TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN, ĐHQG-HCM

KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

BÁO CÁO ĐỒ ÁN MÔN HỌC

Amazon Web Services Foundations

Môn học: NT542.Q11 - Lập trình kịch bản tự động hóa cho quản trị và bảo mật mạng

Giảng viên hướng dẫn: TS. Trần Thị Dung, ThS. Văn Thiên Luân

Thực hiện bởi nhóm 17, bao gồm:

1. Phạm Hữu Thắng - 22521328 Trưởng nhóm
2. Lương Cao Thắng - 22521340 Thành viên
3. Phạm Xuân Tuấn Anh - 22520071 Thành viên

Thời gian thực hiện: -

MỤC LỤC

BÁO CÁO CHI TIẾT

1. TỔNG QUAN BENCHMARK

1.1 TỔNG QUAN

CIS Amazon Web Services (AWS) Foundations Benchmark v6.0.0 là bộ hướng dẫn bảo mật được phát triển bởi Center for Internet Security (CIS), cung cấp các khuyến nghị cấu hình kỹ thuật nhằm xây dựng một nền tảng bảo mật vững chắc cho môi trường AWS.

Mục tiêu của Benchmark là thiết lập baseline an toàn, giúp các tổ chức đảm bảo tuân thủ và phòng chống các mối đe dọa mạng. Các khuyến nghị được xây dựng dựa trên đồng thuận của cộng đồng chuyên gia toàn cầu, đảm bảo tính thực tiễn và khả năng áp dụng cao.

Benchmark này được chia thành các hạng mục chính, tương ứng với các lĩnh vực bảo mật quan trọng nhất trong môi trường AWS:

- Quản lý Danh tính và Truy cập (Identity and Access Management - IAM): Tập trung vào việc bảo mật tài khoản gốc (root), thực thi các chính sách mật khẩu mạnh, áp dụng xác thực đa yếu tố (MFA), và quản lý quyền truy cập theo nguyên tắc đặc quyền tối thiểu.
- Lưu trữ (Storage): Cung cấp các hướng dẫn để bảo mật các dịch vụ lưu trữ phổ biến như Amazon S3, Relational Database Service (RDS) và Elastic File System (EFS), bao gồm việc mã hóa dữ liệu, chặn truy cập công khai và cấu hình các chính sách truy cập an toàn.
- Ghi nhật ký (Logging): Nhấn mạnh tầm quan trọng của việc ghi lại các hoạt động và thay đổi trong môi trường AWS thông qua các dịch vụ như AWS CloudTrail, AWS Config và VPC Flow Logs, đảm bảo khả năng phân tích bảo mật, theo dõi thay đổi và kiểm toán tuân thủ.
- Giám sát (Monitoring): Hướng dẫn thiết lập các cơ chế giám sát thời gian thực bằng cách tích hợp nhật ký CloudTrail với CloudWatch. Điều này cho phép tạo

các bộ lọc và cảnh báo tự động cho các hoạt động đáng ngờ như các lệnh gọi API trái phép, việc sử dụng tài khoản root, hoặc các thay đổi trái phép đối với các cấu hình bảo mật quan trọng.

- Mạng (Networking): Đưa ra các khuyến nghị để củng cố hạ tầng mạng, bao gồm việc cấu hình chặt chẽ các Security Group và Network Access Control List (NACL), hạn chế truy cập vào các cổng quản trị từ xa, và bảo mật các kết nối VPC Peering.

CIS Benchmark gồm hai cấp độ cấu hình:

- Level 1: Các khuyến nghị cơ bản, ít ảnh hưởng đến vận hành.

- Level 2: Bảo mật nâng cao, dành cho môi trường yêu cầu an ninh nghiêm ngặt.

1.2 NỘI DUNG

1.2.1 Giới thiệu:

CIS Amazon Web Services Foundations Benchmarks:

- Benchmark này cung cấp hướng dẫn quy tắc cho việc cấu hình một tập hợp các dịch vụ AWS với sự nhấn mạnh vào các cài đặt nền tảng, có thể kiểm tra và không phụ thuộc vào kiến trúc.
- Do môi trường của các nhà cung cấp dịch vụ đám mây (CSP) liên tục thay đổi, người dùng nên luôn sử dụng phiên bản mới nhất của Foundations Benchmark. Các phiên bản trước có thể chứa tên sản phẩm không chính xác, quy trình lỗi thời và các thông tin không chính xác khác.

CIS AWS Service Category Benchmarks:

- Sau khi cấu hình môi trường với Foundations Benchmark, bước tiếp theo là sử dụng các Service Category Benchmark để có các khuyến nghị phòng thủ theo chiều sâu và dành riêng cho từng dịch vụ.
- Các khuyến nghị trong các benchmark này chỉ nên được áp dụng cho các sản phẩm và dịch vụ CSP đang được sử dụng tích cực trong môi trường của bạn.

Việc áp dụng các khuyến nghị không cần thiết có thể gây ra lỗ hổng, nợ kỹ thuật và chi phí không cần thiết.

1.2.2 Quản lý danh tính và truy cập (Identity and Access Management - IAM):

Đây là phần nền tảng, tập trung vào việc kiểm soát ai có thể truy cập vào tài nguyên nào trong môi trường AWS của bạn.

- Duy trì thông tin liên hệ: Đảm bảo thông tin liên hệ (email và điện thoại) của tài khoản AWS luôn được cập nhật và trỏ đến nhiều hơn một cá nhân để AWS có thể liên lạc trong trường hợp có hoạt động đáng ngờ hoặc vi phạm chính sách. Ngoài ra, cần đăng ký thông tin liên hệ bảo mật riêng để nhóm bảo mật của bạn nhận được các thông báo phù hợp từ AWS.
- Không tồn tại khóa truy cập cho tài khoản 'root': Tài khoản root là người dùng có đặc quyền cao nhất. Việc xóa tất cả các khóa truy cập (access keys) liên quan đến tài khoản này sẽ hạn chế các vector tấn công và khuyến khích việc sử dụng các tài khoản dựa trên vai trò với đặc quyền tối thiểu.
- Bật Xác thực Đa yếu tố (MFA) cho tài khoản 'root': MFA cung cấp một lớp bảo vệ bổ sung cho việc đăng nhập. Khuyến nghị Cấp độ 1 yêu cầu bật MFA cho tài khoản root , trong khi Cấp độ 2 khuyến nghị sử dụng MFA phần cứng để có bề mặt tấn công nhỏ hơn so với MFA ảo (trên điện thoại thông minh).
- Hạn chế sử dụng tài khoản 'root': Tránh sử dụng tài khoản root cho các công việc hàng ngày và quản trị. Việc sử dụng tài khoản này không phù hợp với nguyên tắc đặc quyền tối thiểu và có thể dẫn đến thiệt hại không cần thiết do lỗi hoặc bị xâm phạm.
- Thực thi chính sách mật khẩu mạnh: Cấu hình chính sách mật khẩu IAM yêu cầu độ dài tối thiểu là 14 ký tự và ngăn chặn việc tái sử dụng mật khẩu (khuyến nghị nhớ 24 mật khẩu gần nhất). Điều này làm tăng khả năng chống lại các cuộc tấn công dò mật khẩu (brute force).
- Bật MFA cho tất cả người dùng IAM có mật khẩu console: Tất cả người dùng có quyền truy cập vào Bảng điều khiển quản lý AWS (AWS Management Console) nên được bật MFA để tăng cường bảo mật.

- Không tạo khóa truy cập khi thiết lập người dùng ban đầu: Yêu cầu người dùng thực hiện các bước bổ sung để tạo khóa truy cập sau khi tài khoản của họ đã được tạo sẽ cung cấp một chỉ báo mạnh mẽ hơn về ý định rằng các khóa này thực sự cần thiết cho công việc của họ.
- Vô hiệu hóa thông tin xác thực không sử dụng trong 45 ngày hoặc hơn: Các thông tin xác thực (mật khẩu, khóa truy cập) không được sử dụng trong 45 ngày trở lên nên được vô hiệu hóa hoặc xóa để giảm thiểu cơ hội bị lợi dụng nếu tài khoản bị xâm phạm hoặc bị bỏ quên.
- Quản lý vòng đời khóa truy cập: Mỗi người dùng IAM chỉ nên có một khóa truy cập hoạt động duy nhất. Các khóa truy cập nên được xoay vòng (rotated) định kỳ, ít nhất mỗi 90 ngày một lần, để giảm thiểu rủi ro từ các khóa bị mất, bị bẻ khóa hoặc bị đánh cắp.
- Cấp quyền thông qua Nhóm (Groups): Người dùng IAM chỉ nên nhận quyền thông qua việc được thêm vào các nhóm. Việc này thống nhất việc quản lý quyền, phù hợp với vai trò chức năng của tổ chức và giảm khả năng cấp quyền thừa.
- Không gắn chính sách cho phép toàn quyền quản trị (":"): Tránh sử dụng các chính sách IAM cho phép toàn quyền ("Effect": "Allow", "Action": "*", "Resource": "*"). Thay vào đó, hãy bắt đầu với một bộ quyền tối thiểu và cấp thêm khi cần thiết.
- Tạo vai trò hỗ trợ (Support Role): Tạo một vai trò IAM riêng với chính sách AWSSupportAccess để người dùng được ủy quyền có thể quản lý các sự cố với bộ phận Hỗ trợ của AWS, tuân thủ nguyên tắc đặc quyền tối thiểu.
- Sử dụng Vai trò IAM cho EC2 Instances: Các ứng dụng chạy trên EC2 instance nên sử dụng vai trò IAM để truy cập tài nguyên AWS thay vì lưu trữ khóa truy cập dài hạn trong mã nguồn hoặc tệp cấu hình. Điều này giúp giảm rủi ro liên quan đến việc chia sẻ và xoay vòng thông tin xác thực.
- Xóa chứng chỉ SSL/TLS đã hết hạn: Xóa các chứng chỉ SSL/TLS đã hết hạn được lưu trữ trong IAM để loại bỏ nguy cơ một chứng chỉ không hợp lệ vô tình được triển khai, gây ảnh hưởng đến uy tín của ứng dụng.

- Bật IAM External Access Analyzer ở tất cả các khu vực: Dịch vụ này giúp xác định các tài nguyên (như S3 buckets, IAM roles) được chia sẻ với các thực thể bên ngoài, cho phép bạn xác định và loại bỏ quyền truy cập ngoài ý muốn.
- Quản lý người dùng IAM một cách tập trung: Trong môi trường đa tài khoản, việc quản lý người dùng nên được tập trung thông qua liên kết danh tính (identity federation) với một nhà cung cấp danh tính bên ngoài hoặc sử dụng AWS Organizations. Điều này làm giảm sự phức tạp và khả năng xảy ra lỗi trong quản lý truy cập.
- Hạn chế quyền truy cập vào AWSCloudShellFullAccess: Chính sách này cho phép người dùng tải tệp lên và xuống môi trường CloudShell, tiềm ẩn nguy cơ rò rỉ dữ liệu. Do đó, quyền truy cập vào chính sách này nên được hạn chế.

1.2.3 Lưu trữ (Storage):

Phần này tập trung vào việc bảo mật các dịch vụ lưu trữ dữ liệu của AWS.

- Đảm bảo chính sách S3 Bucket từ chối các yêu cầu HTTP: Cấu hình chính sách bucket để từ chối các yêu cầu không được mã hóa (HTTP) và chỉ cho phép truy cập qua HTTPS, bảo vệ dữ liệu khi đang truyền.
- Bật MFA Delete trên S3 Buckets: Bật tính năng này yêu cầu người dùng phải cung cấp hai hình thức xác thực khi thay đổi trạng thái phiên bản (versioning) của bucket hoặc xóa một phiên bản đối tượng, thêm một lớp bảo mật chống lại việc xóa vô tình hoặc ác ý.
- Khám phá, phân loại và bảo mật dữ liệu trong S3: Sử dụng các công cụ như Amazon Macie hoặc các công cụ của bên thứ ba để tự động phát hiện, phân loại và bảo vệ dữ liệu nhạy cảm được lưu trữ trong các S3 bucket.
- Bật 'Chặn tất cả quyền truy cập công khai' (Block Public Access) cho S3: Cài đặt này ngăn chặn việc vô tình hoặc cố ý để lộ dữ liệu ra công chúng ở cấp độ bucket hoặc toàn bộ tài khoản.
- Bật mã hóa khi lưu trữ (encryption-at-rest) cho RDS Instances: Sử dụng thuật toán mã hóa AES-256 để mã hóa dữ liệu trên máy chủ lưu trữ các RDS instance, cũng như các bản sao lưu, read replicas và snapshots.

- Bật tự động nâng cấp phiên bản phụ (Auto Minor Version Upgrade) cho RDS: Tính năng này đảm bảo các RDS instance tự động nhận các bản nâng cấp phụ, bao gồm các tính năng mới, bản vá lỗi và bản vá bảo mật.
- Đảm bảo RDS Instances không thể truy cập công khai: Việc này giảm thiểu rủi ro bị tấn công từ Internet, chẳng hạn như tấn công brute force hoặc SQL injection.
- Sử dụng triển khai Multi-AZ cho RDS: Cấu hình này cung cấp tính sẵn sàng và độ bền cao bằng cách sao chép dữ liệu đồng bộ sang một instance dự phòng ở một Availability Zone (AZ) khác, cho phép tự động chuyển đổi dự phòng khi có sự cố.
- Đảm bảo mã hóa được bật cho EFS File Systems: Dữ liệu trên EFS nên được mã hóa khi lưu trữ bằng AWS Key Management Service (KMS) để giảm rủi ro vi phạm dữ liệu thông qua truy cập trực tiếp vào thiết bị lưu trữ.

1.2.4 Ghi nhật ký (Logging):

- Bật CloudTrail ở tất cả các khu vực: AWS CloudTrail ghi lại các lệnh gọi API cho tài khoản của bạn. Việc bật nó trên nhiều khu vực (multi-region) giúp phát hiện hoạt động không mong muốn ở các khu vực ít được sử dụng và ghi lại các sự kiện dịch vụ toàn cầu.
- Bật xác thực tệp nhật ký CloudTrail (Log File Validation): Tính năng này tạo ra một tệp digest được ký kỹ thuật số, có thể được sử dụng để xác minh rằng các tệp nhật ký không bị thay đổi hoặc xóa sau khi được CloudTrail gửi đi.
- Bật AWS Config ở tất cả các khu vực: Dịch vụ này thực hiện quản lý cấu hình các tài nguyên AWS được hỗ trợ và ghi lại lịch sử thay đổi cấu hình, cho phép phân tích bảo mật, theo dõi thay đổi và kiểm toán tuân thủ.
- Bật ghi nhật ký truy cập máy chủ (Server Access Logging) trên CloudTrail S3 Bucket: Tính năng này tạo ra một bản ghi chi tiết về mỗi yêu cầu được thực hiện đến S3 bucket chứa nhật ký CloudTrail của bạn, hữu ích cho các quy trình bảo mật và ứng phó sự cố.
- Mã hóa nhật ký CloudTrail bằng KMS CMK: Sử dụng mã hóa phía máy chủ (SSE) với Khóa chính do khách hàng quản lý (CMK) trong KMS để bảo vệ nhật

ký CloudTrail, cung cấp thêm một lớp kiểm soát bảo mật. Người dùng cần có cả quyền đọc S3 và quyền giải mã từ chính sách CMK để truy cập nhật ký.

- Bật xoay vòng (rotation) cho các khóa CMK đối xứng do khách hàng tạo: Việc xoay vòng khóa mã hóa giúp giảm tác động tiềm tàng của một khóa bị xâm phạm, vì dữ liệu được mã hóa bằng khóa mới sẽ không thể truy cập được bằng khóa cũ đã bị lộ.
- Bật VPC Flow Logging trong tất cả các VPC: Tính năng này ghi lại thông tin về lưu lượng IP đi và đến các giao diện mạng trong VPC của bạn, cung cấp khả năng hiển thị để phát hiện lưu lượng bất thường hoặc thu thập thông tin chi tiết trong quá trình điều tra bảo mật.
- Bật ghi nhật ký cấp đối tượng (Object-Level Logging) cho S3 Buckets: CloudTrail theo mặc định không ghi lại các sự kiện dữ liệu (ví dụ: GetObject, PutObject). Việc bật ghi nhật ký cho cả sự kiện ghi (write) và đọc (read) ở cấp độ đối tượng giúp đáp ứng các yêu cầu tuân thủ, phân tích bảo mật và giám sát hành vi người dùng.

1.2.5 Giám sát (Monitoring):

Phần này hướng dẫn cách thiết lập giám sát và cảnh báo tự động cho các hoạt động bảo mật quan trọng.

- Thiết lập Metric Filters và Alarms trong CloudWatch: Các khuyến nghị này đề xuất tạo các bộ lọc và cảnh báo để giám sát trong thời gian thực các sự kiện quan trọng được ghi lại trong nhật ký CloudTrail, bao gồm:
    - Các lệnh gọi API trái phép (Unauthorized API calls).
    - Đăng nhập vào Bảng điều khiển quản lý mà không có MFA.
    - Việc sử dụng tài khoản root.
    - Các thay đổi đối với chính sách IAM.
    - Các thay đổi cấu hình CloudTrail.
    - Các lần xác thực thất bại vào Bảng điều khiển quản lý AWS.
    - Việc vô hiệu hóa hoặc lên lịch xóa các khóa CMK do khách hàng tạo.

  - Các thay đổi chính sách của S3 bucket.
  - Các thay đổi cấu hình AWS Config.
  - Các thay đổi đối với Security Group.
  - Các thay đổi đối với Network Access Control List (NACL).
  - Các thay đổi đối với Network Gateway.
  - Các thay đổi đối với Bảng định tuyến (Route Table).
  - Các thay đổi đối với VPC.
  - Các thay đổi đối với AWS Organizations.
- Bật AWS Security Hub: Dịch vụ này tổng hợp, sắp xếp và ưu tiên các phát hiện bảo mật từ nhiều dịch vụ AWS khác nhau (như GuardDuty, Inspector, Macie) và các sản phẩm của đối tác, cung cấp một cái nhìn toàn diện về tình trạng bảo mật của bạn.

1.2.6 Mạng (Networking):

- Bật mã hóa EBS Volume ở tất cả các khu vực: Cấu hình để tất cả các ổ đĩa EBS mới được tạo đều được mã hóa theo mặc định. Việc này giảm thiểu nguy cơ lộ dữ liệu nếu có quyền truy cập trực tiếp vào thiết bị lưu trữ.
- Hạn chế truy cập CIFS đến các mạng tin cậy: Giao thức chia sẻ tệp CIFS (cổng 445) không nên được mở cho các mạng không tin cậy để ngăn chặn truy cập trái phép vào dữ liệu nhạy cảm.
- Không cho phép truy cập vào các cổng quản trị từ xa từ Internet: Đảm bảo rằng cả Network ACL và Security Group không cho phép truy cập vào các cổng quản trị như SSH (22) và RDP (3389) từ 0.0.0.0/0 (IPv4) hoặc ::/0 (IPv6). Việc để các cổng này mở ra Internet làm tăng đáng kể bề mặt tấn công.
- Hạn chế toàn bộ lưu lượng trên Security Group mặc định của mỗi VPC: Security Group mặc định nên được cấu hình để không có quy tắc nào cho phép lưu lượng vào hoặc ra. Điều này buộc người dùng phải tạo các Security Group với đặc quyền tối thiểu và gán tài nguyên vào đó một cách có chủ đích, giảm thiểu việc phơi

bày tài nguyên.

- Đảm bảo các bảng định tuyến cho VPC Peering tuân thủ "quyền truy cập tối thiểu": Khi kết nối các VPC với nhau, các bảng định tuyến chỉ nên cho phép lưu lượng đến các máy chủ hoặc mạng con cụ thể cần thiết, thay vì cho phép toàn bộ lưu lượng giữa hai VPC, để hạn chế tác động nếu một VPC bị xâm phạm.
- Đảm bảo EC2 Metadata Service chỉ cho phép IMDSv2: Instance Metadata Service Version 2 (IMDSv2) sử dụng phương thức định hướng phiên, an toàn hơn so với IMDSv1, và giúp bảo vệ chống lại các cuộc tấn công SSRF (Server-Side Request Forgery). Do đó, nên yêu cầu bắt buộc sử dụng IMDSv2 cho tất cả các EC2 instance.

2. PHƯƠNG PHÁP THỰC HIỆN

2.1 CẤU HÌNH THỦ CÔNG (MANUAL CONFIGURATION):

Các bước dưới đây yêu cầu thao tác trực tiếp trên AWS Management Console hoặc kiểm tra bằng công cụ quản trị. Chúng bao gồm các cấu hình khó tự động hóa, đòi hỏi xác nhận thủ công hoặc đánh giá theo ngữ cảnh.

Mục tiêu	**Giải thích**
Cập nhật và đăng ký thông tin liên hệ	Đảm bảo AWS liên hệ kịp thời khi phát hiện rủi ro bảo mật. Nên dùng email nhóm (security@domain.com) để tránh gián đoạn.
Bật Hardware MFA cho tài khoản Root (L2)	MFA phần cứng an toàn hơn OTP ảo, giảm nguy cơ lộ mã OTP qua malware.
Hạn chế sử dụng tài khoản Root	Root chỉ nên dùng cho tác vụ đặc biệt. Việc sử dụng thường xuyên làm tăng bề mặt tấn công.
Không tạo Access Key khi cấp console password cho IAM User	Tránh phát sinh khóa API không cần thiết, giảm rủi ro credential leak.

Bật RDS Multi-AZ	Đảm bảo khả năng chịu lỗi và tính sẵn sàng cao của cơ sở dữ liệu.
Bật CloudTrail tất cả vùng	Đảm bảo log được thu thập toàn diện cho mọi hoạt động API.
Thiết lập Metric Filter và SNS Alarm giám sát API quan trọng	Tạo cảnh báo khi phát hiện hành vi đáng ngờ (xóa log, sửa IAM policy, đăng nhập thất bại...).
Kích hoạt AWS Security Hub (L2)	Tự động tổng hợp cảnh báo bảo mật và chấm điểm tuân thủ CIS, PCI DSS.
Hạn chế truy cập mạng (NACL, SG, Default SG, Peering Route)	Giới hạn traffic 0.0.0.0/0 đến các cổng quản trị (22, 3389), tuân thủ nguyên tắc “least access”.

Ưu điểm:

- Dễ triển khai với môi trường nhỏ hoặc kiểm tra định kỳ.

- Phù hợp với các khuyến nghị 'Manual' trong CIS Benchmark.

Nhược điểm:

- Tốn thời gian và dễ bỏ sót khi kiểm tra hàng loạt tài khoản.

2.2 CẤU HÌNH TỰ ĐỘNG (AUTOMATION CONFIGURATION)

Các bước sau có thể triển khai thông qua **AWS CLI, boto3, hoặc Terraform**, đảm bảo nhất quán và có thể kiểm toán lại dễ dàng.
Mỗi cấu hình gồm: Mục tiêu → Cách kiểm tra → Lệnh/Terraform → Giải thích.

2.2.1 Quản lý Danh tính và Truy cập (IAM Security):

- Tự động hóa Chính sách Mật khẩu:
    - Phương pháp: Khai báo tài nguyên aws_iam_account_password_policy trong Terraform.
    - Chi tiết cấu hình: Thiết lập cứng các tham số: **minimum_password_length = 14, require_symbols = true,**

password_reuse_prevention = 24. Bất kỳ nỗ lực nào nhằm hạ thấp tiêu chuẩn này trên AWS Console sẽ bị Terraform ghi đè lại trong lần chạy pipeline kế tiếp.

- Giải thuật tự động vô hiệu hóa Credential cũ:
    - Công cụ: Script Python (Boto3).
    - Luồng xử lý (Workflow):
        1. Khởi tạo: Script gọi API generate_credential_report để yêu cầu AWS tạo báo cáo tổng thể.
        2. Chờ & Tải: Script chờ trạng thái báo cáo chuyển sang COMPLETE, sau đó tải về và giải mã (Decode) định dạng CSV.
        3. Phân tích: Duyệt qua từng dòng (từng User). Với mỗi User, kiểm tra trường access_key_last_used_date.
        4. Ra quyết định: Tính toán: Delta = (Ngày hiện tại) – (Ngày sử dụng cuối).
        5. Khắc phục: Nếu Delta > 45 ngày, thực thi API iam.update_access_key(Status='Inactive'). Hành động này khóa key ngay lập tức nhưng không xóa, đảm bảo khả năng khôi phục nếu cần.

2.2.2 Lưu trữ & Cơ sở dữ liệu (Storage & Database Security):

- Tự động hóa Mã hóa Dữ liệu:
    - Với RDS: Module Terraform được thiết kế để tham số storage_encrypted luôn nhận giá trị True và tự động liên kết với KMS Key ID (aws_kms_key). Điều này đảm bảo không một Database nào được tạo ra mà thiếu mã hóa.
    - Với EBS: Sử dụng lệnh CLI hoặc Terraform để bật cờ EBS Encryption by Default ở cấp độ Region. Cơ chế này hoạt động như một "Global Policy", tự động mã hóa mọi ổ đĩa được tạo ra sau đó, bất kể người dùng có chọn mã hóa hay không.

- Cơ chế "Khóa" truy cập công khai S3:
    - Phương pháp: Sử dụng tính năng S3 Block Public Access ở cấp độ Tài khoản (Account Level).
    - Chi tiết: Terraform kích hoạt resource aws_s3_account_public_access_block với 4 cờ bảo mật (BlockPublicAcls, IgnorePublicAcls, BlockPublicPolicy, RestrictPublicBuckets) đều là True. Đây là lớp bảo vệ bao trùm ngay cả khi một Admin cố tình public một bucket cụ thể, cấu hình cấp tài khoản này sẽ ghi đè và ngăn chặn hành động đó.

2.2.3. Ghi nhật ký & Giám sát (Logging & Monitoring):

- Hệ thống Giám sát hướng sự kiện:
    - Kiến trúc tự động: Thay vì tạo thủ công từng cảnh báo, hệ thống sử dụng Terraform để xây dựng một chuỗi liên kết tự động: CloudTrail (Ghi log) -> CloudWatch Logs (Chứa log) -> Metric Filter (Lọc từ khóa) -> CloudWatch Alarm (Cảnh báo) -> SNS Topic (Gửi mail).
    - Cơ chế lọc: Metric Filter sử dụng các mẫu (Pattern Syntax) của CloudWatch để "nghe" các sự kiện rủi ro trong thời gian thực. Ví dụ, mẫu { $.userIdentity.type = "Root" ... } sẽ kích hoạt cảnh báo ngay khi có bất kỳ API call nào từ tài khoản Root.
- Đánh giá tuân thủ liên tục:
    - Phương pháp: Kích hoạt AWS Security Hub và cơ chế "Auto-enable Controls".
    - Chi tiết: Security Hub được cấu hình để tự động đăng ký gói tiêu chuẩn CIS AWS Foundations Benchmark. Dịch vụ này sẽ chạy ngầm định kỳ (mỗi 12-24 giờ) để quét toàn bộ tài nguyên và trả về điểm số tuân thủ (Compliance Score) mà không cần can thiệp thủ công.

2.2.4. An ninh Mạng (Network Security):

- Giải thuật tự động đóng các cổng quản trị rủi ro:

- Công cụ: Script Python (Boto3).
- Luồng xử lý (Workflow):
    1. Quét: Script gọi API describe_security_groups để lấy danh sách toàn bộ nhóm bảo mật trong VPC.
    2. Lọc vi phạm: Với mỗi nhóm, duyệt qua danh sách IpPermissions (Inbound Rules). Tìm các rule thỏa mãn điều kiện logic: (Port == 22 OR Port == 3389) AND (CidrIp == '0.0.0.0/0').
    3. Khắc phục: Nếu tìm thấy, script lấy GroupId và thực thi API revoke_security_group_ingress.
    4. Kết quả: Cổng quản trị bị đóng ngay lập tức đối với truy cập toàn cầu, đưa Security Group về trạng thái an toàn.

- Tự động hóa bảo vệ Metadata:
    - Phương pháp: Cập nhật hàng loạt (Batch Remediation).
    - Chi tiết: Script sẽ quét tất cả EC2 Instances đang chạy. Nếu phát hiện instance đang sử dụng IMDSv1 (hoặc HttpTokens là optional), script sẽ thực thi lệnh modify_instance_metadata_options để chuyển sang HttpTokens=required, ép buộc sử dụng IMDSv2 (phiên bản an toàn chống lỗ hổng SSRF).

2.2.5 Công cụ & Framework:

Phân loại	Công cụ/Framework	Mô tả sử dụng
Local Tools	CIS-CAT Pro Assessor	Đánh giá tuân thủ benchmark CIS tự động, xuất báo cáo CSV/HTML.
Local Tools	AWS CLI	Thực thi lệnh kiểm tra (MFA, IAM keys, RDS encryption).
Local Tools	Python (boto3)	Viết script tự động kiểm tra và khắc phục cấu hình IAM, S3, CloudTrail.
Remote Tools	AWS Config	Giám sát tuân thủ cấu hình trên cloud.
Remote Tools	AWS Security Hub	Tổng hợp và đối chiếu các vi phạm CIS benchmark toàn hệ thống.

Remote Tools	AWS Trusted Advisor	Phát hiện lỗ hổng bảo mật và đề xuất cải thiện tự động.
Automation Frameworks	Terraform / Ansible	Quản lý hạ tầng theo chuẩn IaC, đảm bảo tuân thủ tự động.
CI/CD Integration	GitHub Actions / Jenkins / AWS CodePipeline	Tự động chạy kiểm tra CIS mỗi khi cập nhật cấu hình AWS.

❖ **Môi trường triển khai**

– Local: chạy script Python từ máy quản trị (Ubuntu/Windows).

– Remote: sử dụng AWS Config + Security Hub để giám sát tự động trên cloud.(optional)

❖ **Đầu ra (Output)**

– Định dạng: .csv, .json, .html (tùy tool).

– Tần suất: theo lịch cron hoặc CI/CD pipeline.

– Mục đích: lưu trữ, so sánh mức độ tuân thủ định kỳ.

❖ **Ví dụ quy trình tự động hóa**

1. Python (boto3) xác minh trạng thái MFA, quyền IAM, cấu hình S3.
2. Kết quả gửi về AWS Security Hub → tạo cảnh báo CloudWatch.
3. Pipeline CI/CD tự động khắc phục bằng Terraform hoặc Ansible.

3. MÔ HÌNH VÀ TRIỂN KHAI

3.1 MÔ HÌNH:

Lớp 1: Nền tảng và Cung ứng (Provisioning Layer):

- Mục tiêu: Đảm bảo mọi tài nguyên được tạo ra ngay từ đầu đã tuân thủ CIS. Đây là lớp phòng thủ chủ động.
- Công cụ chính: Terraform.

Dịch vụ AWS (Triển khai bằng Terraform)	Mục tiêu CIS
AWS Organization & IAM	- Tạo vai trò (roles) thay vì user. - Áp dụng chính sách mật khẩu mạnh (aws_iam_account_password_policy).

VPC & Networking	- Tạo Security Group (aws_security_group) không cho phép cổng 22/3389 từ 0.0.0.0/0. - Cấu hình VPC Flow Logs (aws_flow_log).
Storage (S3, EBS, RDS)	- Bật mã hóa EBS mặc định (aws_ebs_encryption_by_default). - Chặn S3 Public Access (aws_s3_bucket_public_access_block). - Mã hóa RDS (storage_encrypted = true trong aws_db_instance).
Dịch vụ Giám sát (Lớp 2, 4)	- Kích hoạt CloudTrail đa vùng (is_multi_region_trail = true) và bật xác thực tệp log. - Kích hoạt AWS Config đa vùng (aws_config_configuration_recorder). - Kích hoạt Security Hub (aws_securityhub_account) và bật chuẩn CIS.
CI/CD Pipeline	- (Sử dụng GitHub Actions/CodePipeline) - Mục tiêu: Quản lý việc chạy terraform apply và lưu terraform.tfstate an toàn trên S3.

Lớp 2: Phát hiện (Detection Layer)

- Mục tiêu: Giám sát môi trường 24/7 để phát hiện các thay đổi (drift) hoặc cấu hình thủ công vi phạm CIS.
- Công cụ chính: Dịch vụ AWS (được triển khai bằng Terraform ở Lớp 1).

Dịch vụ AWS	Vai trò trong mô hình
AWS Config	- Hoạt động như "máy quay" ghi lại mọi thay đổi cấu hình. - Sử dụng các Config Rules để liên tục đánh giá tài nguyên.
AWS Security Hub	- Trung tâm điều khiển: Tổng hợp phát hiện từ Config, GuardDuty, Macie. - Quan trọng: Tự động đánh giá toàn bộ tài khoản dựa trên chuẩn CIS Benchmark. Nó sẽ tạo ra "Findings" (Phát hiện) cho các vi phạm.
CloudWatch Events	- (Hiện là Amazon EventBridge) - Lắng nghe các "Findings" mới từ Security Hub hoặc các thay đổi từ Config.

Lớp 3: Khắc phục Tự động (Remediation Layer):

Mục tiêu: Tự động sửa chữa các vi phạm mà Lớp 2 phát hiện.

Công cụ chính: Python (Boto3) trên AWS Lambda, được điều phối bởi EventBridge.

Dịch vụ AWS	Vai trò trong mô hình
Amazon EventBridge	- Bộ điều phối: Nhận sự kiện từ Security Hub (ví dụ: CIS 4.3 - Public SSH access detected). - Lọc và gửi sự kiện đến đúng hàm Lambda khắc phục.

AWS Lambda	- Người thực thi: Chứa các script Python (Boto3) của bạn. - Mỗi hàm Lambda được thiết kế để khắc phục một vi phạm cụ thể. - Ví dụ 1 (Từ báo cáo): Lambda nhận sự kiện "Credential không dùng > 45 ngày" và chạy script Boto3 để vô hiệu hóa key. - Ví dụ 2 (Từ báo cáo): Lambda nhận sự kiện "Cổng 22 mở công khai" và chạy ec2.revoke_security_group_ingress để đóng cổng đó.
IAM Roles	- Cung cấp quyền (theo nguyên tắc đặc quyền tối thiểu) để Lambda có thể sửa đổi các tài nguyên (ví dụ: sửa Security Group, cập nhật IAM).

Lớp 4: Ghi nhật ký & Cảnh báo (Logging & Alerting Layer)

Mục tiêu: Ghi lại mọi thứ để kiểm toán (audit) và thông báo cho con người về các sự kiện quan trọng.

Công cụ chính: CloudTrail, CloudWatch, SNS (Tất cả được triển khai bằng Terraform).

Dịch vụ AWS	Vai trò trong mô hình
CloudTrail	- Ghi lại mọi lệnh gọi API (ai đã làm gì, khi nào). - Bật xác thực tệp log (chống giả mạo). - Mã hóa log bằng KMS.
CloudWatch	- Giám sát: Thu thập log (từ CloudTrail, VPC Flow Logs, Lambda). - Cảnh báo (Alarms): Thiết lập các Metric Filter và Alarms cho các sự kiện cực kỳ quan trọng (như trong báo cáo: sử dụng tài khoản root, thay đổi IAM policy, đăng nhập thất bại...) .
Amazon SNS	- Thông báo: Gửi cảnh báo từ CloudWatch (ví dụ: "Phát hiện đăng nhập Root!") hoặc thông báo từ Lambda (ví dụ: "Đã tự động khóa cổng 22 tại SG-...") đến email/Slack của đội bảo mật.
S3	- Nơi lưu trữ an toàn và lâu dài cho tất cả các log (CloudTrail, VPC Flow, S3 Access Logs).

3.2 TRIỂN KHAI MANUAL:

3.2.1. Quản lý Danh tính và Truy cập (IAM):

a. Cập nhật thông tin liên hệ an ninh

- Mục tiêu: Đảm bảo AWS có thể liên hệ với đúng bộ phận (Security Team) khi phát hiện sự cố, thay vì chỉ liên hệ với chủ sở hữu tài khoản.
- Các bước thực hiện:
    1. Đăng nhập vào AWS Console với quyền Root hoặc IAM User có quyền Billing.
    2. Chọn tên tài khoản ở góc trên bên phải > chọn Account.

3. Kéo xuống phần Alternate Contacts.
4. Tại mục Security contact, nhấn Edit.
5. Điền thông tin:
    - Full name: Nhập tên đội bảo mật (ví dụ: Cloud Security Team).
    - Title: Chức danh (ví dụ: Security Lead).
    - Email address: Nhập email để nhận được thông báo.
    - Phone number: Số điện thoại khẩn cấp.
6. Nhấn Update.

b. Hạn chế sử dụng tài khoản Root cho tác vụ hàng ngày

- Mục tiêu: Bảo vệ tài khoản Root (quyền lực tuyệt đối) bằng cách không sử dụng nó cho các hoạt động quản trị thông thường.
- Các bước thực hiện:
    1. Đăng nhập bằng tài khoản Root lần cuối để tạo người dùng quản trị (Admin User).
    2. Truy cập IAM Console > Users > Create user.
    3. Tạo user mới (ví dụ: admin-user).
    4. Gán quyền: Chọn Attach policies directly > Chọn AdministratorAccess (Hoặc tốt hơn là gán vào nhóm Admins).
    5. Đăng xuất khỏi tài khoản Root.
    6. Từ nay về sau, chỉ đăng nhập bằng admin-user để quản trị hệ thống.
    7. (Tùy chọn) Thiết lập cảnh báo CloudWatch khi tài khoản Root được sử dụng (xem phần Giám sát).

3.2.2. Lưu trữ (Storage) – RDS:

a. Cấu hình RDS Multi-AZ để tăng tính sẵn sàng

- Mục tiêu: Đảm bảo cơ sở dữ liệu vẫn hoạt động nếu một Availability Zone (AZ) gặp sự cố. Đây là cấu hình Manual trong Level 1.
- Các bước thực hiện:
    1. Truy cập RDS Console > Databases.
    2. Chọn instance DB cần cấu hình > Nhấn Modify.
    3. Trong phần Availability & durability, tích chọn Create a standby instance (Multi-AZ deployment).
    4. Nhấn Continue.
    5. Tại mục Scheduling of modifications, chọn Apply immediately (nếu đang trong môi trường lab/test) hoặc chọn thời gian bảo trì (nếu production).
    6. Nhấn Modify DB Instance.

3.2.3. Ghi nhật ký (Logging) – CloudTrail:

a. Kích hoạt CloudTrail trên tất cả các vùng

- Mục tiêu: Đảm bảo mọi hành động API trên tài khoản đều được ghi lại, kể cả ở những Region không sử dụng (nơi hacker thường lợi dụng để ẩn mình). Đây là cấu hình Manual trong Level 1.
- Các bước thực hiện:
    1. Truy cập CloudTrail Console > Trails.
    2. Nhấn Create trail.
    3. Trail name: Đặt tên (ví dụ: management-events-trail).
    4. Storage location: Chọn Create new S3 bucket (để lưu log) và đặt tên bucket duy nhất.
    5. Bỏ chọn Log file SSE-KMS encryption (nếu muốn tiết kiệm chi phí lab, nhưng khuyến nghị bật cho production - CIS 4.5).
    6. Tích chọn Log file validation (để đáp ứng CIS 4.2).
    7. Nhấn Next.
    8. Tại phần Events, chọn Management events.
    9. Mục API activity, chọn cả Read và Write.
    10. Nhấn Next > Create trail.

b. Bật ghi log truy cập (Server Access Logging) cho S3 Bucket của CloudTrail (CIS 4.4)

- Mục tiêu: Giám sát xem ai đang cố gắng truy cập vào các file log bảo mật quan trọng. Đây là cấu hình Manual trong Level 1.
- Các bước thực hiện:
    1. Truy cập S3 Console.
    2. Tìm và chọn Bucket vừa được tạo bởi CloudTrail ở bước trên.
    3. Chuyển sang tab Properties.
    4. Cuộn xuống mục Server access logging, nhấn Edit.
    5. Chọn Enable.
    6. Target bucket: Chọn một bucket S3 khác (bucket dùng riêng để chứa access logs).
    7. Nhấn Save changes.

3.2.4. Giám sát (Monitoring) - CloudWatch Alarms:

Lưu ý: CIS Benchmark đánh dấu các mục giám sát (5.1 - 5.15) là Manual vì quy trình kiểm toán và khắc phục thường yêu cầu thiết lập thủ công các bộ lọc (Metric Filters).

a. Tạo cảnh báo khi tài khoản Root được sử dụng (CIS 5.3)

- Mục tiêu: Nhận email cảnh báo ngay lập tức nếu có ai đó đăng nhập hoặc dùng quyền Root.
- Các bước thực hiện:
    1. Tạo Metric Filter:
        - Truy cập CloudWatch Console > Logs > Log groups.
        - Chọn Log group của CloudTrail (thường tên là aws-cloudtrail-logs-...).
        - Chọn tab Metric filters > Create metric filter.
        - Filter pattern: Nhập đoạn mã sau:

          { $.userIdentity.type = "Root" && $.userIdentity.invokedBy NOT EXISTS && $.eventType != "AwsServiceEvent" }
        - Nhấn Next.
        - Metric name: RootUsageMetric.
        - Metric namespace: CISBenchmark.
        - Metric value: 1.
        - Nhấn Create metric filter.
    2. Tạo Alarm:
        - Tại màn hình Metric filters vừa tạo, tích chọn metric RootUsageMetric và nhấn Create alarm.
        - Statistic: Sum. Period: 5 minutes (hoặc thấp hơn để test nhanh).
        - Threshold type: Static.
        - Condition: Greater/Equal (>=) 1.
        - Nhấn Next.
        - Notification: Chọn Create new topic (ví dụ: SecurityAlerts), nhập email của bạn vào ô Email endpoints.
        - Nhấn Create topic (đừng quên vào email xác nhận subscription).
        - Nhấn Next > Đặt tên Alarm (ví dụ: RootAccountUsageAlarm) > Create alarm.

b. Tạo cảnh báo cho các lần đăng nhập thất bại (CIS 5.6 - Level 2)

- Mục tiêu: Phát hiện tấn công dò mật khẩu (Brute-force) vào AWS Console.
- Các bước thực hiện:
    1. Tương tự như trên, vào Log group của CloudTrail > Create metric filter.
    2. Filter pattern:{ ($.eventName = ConsoleLogin) && ($.errorMessage = "Failed authentication") }

3. Metric details: Tên ConsoleLoginFailureMetric, namespace CISBenchmark, giá trị 1.
4. Tạo Alarm tương tự: Ngưỡng >= 1, gửi thông báo về SNS Topic SecurityAlerts.

3.2.5. Mạng (Networking):

a. Đảm bảo bảng định tuyến (Route Table) cho VPC Peering là "Least Access" (CIS 6.6 - Level 2)

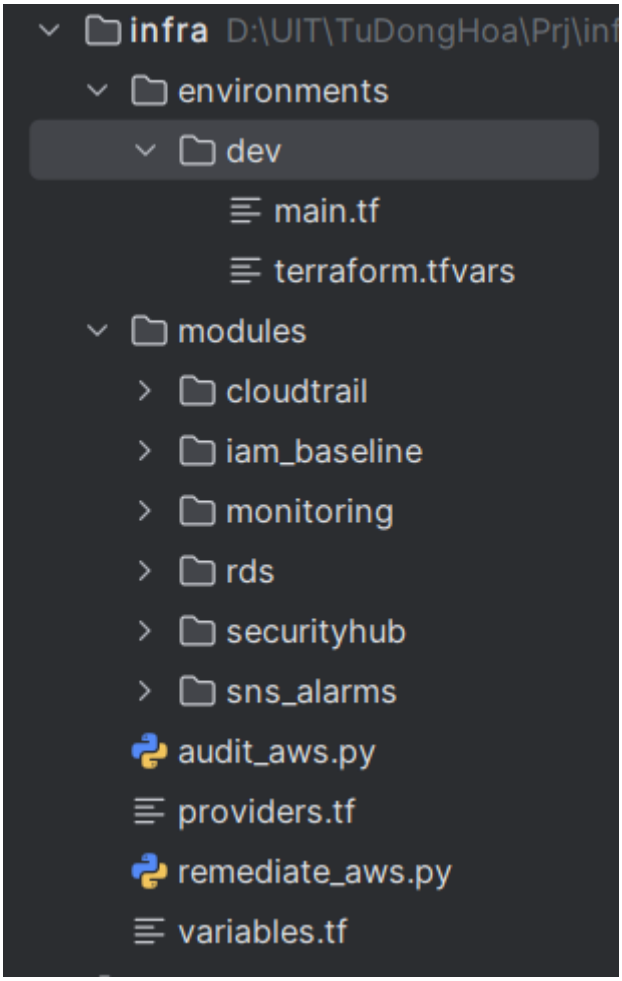
- Mục tiêu: Khi nối hai mạng VPC, chỉ cho phép đi đến đúng subnet cần thiết, không mở toang toàn bộ mạng. Đây là cấu hình Manual.
- Các bước thực hiện:
    1. Truy cập VPC Console > Route Tables.
    2. Chọn Route Table đang liên kết với VPC Peering.
    3. Chọn tab Routes > Edit routes.
    4. Kiểm tra các dòng có Target là pcx-xxxxxx (Peering Connection).
    5. Tại cột Destination, thay vì để 10.0.0.0/16 (toàn bộ VPC kia), hãy sửa thành dải IP cụ thể của server cần kết nối (ví dụ: 10.0.1.5/32 hoặc 10.0.1.0/24).
    6. Nhấn Save changes.

3.3 TRIỂN KHAI AUTOMATION:

3.3.1. Giai đoạn 1: Thiết lập nền tảng tuân thủ bằng Terraform (Preventive):

Mục tiêu: Sử dụng Terraform để đảm bảo các tài nguyên ngay khi được tạo ra đã tuân thủ các tiêu chuẩn CIS.

a. Cấu trúc thư mục dự án: Dự án được tổ chức theo mô hình module hóa để dễ quản lý và tái sử dụng:




```
infra/
        modules/              # Thư viện các module tái sử dụng
        iam_baseline/          # Quản lý IAM & Password Policy
        rds_secure/           # Database RDS mã hóa & an toàn
        cloudtrail/           # Hệ thống ghi log tập trung
        monitoring/           # Giám sát & Cảnh báo (CloudWatch)
        networking/           # VPC, Security Groups, Flow Logs
        environments/         # Cấu hình môi trường triển khai
            dev/              # Môi trường phát triển
                main.tf       # File chính gọi các module
                variables.tf  # Khai báo biến đầu vào
                terraform.tfvars  # Giá trị bí mật thực tế
```

b. Triển khai chính sách mật khẩu IAM : Thay vì cấu hình thủ công, Terraform sẽ cấu hình tự động vào tài khoản AWS: (modules/iam_baseline/main.tf)

```
resource "aws_iam_account_password_policy" "cis_compliant" {
  minimum_password_length        = 14
  require_lowercase_characters   = true
  require_numbers                = true
  require_uppercase_characters   = true
  require_symbols                = true
  allow_users_to_change_password = true
  password_reuse_prevention      = 24
  max_password_age               = 90
}
```

c. Triển khai hệ thống giám sát CloudWatch (CIS 5.x): Tự động tạo Metric Filter và Alarm để phát hiện các hành vi nguy hiểm (ví dụ: Thay đổi Network Gateway).

```
resource "aws_cloudwatch_log_metric_filter" "network_gateway_changes" {
  name           = "CIS-5.12-NetworkGatewayChanges"
  pattern        = "{ ($.eventName = CreateCustomerGateway) || ($.eventName =
DeleteCustomerGateway) || ($.eventName = AttachInternetGateway) ... }"
  log_group_name = var.cloudtrail_log_group_name

  metric_transformation {
    name      = "NetworkGatewayChanges"
    namespace = "CISBenchmark"
    value     = "1"
  }
}
```

3.3.2. Giai đoạn 2: Kiểm toán tự động bằng Python (Detective):

Mục tiêu: Thực hiện bằng scripts Python sử dụng thư viện boto3 để quét định kỳ và phát hiện các cấu hình sai lệch.

a. Kịch bản Audit (audit_aws.py): Script này hoạt động như một "Auditor" ảo, thực hiện các lệnh Describe và List để kiểm tra trạng thái thực tế so với chuẩn CIS.

Ví dụ logic kiểm tra CIS 2.3 (Root Access Keys):

```
import boto3
def audit_root_access_keys():
    iam = boto3.client('iam')
    summary = iam.get_account_summary()
    # Kiểm tra xem tài khoản Root có Access Key không
    if summary['SummaryMap']['AccountAccessKeysPresent'] == 1:
        return "Tài khoản Root đang có Access Key hoạt động."
    else:
        return "Tài khoản Root an toàn."
```

VVí dụ logic kiểm tra CIS 6.3 (Security Group mở Port 22):

```
def audit_ssh_port_open():
    ec2 = boto3.client('ec2')
    # Lọc các Security Group cho phép port 22 từ 0.0.0.0/0
    response = ec2.describe_security_groups(
        Filters=[
            {'Name': 'ip-permission.from-port', 'Values': ['22']},
            {'Name': 'ip-permission.cidr', 'Values': ['0.0.0.0/0']}
        ]
    )
    if len(response['SecurityGroups']) > 0:
        return f"[Phát hiện {len(response['SecurityGroups'])} nhóm bảo mật mở SSH public."
    return "[Không có nhóm bảo mật nào mở SSH public."
```

3.3.3. Giai đoạn 3: Khắc phục tự động (Remediation/Corrective):

Mục tiêu: Tự động sửa lỗi khi phát hiện vi phạm nghiêm trọng, giảm thời gian phản ứng (MTTR).

a. Kịch bản Remediation (remediate_aws.py): Script này có thể chạy độc lập hoặc tích hợp vào AWS Lambda để chạy khi có sự kiện Trigger từ CloudWatch.

Ví dụ logic khắc phục CIS 6.3 (Đóng Port 22 tự động):

```
def remediate_ssh_port(group_id):
    ec2 = boto3.client('ec2')
    print(f"Đang thực hiện đóng port 22 trên Group ID: {group_id}...")

    try:
        # Thu hồi quyền truy cập SSH từ 0.0.0.0/0
        ec2.revoke_security_group_ingress(
            GroupId=group_id,
            IpPermissions=[
                {'IpProtocol': 'tcp', 'FromPort': 22, 'ToPort': 22, 'IpRanges': [{'CidrIp': '0.0.0.0/0'}]}
            ]
        )
        print("--> Đã khắc phục thành công. Port 22 đã đóng.")
    except Exception as e:
        print(f"--> Lỗi khi khắc phục: {str(e)}")
```

3.3.4. Quy trình vận hành thực tế:

Quy trình được thực hiện theo vòng lặp khép kín:

1. Build: Chạy terraform apply để triển khai hạ tầng sạch.
2. Scan: Chạy python **audit_aws.py** định kỳ (cronjob hoặc CI/CD) để xuất báo cáo tình trạng.
3. Alert: Nếu script Audit phát hiện vi phạm -> Gửi cảnh báo qua SNS/Email.
4. Fix: Quản trị viên chạy python **remediate_aws.py** để đưa hệ thống về trạng thái an toàn.

TÀI LIỆU THAM KHẢO