupPairs": []
}

```

- From the configuration mentioned in the above screenshot, it is understandable that the security groups allow access to all the IP addresses in the internet on port 22.
- Port 22 when exposed to all the IP addresses in the internet result in service enumeration as the attackers would scan for the vulnerable SSH versions.
- Attackers might look for vulnerabilities in the remote SSH service and may try to bypass the password authentication and gain access to the instance
- From the below screenshot, while scanning for one of the Instance IP addresses in the healthcare infrastructure, the instance has exposed the SSH server version, this might allow attackers to further scan for vulnerabilities in SSH service.



```
(kali㉿ kali)-[~]
$ sudo nmap -sV -sS -p22 3.223.17.230
Starting Nmap 7.93 ( https://nmap.org ) at 2023-10-27 11:11 EDT
Nmap scan report for ec2-3-223-17-230.compute-1.amazonaws.com (3.223.17.230)
Host is up (0.0021s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.4 (protocol 2.0)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.03 seconds
```

Vulnerability: Default security groups allow access on all ports from the resources inside the VPC and provide outbound access to all the ports for the internet

```
"SecurityGroups": [
  {
    "Description": "default VPC security group",
    "GroupName": "default",
    "IpPermissions": [
      {
        "IpProtocol": "-1",
        "IpRanges": [],
        "Ipv6Ranges": [],
        "PrefixListIds": [],
        "UserIdGroupPairs": [
          {
            "GroupId": "sg-0fa64e1b46ca37660",
            "UserId": "545361066637"
          }
        ]
      }
    ],
    "OwnerId": "545361066637",
    "GroupId": "sg-0fa64e1b46ca37660",
    "IpPermissionsEgress": [
      {
        "IpProtocol": "-1",
        "IpRanges": [
          {
            "CidrIp": "0.0.0.0/0"
          }
        ],
        "Ipv6Ranges": [],
        "PrefixListIds": [],
        "UserIdGroupPairs": []
      }
    ]
  }
]
```

- From the above configuration of the default security group in the health care infrastructure, it is understandable that the default security group provides an inbound access on all ports to all the resources inside the respective VPC and provide outbound access to all the resources in the internet.
- Although default security group does not possess direct risk to the infrastructure when not attached to any resource, however, this security group when accidentally attached to any of the resource in the HealthCare Infrastructure, this might lead to compromise of the resource in couple of ways:
  - 1) Pivoting would become easier because of the misconfigured security groups
  - 2) Attackers might try to exfiltrate sensitive data from the sever/try to communicate to C&C for further exploitation.

### Medium Vulnerabilities in the current infrastructure:

S.No	Vulnerability	Risks Involved	Remediation
1	IMDSv2 is disabled/ IMDSv1 is enabled	The sensitive information from the AWS backend services can be exposed when IMDSv1 is enabled	Always Disable IMDSv1 and Enable IMDSv2
2	IAM instance role is not configured to access EC2 instance	Increases the risks associated with sharing and rotating the credentials that can be used	Enable the Instance IAM role for the ec2 instance
3	Network ACL's allow access to the resources from 0.0.0.0/0 on all the ports including sensitive ports like 22 and 3389	This increases the attack surface of the infrastructure that allows the internet users to scan the network for sensitive ports	Restrict the access to the network ACLs on sensitive ports for specific prefix lists/IP addresses

### Proof of concept for the vulnerability:

Vulnerability: IMDSv2 is disabled on the instance/IMDSv1 enabled on the instance

```
(kali@ kali)-[~]
$ aws ec2 describe-instances --instance-id i-06f86ff66b6a220e2 | grep -A 3 "Metadata"
    "MetadataOptions": {
        "State": "applied",
        "HttpTokens": "optional",
        "HttpPutResponseHopLimit": 1,
--
        "InstanceMetadataTags": "disabled"
    },
```

- From the above screenshot, it is observed that the deployed instance is configured to use IMDSv1 or not using IMDSv2. HTTP Tokens Says “Optional” which means the instance supports both IMDSv1 and IMDSv2 However, IMDSv1 is enabled.
- IMDS in AWS allows to retrieve the information about the instances from the backend endpoints. An EC2 instance can use IMDS to provide an API request to the backend service and fetch the information about the instance. IMDSv1 work on request/response mechanism that does not need any security tokens for authentication, Attackers can perform server-side request forgery to manipulate the requests and retrieve the metadata of the instance for further exploitation. IMDSv1 makes the instance vulnerable as no authentication mechanism involved in retrieving the metadata of the instance.
- This vulnerability is circumvented by enabling IMDSv2 on the instance which needs token authentication for retrieving metadata from backend service. This token is short lived and specific to the process which requested it.
- IMDSv1 is identified to be vulnerable because of its nature of retrieving the data about an instance by performing request/response type of API calls. This would allow attackers to perform server-side request forgery by manipulating the requests if an intermediate reverse proxy or load balancer in AWS environment is misconfigured to forward the requests. Attackers might manipulate the URLs to retrieve the information about the instance for further exploitation.

Vulnerability: IAM instance role is not configured to access EC2 instance

```
(kali@kali)~$ aws ec2 describe-instances --instance-id i-06f86ff66b6a220e2 --query 'Reservations[*].Instances[*].IamInstanceProfile'
{
  []
}
```

- IAM instance role is the feature of AWS that grants permissions to the applications on the EC2 instances to communicate with other resources securely. Lack of the IAM instance roles as observed in the above policy listing would result in risks associated with rotating the credentials in AWS environment.
- Applications on the EC2 instances must sign the API requests with AWS credentials to access other resources in AWS through API calls. The credentials must be updated on all the EC2 instances which needs application access, this might result in security risk when credentials are not properly rotated, moreover, any misconfigured instance might expose the credentials and result in further exploitation.
- To overcome this vulnerability, AWS instance role must be created and attached to EC2 instances that needs access to other resources in AWS. This strategy would reduce the risk of configuring the credentials on the instance and prevent the exposure.

Vulnerability: NACL's are exposed to internet on sensitive ports

```
{
  "CidrBlock": "0.0.0.0/0",
  "Egress": true,
  "Protocol": "-1",
  "RuleAction": "allow",
  "RuleNumber": 100
},
{
  "CidrBlock": "0.0.0.0/0",
  "Egress": false,
  "Protocol": "-1",
  "RuleAction": "allow",
  "RuleNumber": 100
},
```

- North-South and East-West Traffic has no restriction on the ports ingress/egress directions. This would allow the Attack surface to scan through the network and enumerate the ports on the resources of AWS infrastructure. NACLs exhibit the functionality of protecting the infrastructure at network perimeter and hence misconfiguration in the NACL rules allow attackers to scan through the network
- Access to the sensitive ports like SSH, RDP must be restricted at the NACL level so that the filtering happens in the perimeter level instead of entering the infrastructure.

#### Low Vulnerabilities in the current infrastructure:

S.No	Vulnerability	Risks involved	Remediation
1	Monitoring not enabled on EC2 instances	Failure to configure monitoring for the EC2 instances result in losing the insights about each instance and make it complex for any troubleshooting purposes	Enable Monitoring on the instances
2	Ensure there are no Security Groups not being used.	Unused security groups although does not pose any direct risks, misconfiguration might result in increasing attack surface	Delete all unused security groups/ properly tune them as per best practices

Vulnerability: Monitoring not enabled on EC2 instances

- EC2 Monitoring helps in capturing the logs on the AWS instance that helps administrator to identify the events on the instances.
- Failing to enable monitoring would result in missing security events on the instance which would increase complexity in incident response.

Vulnerability: Ensure there are no Security Groups not being used.

- Unused Security groups always possess threat if those groups are misconfigured. If an administrator attaches any of those misconfigured security groups to any of the resources, that would result in exploitation.
- To Remediate this vulnerability, always delete the unused security groups, or follow best practices to configure those unused security groups

Vulnerability: Patching not implemented properly

Besides the vulnerabilities mentioned in the infrastructure, All the healthcare infrastructure in the AWS environment must be upgraded to the latest software version to prevent the exploitation due to vulnerable software versions. This can be regarded as the high vulnerability in the infrastructure as the vulnerable versions of software is not resistant against zero-day attacks

Vulnerability: Amazon inspector not enabled

Amazon inspector which is not currently enabled in the healthcare infrastructure. It must be enabled to discover and scan the EC2 instances automatically for software vulnerabilities and unintended network exposure. This can be considered as a low vulnerability because a periodic scan of the infrastructure with an open-source tool would provide the necessary information about the vulnerabilities.

Vulnerability: Security Hub not enabled

It is identified that security hub is not enabled in the current healthcare infrastructure. It is considered as the infrastructure best practice to enable security hub as it scans the EC2 instances against security best practices and security standards. This can be considered as the low vulnerability

Vulnerability: Auto Scaling disabled

The current infrastructure enables auto scaling for most of the instances, However, it is recommended to monitor the resources and implement auto scaling groups for all the instances so that the application is available to the users all the time. This can be considered as the failure to implement best practices in the infrastructure.

## 5. Remediations, Security Policy and Recommended Configuration

Vulnerability: Security Groups are open to internet 0.0.0.0/0 on port 22

Remediation: Restrict the ingress traffic on port 22 to specific prefix lists.

Recommended Configuration: Below configuration restricts the sensitive ports to specific prefix lists

```

1  {
2      "IpPermissions": [
3          {
4              "IpProtocol": "tcp",
5              "FromPort": 22,
6              "ToPort": 22,
7              "UserIdGroupPairs": [],
8              "IpRanges": [
9                  {
10                     "CidrIp": "x.x.x.x/24" // Replace with your specific CIDR block
11                 },
12                 {
13                     "CidrIp": "y.y.y.y/24" // Replace with another specific CIDR block if needed
14                 }
15             ]
16          }
17      ],
18      "Description": "Allow SSH from specific CIDR blocks",
19      "GroupName": "SSH-Only-SG", // Name of your Security Group
20      "VpcId": "your-vpc-id", // Replace with your VPC ID
21      "Tags": []
22  }
23

```

**Vulnerability:** Ensure the default security group of every VPC restricts all traffic

**Remediation:** Follow the best practices to configure the Default security groups. Minimize the number of ports that are allowed and restrict the access to authorized prefix lists

**Recommended Configuration:** Below configuration restricts the sensitive ports to specific prefix lists

```

1  {
2      "IpPermissions": [
3          {
4              "IpProtocol": "tcp",
5              "FromPort": 22,
6              "ToPort": 22,
7              "UserIdGroupPairs": [],
8              "IpRanges": [
9                  {
10                     "CidrIp": "x.x.x.x/24" // Replace with your specific CIDR block for SSH access
11                 }
12             ]
13          },
14          {
15              "IpProtocol": "tcp",
16              "FromPort": 80,
17              "ToPort": 80,
18              "UserIdGroupPairs": [],
19              "IpRanges": [
20                  {
21                     "CidrIp": "y.y.y.y/24" // Replace with your specific CIDR block for HTTP access
22                 }
23             ]
24          }
25      ],
26      "Description": "Default Security Group with restricted ports",
27      "GroupName": "default",
28      "VpcId": "your-vpc-id", // Replace with your actual VPC ID
29      "Tags": []
30  }
31

```

**Vulnerability:** IMDSv2 is disabled/ IMDSv1 is enabled

**Remediation:** Always Disable IMDSv1 and Enable IMDSv2

**Recommended Configuration:** HTTP tokens must be changed from “optional” to “required”

```

1  {
2      "HttpTokens": "required"
3  }
4

```

**Vulnerability:** IAM instance role is not configured to access EC2 instance

**Remediation:** Enable the Instance IAM role for the ec2 instance

**Recommended Configuration:** This enables the Instance IAM role on ec2 instances

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}

```

**Vulnerability:** Network ACL's allow access to the resources from 0.0.0.0/0 on all the ports including sensitive ports like 22 and 3389

**Remediation:** Restrict the access to the network ACLs on sensitive ports for specific prefix lists/IP addresses

**Recommended Configuration:** The sample configurations allow only specific address to access sensitive ports.

```

{
  "NetworkAclId": "nacl-12345678",
  "InboundRules": [
    {
      "RuleNumber": 100,
      "Protocol": "6",
      "RuleAction": "allow",
      "CidrBlock": "x.x.x.x/24",
      "PortRange": {
        "From": 22,
        "To": 22
      }
    },
    {
      "RuleNumber": 110,
      "Protocol": "6",
      "RuleAction": "deny",
      "CidrBlock": "0.0.0.0/0",
      "PortRange": {
        "From": 22,
        "To": 22
      }
    }
  ]
}

```

**Vulnerability:** Monitoring not enabled on EC2 instances

**Remediation:** Enable Monitoring on the instances

**Recommended Configuration:** The below sample configuration enables monitoring on the instances

```
{
  "InstanceStatuses": [
    {
      "AvailabilityZone": "us-east-1a",
      "InstanceId": "i-1234567890abcdef0",
      "InstanceStatus": {
        "Details": [],
        "Status": "ok"
      },
      "InstanceState": {
        "Code": 16,
        "Name": "running"
      },
      "InstanceStateReason": "",
      "Monitoring": {
        "State": "enabled"
      },
      "SystemStatus": {
        "Details": [],
        "Status": "ok"
      }
    }
  ]
}
```

## 5. Conclusion

Virtual Machines are heart of the health care infrastructure as the critical health care applications are hosted on the ec2 instances. Personal Identifiable Information (PII) is processed and stored on those instances. These components have got significant importance when it comes to securing them. Several vulnerabilities identified in this critical infrastructure related to VM security are listed, remediations and recommendations have been provided above. It is important to follow those recommendations to protect the instance from exploits and secure the data.



## 6. References

- 1) <https://docs.aws.amazon.com/autoscaling/ec2/userguide/what-is-amazon-ec2-auto-scaling.html>
- 2) <https://docs.aws.amazon.com/vpc/latest/userguide/vpc-network-acls.html>
- 3) <https://docs.aws.amazon.com/vpc/latest/userguide/vpc-security-best-practices.html>

# NETWORK SECURITY ASSESSMENT REPORT

## Table of Contents

1. Executive Summary
2. Scan results
3. Findings
4. Current State analysis, Risks and Remediations
5. Conclusion
6. References

## 1. Executive Summary

The project assesses the security of a healthcare organization's cloud infrastructure, which is essential when processing and storing sensitive patient data. The goal of this network assessment is to identify the misconfiguration in the network infrastructure of the health care environment and highlight the possible risks that would affect the confidentiality, integrity, and availability in the current infrastructure.

Cloud services are widely used by healthcare organization to host critical healthcare applications, store patient data, and consult with telemedicine. Network Security assessment has identified that the infrastructure is vulnerable to data exposure due to misconfigurations in the infrastructure. The goal is to identify and exploit these vulnerabilities to highlight the threats associated with an insecure cloud configuration and provide recommendations for remediation. This project is necessary to protect the patient's data and adhere to Data standards.

## 2. Scan Results

We have included the Network Security assessment report that is consolidated from the scans of AWS config, inspector, Security Hub, and an open-source tool "prowler"

**Scan Results-** The scan results provide granular detail of each vulnerability, which are categorized by their severity: critical, high, medium, and low. An expanded definition of the known threat and solutions for remediating the vulnerability are also available.

## 3. Findings

Severity	Count
Critical	6
High	2
Medium	3

## 4.Current State analysis, Risks and Remediations

### Critical Vulnerabilities in the infrastructure

**Vulnerability:** Public and Private VPC Subnets are not separated in the infrastructure.

**Description:** In AWS best practices, a VPC should have both public and private subnets. This architecture ensures that services accessible from the internet and those used internally are adequately isolated.

**Risks:** Lack of subnet separation can result in unintended exposures, potentially compromising the confidentiality, integrity, and availability of services and data.

```
(kali@kali)-[~]
$ aws ec2 describe-subnets --filters "Name=vpc-id,Values=vpc-04ca092b35ee2f706" --region us-east-1 --query "Subnets[*].{ID:SubnetId, CIDR:CidrBlock, PublicIP:MapPublicIpOnLaunch}"
[
  {
    "ID": "subnet-0d0807a1ecf23b62a",
    "CIDR": "172.31.0.0/20",
    "PublicIP": true
  },
  {
    "ID": "subnet-0ca5b62ea14389bd7",
    "CIDR": "172.31.0.0/20",
    "PublicIP": true
  },
  {
    "ID": "subnet-0e1ffcfa318b074a9",
    "CIDR": "172.31.32.0/20",
    "PublicIP": true
  },
  {
    "ID": "subnet-07d721482b3a2b769",
    "CIDR": "172.31.16.0/20",
    "PublicIP": true
  },
  {
    "ID": "subnet-0cc20a6d1e4a9701e",
    "CIDR": "172.31.80.0/20",
    "PublicIP": true
  },
  {
    "ID": "subnet-0498aaac88e3969f4",
    "CIDR": "172.31.64.0/20",
    "PublicIP": true
  }
]
```

All the subnets have "Public IP": true, which indicates that they are set to automatically assign public IP addresses to instances launched in these subnets. This is a characteristic of public subnets.

Looking at the route with a DestinationCidrBlock of 0.0.0.0/0 and a GatewayId that starts with igw-, then that indicates the subnet has a direct route to the internet via an Internet Gateway, making it a public subnet.

```

(kali@kali)-[~]
$ aws ec2 describe-route-tables --filters "Name=vpc-id,Values=vpc-04ca092b35ee2f706" --region us-east-1
{
  "RouteTables": [
    {
      "RouteTableId": "rtb-0932d057dd7dbd047",
      "Associations": [
        {
          "Main": true,
          "RouteTableAssociationId": "rtbassoc-05f89ee886f429b70",
          "RouteTableId": "rtb-0932d057dd7dbd047",
          "AssociationState": {
            "State": "associated"
          }
        }
      ],
      "PropagatingVgws": [],
      "Routes": [
        {
          "DestinationCidrBlock": "0.0.0.0/0",
          "GatewayId": "igw-0e79bd17c9f5135ef",
          "Origin": "CreateRoute",
          "State": "active"
        }
      ],
      "Tags": [],
      "VpcId": "vpc-04ca092b35ee2f706",
      "OwnerId": "442346618503"
    }
  ]
}

```

## Remediations:

- Ensure VPC is divided into private and public subnets.
- Equip public subnets with appropriate security measures such as security groups and NACLs.
- For private subnets, employ a NAT gateway for safe external resource access.

## Vulnerability: VPC Distribution Across Regions

**Description:** VPCs distributed across multiple regions result in better availability and resilience against region-specific issues.

**Risks:** Limiting VPCs to a singular region can introduce vulnerabilities to regional outages, leading to potential service disruptions.

## Proof of Concept

```
| DescribeVpcs |
+-----+-----+
| CIDR | VPC_ID |
+-----+-----+
| 172.31.0.0/16 | vpc-0dda94d8db272f578 |
+-----+-----+

Region: ap-southeast-2

| DescribeVpcs |
+-----+-----+
| CIDR | VPC_ID |
+-----+-----+
| 172.31.0.0/16 | vpc-01a7f7e3893c05c1a |
+-----+-----+

Region: eu-central-1

| DescribeVpcs |
+-----+-----+
| CIDR | VPC_ID |
+-----+-----+
| 172.31.0.0/16 | vpc-00899f8e88963687f |
+-----+-----+

Region: us-east-1

| DescribeVpcs |
+-----+-----+
| CIDR | VPC_ID |
+-----+-----+
| 172.31.0.0/16 | vpc-04ca092b35ee2f706 |
| 10.1.0.0/16 | vpc-0d8b83d7d7d49b9df |
+-----+-----+

Region: us-east-2

| DescribeVpcs |
+-----+-----+
| CIDR | VPC_ID |
+-----+-----+
```

we can see that VPC's in region us-east-1 are not available in other regions.

## Remediations:

- Distribute VPCs across multiple AWS regions to boost redundancy.
- Minimize dependency on a single region to mitigate risks.

Vulnerability: VPC flow logs are not enabled

**Description:** VPC Flow Logs are instrumental in offering insights into network traffic within the VPC. These logs are important for both security monitoring and troubleshooting connectivity concerns.

**Risks:** Without VPC Flow Logs, anomaly detection in traffic or security incident analysis becomes significantly more challenging.

**Proof of Concept:**

POC for vpc-04ca092b35ee2f706:

```
(kali@kali)-[~]$ for security, out of 105 available
$ aws ec2 describe-flow-logs --filter "Name=resource-id,Values=vpc-04ca092b35ee2f706"
{ec2-user@ip-10-1-10-104 ~}$ exit
logout
Conn "FlowLogs": [ ]4.94.98 closed.
}
```

**Remediations:**

- Activate VPC Flow Logs for all VPCs.
- Ensure to capture packet rejects for a more holistic understanding of traffic.

**Vulnerability:** Default Public IP Assignment in VPC Subnets

**Description:** Within a VPC, each subnet can have its unique traffic rules. Auto-assigning public IPs can unintentionally make instances within these subnets' internet-accessible.

**Risks:** This configuration escalates risks by potentially exposing internal instances, thus compromising security.

```
(kali@kali)-[~]$ aws ec2 describe-subnets --subnet-ids subnet-0d0807a1ecf23b62a subnet-0ca5b62ea14389bd7 subnet-0e1ffcfa318b074a9 subnet-07d721482b3a2b769 subnet-0498aaac88e3969f4 --query 'Subnets[][SubnetId,MapPublicIpOnLaunch]'
[
  [
    "subnet-0d0807a1ecf23b62a",
    true
  ],
  [
    "subnet-0ca5b62ea14389bd7",
    true
  ],
  [
    "subnet-0e1ffcfa318b074a9",
    true
  ],
  [
    "subnet-07d721482b3a2b769",
    true
  ],
  [
    "subnet-0498aaac88e3969f4",
    true
  ]
]
```

Here MapPublicIpOnLaunch is true, it means the subnet is set to auto-assign public IPs by default.

**Remediations:**

- Modify VPC subnets to refrain from default public IP assignments.

Vulnerability: Encryption at transit is not implemented properly

Risks: Web Server Listens on Port 80 which results in unencrypted communication over the internet. This could possibly result in Man in the middle and DDOS attacks

Remediations: Enable the Communication on port 443 rather than using port 80. This would encrypt the data at transit.

For instances requiring public access, assign IPs manually ensuring that robust security protocols are enforced.

All subnets having security group configurations with two inbound rules:

1. HTTP traffic (Port 80) is allowed from any source (0.0.0.0/0).
2. SSH traffic (Port 22) is also allowed from any source (0.0.0.0/0).

Allowing HTTP traffic from any source is common for web servers that are meant to be accessible to the public. However, allowing SSH traffic from any source can be risky. It means that anyone from any IP address can attempt to SSH into instances associated with this security group, which might expose the instances to potential unauthorized access or brute-force attacks.

#### High Vulnerabilities in the Infrastructure:

Vulnerability: Security groups are open to internet on sensitive ports

Risks: Security groups on the instances restricts the traffic ingress/egress of the instances, configuring security groups to allow traffic on sensitive ports leads to increase in attack surface in the infrastructure

Remediations:

Restrict SSH access only to specific IP addresses or ranges, typically your own IP or a set of known IPs that require SSH access to the instances.

Vulnerability: Public IP of the instance is used to access the resources in AWS

Risks: Accessing the resources through public IPs in the infrastructure exposes the IP addresses of the infrastructure. This would allow administrators to configure access to sensitive ports on the instance security groups. This might result in enumeration, followed by exploitation of the instances in AWS



**Remediations:** Configure AWS VPN or direct connect in the AWS infrastructure to gain the management access to the infrastructure instead of connecting over internet. This would reduce the possibility of exposure. This would add an additional layer of encryption on the data at transit.

#### Medium Vulnerabilities in the infrastructure:

**Vulnerability:** Infrastructure is directly exposed to DDOS attacks due to lack of perimeter defense

**Risks:** With the Application hosted on one of the ec2 instances in AWS infrastructure, there is a high possibility of DOS attacks which might result in delay in processing genuine traffic, hence affects the availability of healthcare infrastructure.

**Remediations:** Enabling AWS shield in the AWS infrastructure is an added advantage to prevent downtime in the healthcare infrastructure. This would protect the infrastructure from any DOS attacks from the internet.

**Vulnerability:** Health Care Infrastructure is susceptible to Application-Layer attacks

**Risks:** As the applications are directly exposed to internet, the application might be susceptible to application layer attacks such as SQL injection, cross site scripting, CSRF, SSRF etc. This might result in unauthorized access, data theft and site defacement.

**Remediations:** Enable AWS WAF that protects the applications from OWASP top 10 attack vectors, prevent the applications to be exploited.

**Vulnerability:** Network Access Lists Provides full access to internet

**Risks:** It is identified that the NACLs are not configured as per the best practices as it would provide full access for the instances to access the internet. This might result in data exposure in the form of covert channels or instance might communicate with internet for Command-and-control communication

**Remediations:** Restrict the outgoing communication at the perimeter level using NACL's for only the authorized ports

## 5.Conclusion

Network plays a huge role in Health care infrastructure. Securing the network components in the infrastructure prevent the threat actor to perform enumeration otherwise, at the perimeter level

The report highlighted the risks those are the results of misconfigurations which made the components vulnerable. Following the remediations build a robust perimeter defense and protects the services in the health care infrastructure.

## 6. References

1. <https://docs.aws.amazon.com/vpc/latest/userguide/vpc-security-best-practices.html>
2. <https://docs.aws.amazon.com/vpc/latest/userguide/configure-subnets.html>
3. <https://docs.aws.amazon.com/waf/latest/developerguide/shield-chapter.html>
4. <https://docs.aws.amazon.com/waf/latest/developerguide/how-aws-waf-works.html>
5. <https://docs.aws.amazon.com/devicefarm/latest/developerguide/amazon-vpc-cross-region.html>

# DISASTER RECOVERY ASSESSMENT REPORT

## Table of Contents

1. Executive Summary
2. Scan results
3. Findings
4. Current State analysis, Risks and Remediations
5. Configuration best practices for current infrastructure
6. Conclusion
7. References

## 1. Executive Summary

The project involves assessing a healthcare organization's cloud infrastructure security, which is crucial for managing and preserving sensitive patient data. This includes personal data, medical records, and billing information. The main objectives are to identify the infrastructure misconfigurations, weaknesses, architecture lapses and propose the remediations and best practices to mitigate them to maintain the confidentiality, integrity, and availability of the application. The review will include several cloud security issues, such as managing vulnerabilities, securing the network, limiting user access, and encrypting data.

The healthcare company makes considerable use of the cloud to store patient data, run telemedicine consultations, and host essential healthcare apps. Unfortunately, due to misconfigurations and gaps in security, the system is very susceptible to potential assaults. Ensuring the quality of medical treatment provided and safeguarding confidential patient data are critical goals of this project.

## 2. Scan Results

We have included the Disaster recovery assessment report that are consolidated from the scans of AWS config, an open-source tool “prowler” and best practices from AWS documentations.

**Scan Results** - The scan results provide granular detail of each vulnerability, which are categorized by their severity: critical, high, medium, and low. An expanded definition of the known threat and solutions for remediating the vulnerability are also available.

## 3. Findings

Severity	Count
High	7
Medium	1

## 4. Current State analysis, Risks and Remediations:

The current infrastructure is configured with virtual machines, Databases, and storage devices; However, the infrastructure best practices have not been followed which led to below vulnerabilities.

### High Vulnerabilities in current infrastructure:

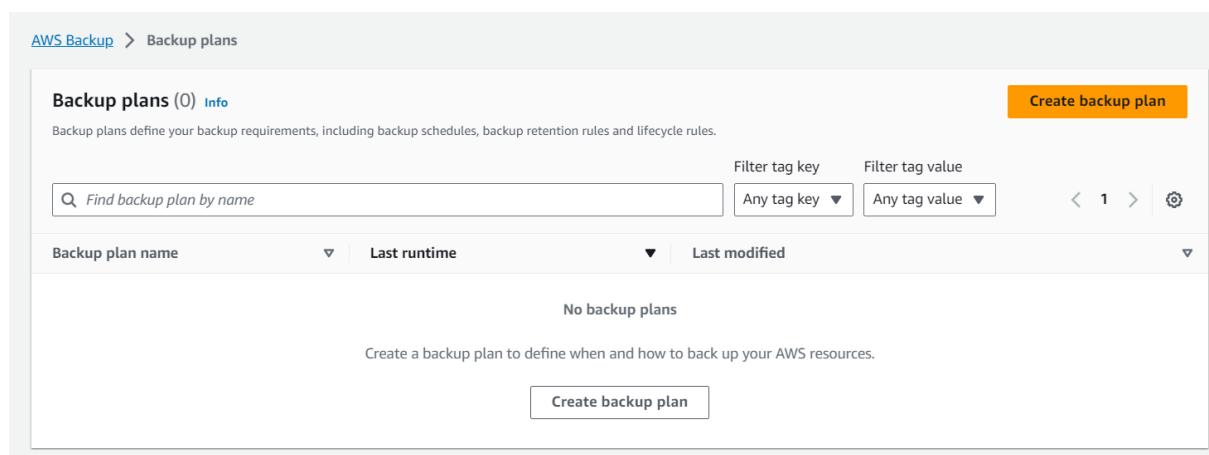
s.no	Vulnerability	Risks involved
1	Irregular Backups	Data loss and extended downtime
2	Multi-AZ strategy not implemented	Data loss and infrastructure runs on single point of failure

3	Multi Region strategy not implemented	Data loss and non-resiliency
4	Incremental Backup strategy not implemented	Data loss, Operational inefficiency, lack of updated configurations
5	Lack of encryption on backups	Data exfiltration
6	Lack of comprehensive testing plan	No resiliency, single point of failure
7	Failure to monitor backup jobs	Lack of Data integrity, Delayed response for failures, Data loss

### Vulnerability: Irregular Backups

It is identified that the automated backups are not enabled in the infrastructure.

From the configuration, it is identified that the account does not use the backup service which initiates automated backups



- Backups play a significant role during the events of disaster or security incident like ransomware attacks in the infrastructure, Lack of regular backups may increase the likelihood of permanent loss of data.
- Disaster recovery time is more during the time of an incident.
- Non compliancy with HIPAA standard related to data retention.
- Loss of productivity
- Irregular backups lead to data inconsistencies hence affecting data integrity.
- Loss of intellectual property

## Remediation:

- Enable scheduling Automated Backups for the resources like EC2, RDS
- Implement AWS backup service for regular and consistent backups
- Automated snapshot backups can be implemented for EBS volumes
- Enable Lifecycle policies for S3 Bucket implementations
- Enable versioning for S3 Buckets to create multiple versions of an object
- Cross-Region Backups helps in disaster recovery in case of failure of entire region
- Perform regular audits for backup policies

## Vulnerability: Multi-AZ strategy not implemented

In the current infrastructure, the resources are only created in “US-EAST-1a” availability zone and the resources are not spanned across the availability zones, Failure in the AZ would result in downtime for the infrastructure resources.

Details

Security

Networking

Storage

Status checks

Monitoring

Tags

▼ Networking details

Info

Public IPv4 address

54.225.44.254 |open address

Public IPv4 DNS

-

Subnet ID

subnet-09c6aa0f777670d0a (MIDTERM-CREATEINFRASTRUCTURE-Public-A)

Availability zone

us-east-1a

Use RBN as guest OS hostname

Disabled

Private IPv4 addresses

10.1.10.95

Private IP DNS name (IPv4 only)

ip-10-1-10-95.ec2.internal

IPv6 addresses

-

Carrier IP addresses (ephemeral)

-

Answer RBN DNS hostname IPv4

Disabled

VPC ID

vpc-064bde761a0de7fa (MIDTERM-CREATEINFRASTRUCTURE-VPC)

Secondary private IPv4 addresses

-

Outpost ID

-

▼ Network Interfaces (1)

Info

- Multi-AZ strategy is an approach that involves replicating the resources and services across multiple availability zones in an AWS region
- It is mandatory to implement this strategy for the critical health care infrastructure to avoid loss of patient data and records in case of a disaster in an AWS availability zone.
- The strategy must involve synchronous and continuous data replication of resources across the availability zones to avoid data inconsistencies.
- It must also include the configurations for automated failover from primary zone to secondary zone in the event of a failure.
- Efficient Multi-AZ strategy helps the health care organizations meet compliance requirements and result in business continuity
- Failure to implement the multi-AZ strategy in the healthcare infrastructure is prone to below mentioned risks
  - Service disruption during AZ outages
  - Disruption in Patient’s data integrity
  - Delayed recovery times

- HIPAA violation
- Sensitive Data exposure

### Remediations:

- Implement Multi-AZ deployments wherever needed
- Automated Data synchronization process across the AZ's
- Regular DR drill to test the procedures
- Perform security Audit regularly.

### Vulnerability: Multi-region Strategy Not implemented

In the current infrastructure, the resources are deployed in single region and the resources are not spanned across the region which makes disaster recovery critical.

Details

Security

Networking

Storage

Status checks

Monitoring

Tags

▼ Networking details

Info

Public IPv4 address

54.225.44.254

open address

Public IPv4 DNS

-

Subnet ID

subnet-09c6aa0f777670d0a

(MIDTERM-CREATEINFRASTRUCTURE-Public-A)

Availability zone

us-east-1a

Use RBN as guest OS hostname

Disabled

Private IPv4 addresses

10.1.10.95

Private IP DNS name (IPv4 only)

ip-10-1-10-95.ec2.internal

IPv6 addresses

-

Carrier IP addresses (ephemeral)

-

Answer RBN DNS hostname IPv4

Disabled

VPC ID

vpc-064bde761a0de7fa

(MIDTERM-CREATEINFRASTRUCTURE-VPC)

Secondary private IPv4 addresses

-

Outpost ID

-

▼ Network Interfaces (1)

Info

Multi-Region strategy is the approach that the organization spans their infrastructure across the regions to improve the high availability, efficiently implement disaster recovery and protect the data sovereignty.

- This strategy must be implemented to improve the high availability of critical health care systems
- This strategy helps to protect the data integrity in the healthcare infrastructure
- Helps the organizations demonstrate regulatory compliance with different standards like HIPAA etc.
- Failure to implement the multi-region strategy involves below mentioned risks
  - Infrastructure is in single point of failure and any event of failure in one region results in downtime
  - Increased Data loss during the time of a disaster

- Compliance violation

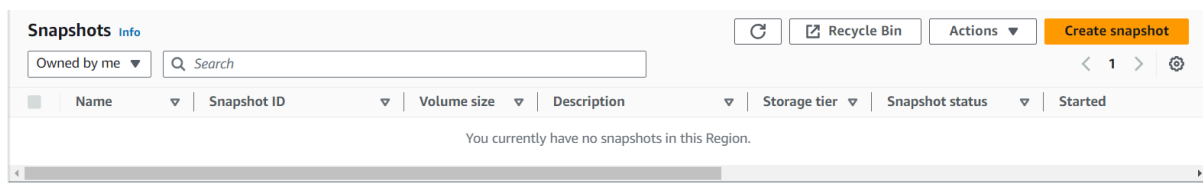
### Remediations:

- Implement Multi Region deployment wherever needed
- Implement Cross-region backup and redundancy for the resources

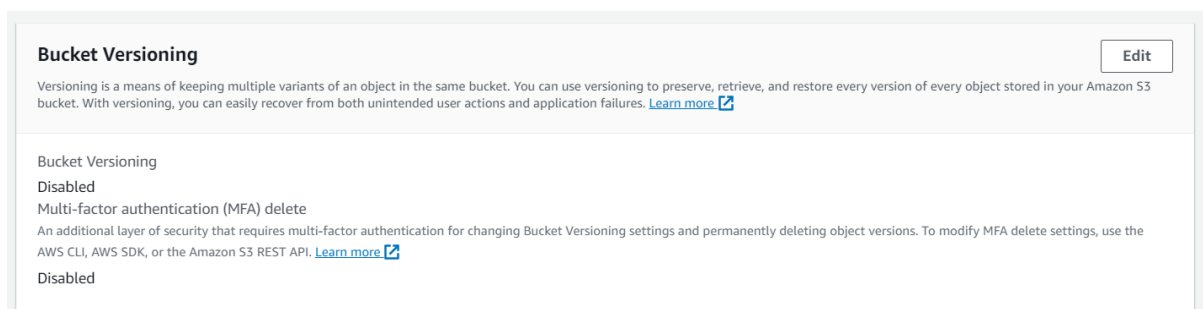
### Vulnerability: Incremental Backup strategy not implemented

Incremental Backups/Versioning are not implemented in the infrastructure.

### EBS volumes are not configured to take snapshots:



### Versioning not enabled:



Incremental backups help the organizations to retain the newest revision of the data that has been backed up in an automated way.

- Incremental Backups helps the organization to restore the state of the instances just before any security incident occurred
- This strategy protects from the data loss and reduces the recovery time of any resources in an event of failure
- Failure to configure incremental backups led to below risks
  - Data loss
  - Inconsistency in data
  - Loss of intellectual property
  - Increased downtime





## Remediations:

- Implement versioning on S3 Instances allow organizations to have multiple versions of a single object so that any drifts in data can be restored back to the old version
- Implement Lifecycle policies on S3 objects to transition data to different storage classes
- Use the AWS Backup service to take automated incremental backups of the resources like EBS, RDS, Dynamo DB.
- Configure the EBS to take automated scheduled snapshots
- Retention policies configuration allows an automated way of deletion of snapshots after a mentioned time.

## Vulnerability: Lack of encryption on backups

EBS volumes are not encrypted in the infrastructure.

Volume ID: vol-0e2bab5f69015a26c			
commendations.   <a href="#">Learn more</a>			
Encryption	KMS key ID	KMS key alias	KMS key ARN
Not encrypted	-	-	-
Fast snapshot restored	Snapshot	Availability Zone	Created
No	 snap-019159f1e06f32720	us-east-1a	 Sat Nov 04 2023 11:26:57 GMT-0400 (Eastern Daylight Time)
Multi-Attach enabled	Attached Instances	Outposts ARN	
No	<a href="#">i-0df9a90b347e52334</a> : /dev/xvda (attached)	-	

Encryption plays an important role in preserving Data integrity in the healthcare infrastructure. Encryption at rest protects the data from exposure.

- Encryption of sensitive data on backups helps in data protection, prevent data breaches, prevent data tampering.
- Encryption does protect the data from leakage
- Lack of encryption results in below risks
  - Data loss
  - Exfiltration of sensitive data

## Remediations:

- Enable Server-Side encryption using AWS KMS (Key Management Service) and rotate the keys present in AWS KMS.
- Implement Encryption on EBS snapshots, RDS Backups and on AWS Backups
- Encrypt S3 buckets
- Maintain and rotate the keys using KMS (Key Management Service)

## Vulnerability: Lack of comprehensive testing plan

DC-DR drills are mandatory to test the resiliency of the infrastructure for business continuity operations in terms of failure.

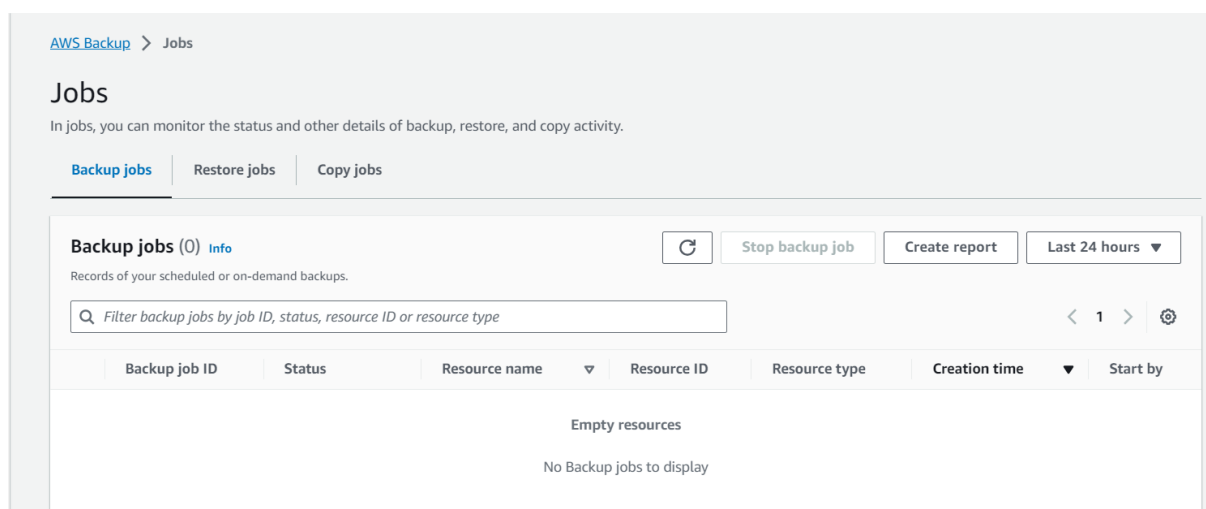
### Risks:

- Lack of robust testing plan leads to discrepancies in data during a disaster
- Data loss and downtime during the incidents of failure

### Remediations:

- Prepare a comprehensive DC-DR drill plan and implement a quarterly DC-DR drill to check the resiliency in the infrastructure.

### Vulnerability: Failure to monitor backup jobs



Monitoring must be enabled for scheduled automated backup jobs. This helps preserving data integrity in healthcare infrastructure.

Failure to Monitor backup jobs resulted in below risks:

- Undetected backup jobs left the critical data unprotected
- Lack of monitoring the backup jobs leaves the corrupted backup jobs undetected
- Involve Data loss and makes the data recovery complex
- The infrastructure does not adhere to compliance standards.

### Remediations:

- Document a plan to review backup jobs regularly
- Enable monitoring for automated backups
- Set alerts and notifications for the events of backup failures and corruption
- Perform regular policy audit

### Medium Vulnerabilities in Current Infrastructure:

S.No	Vulnerability	Risks
1	Improper Configuration of Backup Policies	Data Loss, Extended Downtime and non-compliance

**Vulnerability:** Improper Configuration of Backup Policies:

Improper Backup policies in the current infrastructure led to below risks

- Lack of versioning on S3 buckets will leads to zero recovery of data after a overwrite
- Incorrect or Insufficient data retention policies observed in the infrastructure results in deletion of the historical patient data
- S3 buckets are configured with improper permissions that leads to sensitive data exposure
- Over dependency on Snapshot backups for data retention instead of using AWS backup service leads to delay in restoration.

### Remediations:

- Enable AWS backup service in AWS
- Follow Principle of least privilege for configuring S3 buckets

Besides these vulnerabilities, it is suggestable to enable AWS config in AWS that scans automatically on the event of configuration drift and notify the administrator with the misconfigurations in the infrastructure.

## 5. Configuration best practices for current infrastructure

For Databases: The below recommended sample configuration overcomes the risks involved with backups on database.

Key components in the configuration best practices for databases are

- Configuring proper retention period to eliminate the risk for data loss
- Encryption enabled on backups all the time to protect data exfiltration and integrity
- Notifications and alerts are set for automated backup to notify the administrator about corrupted backup jobs or insufficient backups
- Configure the policy for compliance so that the backups follow those specific standards while backing up the data

```
{
  "backupConfigurations": [
    {
      "name": "EHR Database Backup",
      "source": "/var/db/ehr",
      "destination": "s3://healthcare-backups/ehr_database",
      "frequency": "daily",
      "retention": {
        "daily": 7,
        "weekly": 4,
        "monthly": 12
      },
      "encryption": {
        "enabled": true,
        "algorithm": "AES256"
      },
      "compression": "gzip",
      "notifications": {
        "email": ["admin@healthcareprovider.com"],
        "slack": ["#backup-notifications"]
      },
      "compliance": "HIPAA"
    }
  ]
}
```

### For EBS Volumes:

This configuration is very similar to Databases; besides that, it is recommended to take the automated snapshot backups of the EBS volumes in the infrastructure

```
{
  "ebsVolumeConfigurations": [
    {
      "name": "EHR Database Volume",
      "volumeType": "gp2",
      "sizeGB": 100,
      "encrypted": true,
      "kmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/abcdef01-2345-6789-0abc-def012345678",
      "iops": 3000,
      "tags": {
        "Environment": "Production",
        "Application": "EHR System"
      },
      "snapshots": {
        "frequency": "daily",
        "retention": {
          "daily": 7,
          "weekly": 4,
          "monthly": 12
        }
      },
      "compliance": "HIPAA"
    }
  ],
}
```

### For S3 Buckets:

It is essential to follow the best practices to secure the data in health care infrastructure,

The most notable components from the below best practices configuration specific to Disaster recovery are identified as follows:

- Versioning must be true to that prevents data loss in the infrastructure due to configuration changes
- Life cycle policies must be set to true that automatically deletes the objects based on the policy
- Cross region replication must be enabled to provide high availability of data
- “MFADelete” must be true that needs a multi factor authentication to delete an object

```
{
  "bucketConfigurations": [
    {
      "name": "my-production-bucket",
      "bucketNaming": "healthcare-domain",
      "versioning": true,
      "accessControl": "IAM-Policies-and-ACLs",
      "bucketPolicies": true,
      "objectLevelPermissions": true,
      "encryption": "SSE-S3 or SSE-KMS",
      "accessLogging": true,
      "crossRegionReplication": true,
      "lifecyclePolicies": true,
      "objectTagging": true,
      "requesterPays": false,
      "mfaDelete": true,
      "monitoringAndAlerts": true,
      "inventoryReports": true,
      "staticWebsiteHosting": true,
      "accessPoints": true,
      "complianceAndDataGovernance": "HIPAA"
    },
  ],
}
```

### For Snapshot Backups:

Below Mentioned configurations provides the best practices for Snapshot backups in the Healthcare infrastructure.

Here is the explanation of the same

- Permissions to create snapshot must be permitted to backup user but deletion must be permitted only to backup admin
- Scheduled automatic backups must be enabled every day
- Server-side encryption Must to be enabled and key must be rotated using KMS
- Snapshot backups must be synchronized across the regions as a measure of disaster recovery
- Life-Cycle policies for backups must be configured properly for retention
- Monitoring and notifications must be enabled to for backup jobs

```
{
  "snapshotBackupConfig": {
    "backupSchedule": "cron(0 2 * * ? *)",
    "automateSnapshotCreation": true,
    "tags": {
      "Application": "MyApp",
      "Environment": "Production"
    },
    "encryption": "SSE-S3",
    "permissions": {
      "createSnapshot": ["arn:aws:iam::545361066637:user/backup-user"],
      "deleteSnapshot": ["arn:aws:iam::545361066637:user/backup-admin"]
    },
    "copyToDifferentRegion": true,
    "retentionPolicy": {
      "daily": 7,
      "weekly": 4,
      "monthly": 12
    },
    "lifecyclePolicy": {
      "transitionToColdStorageAfterDays": 30,
      "deleteAfterDays": 365
    },
    "monitoringAndAlerts": true,
    "testSnapshotRecovery": true
  }
}
```

## 6. Conclusion:

Automated Backups and Infrastructure best practices plays an important role for securing the health care infrastructure in AWS environment. Hence, it is imperative to plan the infrastructure that caters the needs if the user without compromising the security aspect involved while planning and building them. By implementing the mentioned remediations in the infrastructure protects the environment from data loss, downtimes caused by the ad-hoc incidents in the infrastructure.

## 7. References

- 1) <https://docs.aws.amazon.com/whitepapers/latest/disaster-recovery-workloads-on-aws/disaster-recovery-options-in-the-cloud.html>
- 2) [https://docs.aws.amazon.com/drs/latest/userguide/best\\_practices\\_drs.html](https://docs.aws.amazon.com/drs/latest/userguide/best_practices_drs.html)
- 3) <https://aws.amazon.com/blogs/architecture/disaster-recovery-dr-architecture-on-aws-part-i-strategies-for-recovery-in-the-cloud/>
- 4) <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSSnapshots.html>
- 5) <https://docs.aws.amazon.com/AmazonS3/latest/userguide/Versioning.html>
- 6) <https://docs.aws.amazon.com/aws-backup/latest/devguide/monitoring.html>