AWS Well-Architected Framework

보안 원칙

보안 원칙: AWS Well-Architected Framework

Table of Contents

보안 원칙 - AWS Well-Architected Framework

게시 날짜: 2023년 12월 6일 (문서 개정)

이 백서에서는 다음을 구성하는 보안 원칙에 중점을 둡니다. AWS Well-Architected Framework단축할 수 있습니다. 안전한 AWS 워크로드 설계, 제공 및 유지 관리에 모범 사례 및 현재 권장 사항을 적용할 때 참조할 수 있는 지침을 제공합니다.

소개

유효 AWS Well-Architected Framework 는 AWS에서 워크로드를 구축할 때 내리는 의사 결정의 상충 관계를 이해하는 데 도움이 됩니다. 이 프레임워크를 사용하면 클라우드에서 안정적이고 안전하며 효율적이고 경제적이면서 지속 가능한 워크로드를 설계하고 운영하기 위한 최신 설계 모범 사례를 알아볼 수 있습니다. 이 프레임워크는 모범 사례를 기준으로 워크로드를 일관적으로 측정하고 개선 영역을 식별하는 방법을 제공합니다. 워크로드를 제대로 설계하면 비즈니스 성공 가능성이 높아집니다.

이 프레임워크는 다음 6가지 원칙을 기반으로 합니다.

- 운영 우수성
- 보안
- 신뢰성
- 성능 효율성
- 비용 최적화
- 지속 가능성

이 백서는 보안 원칙을 중점적으로 다룹니다. 최신 AWS 권장 사항에 따라 비즈니스 및 규제 요구 사항을 충족하는 데 도움이 됩니다. 이 문서는 최고 기술 책임자(CTO), 최고 정보 보안 책임자(CSO/CISO), 아키텍트, 개발자와 같은 기술 업무 담당자와 운영 팀원을 위해 작성되었습니다.

이 백서의 내용을 확인하고 나면 보안을 염두에 두고 클라우드 아키텍처를 설계할 때 사용할 AWS의 현재 권장 사항과 전략을 파악할 수 있습니다. 구현 세부 정보나 아키텍처 패턴은 제공되지 않지만, 이 정보를 확인할 수 있는 적절한 리소스에 대한 참조는 포함되어 있습니다. 이 백서에서 설명하는 사례를 도입하면 데이터와 시스템을 보호하고, 액세스를 제어하고, 보안 이벤트에 자동으로 대응하는 아키텍처를 구축할 수 있습니다.

보안 기초

보안 원칙은 클라우드 기술을 활용하여 보안 상태를 개선함으로써 데이터, 시스템 및 자산을 보호하는 방법을 설명합니다. 이 백서에서는 AWS에서 보안 워크로드를 설계하기 위한 심층 모범 사례 지침을 제공합니다.

설계 원칙

클라우드에는 워크로드 보안을 강화할 수 있는 여러 가지 원칙이 존재합니다.

- 강력한 자격 증명 기반 구현: 권한을 최소화한 보안 주체를 구현하고 AWS 리소스와의 각 상호 작용에 대한 적절한 권한을 부여하여 업무를 분리합니다. 자격 증명 관리를 중앙 집중화하고 장기적인 정적 자격 증명에 대한 의존도를 해소하는 것을 목표로 합니다.
- 추적성 유지: 실시간으로 환경에 대한 작업 및 변경 사항을 모니터링하고 알림을 전송하며 감사합니다. 로그 및 지표 수집을 시스템과 통합하여 자동으로 조사하고 조치를 취합니다.
- 모든 계층에 보안 적용: 여러 보안 제어 기능을 통해 심층 방어 방식을 적용합니다. 모든 계층(예: 네트워크 엣지, VPC, 로드 밸런싱, 모든 인스턴스 및 컴퓨팅 서비스, 운영 체제, 애플리케이션, 코드)에 적용됩니다.
- 보안 모범 사례의 자동 적용: 자동화된 소프트웨어 기반의 보안 메커니즘은 더욱 빠르게 안전한 확장 능력을 향상 시켜주며 비용 효율적입니다. 버전 제어가 가능한 템플릿에서 코드로 정의되고 관리되는 제어 기능의 구현을 비롯한 보안 아키텍처를 생성합니다.
- 전송 및 보관 중인 데이터 보호: 데이터를 민감도 수준으로 분류하고 적절한 경우 암호화, 토큰화 및 액세스 제어와 같은 메커니즘을 사용합니다.
- 사람들이 데이터에 쉽게 액세스할 수 없도록 유지: 데이터의 직접 액세스 또는 수동 처리의 필요성을 줄이거나 없애기 위한 메커니즘 및 도구를 사용합니다. 이를 통해 민감한 데이터를 처리할 때 잘못된 취급이나 수정 및 수작업으로 인한 오류의 위험을 줄일 수 있습니다.
- 보안 이벤트에 대비: 조직의 요구 사항에 부합하는 인시던트 관리 및 조사 정책과 프로세스를 마련하여 인시던트에 대비합니다. 인시던트 대응 시뮬레이션을 실행하고 자동화된 도구를 사용하여 감지, 조사 및 복구 속도를 높입니다.

정의

클라우드의 보안은 다음의 7가지 영역으로 구성됩니다.

- 보안 기초

- 자격 증명 및 액세스 관리
- 탐지
- 인프라 보호
- 데이터 보호
- 사고 대응
- 애플리케이션 보안

공동 책임

보안 및 규정 준수는 AWS와 고객의 공동 책임입니다. 이 공동 모델은 고객의 운영 부담을 더는 데 도움이 될 수 있습니다. 호스트 운영 체제 및 가상화 계층부터 서비스 운영 시설의 물리적 보안에 이르는 구성 요소를 AWS에서 운영, 관리 및 제어하기 때문입니다. 고객의 책임 및 관리 범위에는 AWS가 제공하는 보안 그룹 방화벽의 구성과 게스트 운영 체제(업데이트 및 보안 패치 포함) 및 기타 관련 애플리케이션 소프트웨어가 포함됩니다. 사용하는 서비스, 서비스를 IT 환경에 통합하는 과정 및 준거법과 규제에 따라 책임 범위가 다르기 때문에 고객은 선택하고자 하는 서비스에 대해 신중해야 합니다. 또한 이러한 공동 책임은 본질적으로 유연성을 제공하며 배포를 허용하는 제어 권한을 고객에게 부여합니다. 다음 차트에서 볼 수 있듯이 이 책임의 차이를 일반적으로 클라우드 '자체'의 보안과 클라우드 '내부'의 보안이라고 합니다.

AWS의 책임: '클라우드 자체의 보안' - AWS는 AWS 클라우드에서 제공하는 모든 서비스가 실행되는 인프라를 보호할 책임이 있습니다. 이 인프라는 AWS 클라우드 서비스를 실행하는 하드웨어, 소프트웨어, 네트워킹 및 시설로 구성됩니다.

고객의 책임: '클라우드 내부의 보안' - 고객의 책임은 고객이 선택하는 AWS 클라우드 서비스에 따라 결정됩니다. 서비스에 따라 고객이 보안 책임의 일환으로서 수행해야 하는 구성 작업의 양이 달라집니다. 예를 들어, Amazon Elastic Compute Cloud(Amazon EC2)와 같은 서비스는 서비스형 인프라(IaaS)로 분류되므로 고객이 필수 보안 구성 및 관리 작업을 모두 수행해야 합니다. Amazon EC2 인스턴스를 배포하는 고객의 경우 게스트 운영 체제 관리(업데이트 및 보안 패치 포함), 고객이 인스턴스에 설치하는 모든 애플리케이션 소프트웨어 또는 유틸리티, 각 인스턴스에 AWS에서 제공하는 방화벽 구성(보안 그룹이라고 함)에 대한 책임이 있습니다. Amazon S3 및 Amazon DynamoDB와 같은 추상화된 서비스의 경우 AWS는 인프라 계층, 운영 체제, 플랫폼을 작동하고, 고객은 엔드포인트에 액세스하여 데이터를 저장 및 검색합니다. 고객은 데이터를 관리하고(암호화 옵션 포함), 자산을 분류하고, IAM 도구를 사용하여 적절한 권한을 적용할 책임이 있습니다.

그림 1: AWS:공동 책임 모델

이 고객/AWS 공동 책임 모델은 IT 제어에까지 확대 적용됩니다. IT 환경을 운영하는 책임을 AWS와 고객이 공유하는 것과 마찬가지로, IT 제어의 관리, 운영, 확인 또한 공유됩니다. AWS는 이전에는 고객이 관리했던 AWS 환경에 배포된 물리적 인프라와 관련한 제어를 관리함으로써 운영 제어의 고객 부담을 완화하는 데 도움을 줄 수 있습니다. 고객마다 AWS에서 배포된 방법이 다르므로 고객은 특정 IT 제어 관리를 AWS로 전환하여 새로운 분산 제어 환경을 이용할 수 있습니다. 그런 다음 고객은 제공된 AWS 제어 및 규정 준수 설명서를 참조하여 필요한 제어 평가 및 검증 절차를 실시할 수 있습니다. 다음은 AWS, AWS 고객 또는 AWS와 AWS 고객이 공동으로 관리하는 제어의 예시입니다.

상속된 제어 - 고객이 AWS로부터 완전히 상속하는 제어입니다.

- 물리적 및 환경 제어

공동 제어 - 별개의 컨텍스트 또는 관점에서 인프라 계층 및 고객 계층 모두에 적용되는 제어입니다. 공동 제어에서 AWS는 인프라에 대한 요구 사항을 제공하고 고객은 AWS 서비스 사용과 관련한 자체적인 제어 구현을 제공해야 합니다. 다음을 예로 들 수 있습니다.

- 패치 관리 - AWS는 인프라 내의 패치 작업과 결함 수정에 대한 책임이 있지만, 고객은 게스트 운영 체제와 애플리케이션 패치 작업에 책임이 있습니다.
- 구성 관리 - AWS는 자체 인프라 디바이스의 구성을 유지 관리하지만, 고객은 자체 게스트 운영 체제, 데이터베이스, 애플리케이션 구성에 책임이 있습니다.

- 인식 및 교육 - AWS는 AWS 직원을 교육시키며 고객은 자체 직원을 교육시켜야 합니다.

고객별 - AWS 서비스 내에서 배포하는 애플리케이션에 따라 고객에게 전적인 책임이 있는 제어입니다. 다음을 예로 들 수 있습니다.

- 고객이 특정 보안 환경 내에서 데이터를 라우팅하거나 위치시켜야 하는 서비스 및 통신 보호 또는 영역 보안입니다.

거버넌스

전반적인 접근 방식의 일부로 보안 거버넌스를 포함하는 것은 위험 관리에 도움이 되는 정책과 제어 목표를 정의하여 비즈니스 목표 달성을 지원하기 위한 것입니다. 보안 제어 목표에 계층화된 접근법을 사용하여 위험을 관리하세요. 각 계층은 이전 계층을 기반으로 구현합니다. AWS 공동 책임 모델을 이해하는 것이 가장 기본적인 계층입니다. 이 모델을 이해하면 고객 측에서 어떤 부분에 책임이 있는지, AWS로부터 어떤 책임을 상속하는지 명확히 알 수 있습니다. 유용한 리소스인 AWS Artifact를사용하면 AWS의 보안 및 규정 준수 보고서에 온디맨드로 액세스하고 온라인 계약을 선택할 수 있습니다.

다음 계층에서 대부분의 제어 목표를 달성하세요. 플랫폼 전반의 기능이 이 계층에 있습니다. 예를 들어, 이 계층에는 AWS 계정 벤딩 프로세스, AWS IAM Identity Center과 같은 자격 증명 공급자와의 통합, 일반적인 탐지 제어가 포함됩니다. 플랫폼 거버넌스 프로세스의 결과 중 일부도 이 계층에 있습니다. 새로운 AWS 서비스를 사용하고자 할 때 AWS Organizations 서비스에서 서비스 제어 정책(SCP)을 업데이트하여 서비스 최초 사용의 가드레일을 제공할 수 있습니다. 다른 SCP를 사용하여 '보안 불변수'라고 부르는 일반적인 보안 제어 목표를 구현할 수 있습니다. 보안 불변수는 여러 개의 계정, 조직 단위 또는 전체 AWS 조직에 적용하는 제어 목표 또는 구성입니다. 일반적으로 인프라가 실행되는 리전의 수를 제한하거나 탐지 제어 비활성화를 차단하는 것을 예로 들 수 있습니다. 이 중간 계층에는 구성 규칙 또는 파이프라인 검사 등 코드화된 정책도 포함됩니다.

상단 계층은 제품 팀과 제어 목표가 만나는 부분입니다. 제품 팀이 제어하는 애플리케이션에서 구현이 수행되기 때문입니다. 애플리케이션에 입력 검증을 구현하거나 마이크로서비스 간에 자격 증명이 정확히 전달되도록 하는 것을 예로 들 수 있습니다. 제품 팀에서 구성을 담당하지만, 여전히 중간 계층으로부터 일부 기능을 상속할 수 있습니다.

제어를 어디에 구현하든 목표는 같습니다. 바로 위험을 관리하는 것입니다. 위험 관리 프레임워크의 범위는 구체적인 업계, 리전 또는 기술에 적용됩니다. 주요 목표는 발생 가능성과 결과에 따라 위험을 강조하는 것입니다. 이것을 내재된 위험이라고 합니다. 그런 다음 발생 가능성, 결과 또는 둘 다를 축소하는 제어 목표를 정의할 수 있습니다. 그러고 나서 제어 조치가 마련된 상태에서 위험의 결과가 무엇일지 볼 수 있습니다. 이것을 잔존 위험이라고 합니다. 제어 목표는 하나의 워크로드나 여러 개의 워크로

드에 적용할 수 있습니다. 다음 다이어그램은 일반적인 위험 매트릭스를 보여줍니다. 발생 가능성은 이전 발생 빈도를 기반으로 하며 결과는 이벤트로 인한 재정, 평판, 시간 비용을 기반으로 합니다.

가능성	위험 수준				
매우 높음	낮음	보통	높음	심각	심각
높음	낮음	보통	보통	높음	심각
보통	낮음	낮음	보통	보통	높음
낮음	낮음	낮음	보통	보통	높음
매우 낮음	낮음	낮음		보통	높음
결과	거의 없음	낮음	보통	높음	극심함

그림 2: 위험 수준 발생 가능성 매트릭스

AWS 계정 관리 및 분리

조직의 보고 구조를 미러링하는 대신 기능, 규정 준수 요구 사항 또는 공통 제어 요소를 기반으로 별도의 계정과 그룹 계정에 워크로드를 구성하는 것이 좋습니다. AWS에서 계정은 뚜렷한 경계를 가지고 있습니다. 예를 들어 프로덕션 워크로드를 개발 및 테스트 워크로드에서 격리하려면 계정 수준의 분리를 적극 권장합니다.

중앙에서 계정 관리: AWS Organizations 는 AWS 계정 생성 및 관리,그리고 생성된 계정에 대한 제어를 자동화합니다. AWS Organizations를 통해 계정을 생성할 때는 사용할 이메일 주소를 신중히 선택해야 합니다. 이 이메일 주소가 암호를 재설정할 수 있는 루트 사용자가 되기 때문입니다. Organizations를 사용하면 여러 계정을 OU(조직 단위)로 그룹화하여 워크로드의 요구 사항과 용도에 따라 서로 다른 환경을 나타내도록 할 수 있습니다.

중앙에서 제어 설정: 적절한 수준의 특정 서비스, 리전 및 서비스 작업만 허용하여 AWS 계정에서 수행할 수 있는 작업을 제어하세요. AWS Organizations에서는 서비스 제어 정책(SCP)을 사용하여 모든 조직에 적용되는 조직, 조직 단위 또는 계정 수준에서 권한 가드레일을 적용할 수 있습니다. AWS Identity and Access Management (IAM) 사용자와 역할에 적용할 수 있습니다. 예를 들어 사용자가 명

시적으로 허용하지 않은 리전에서는 리소스를 시작하지 못하도록 제한하는 SCP를 적용할 수 있습니다. AWS Control Tower는 여러 계정을 간편하게 설정하고 관리하는 방법을 제공합니다. 이는 AWS Organizations에서 계정 설정 및 프로비저닝을 자동화하고 가드레일 (예방 및 탐지 포함)을 적용하며 대시보드를 제공하여 가시성을 확보해줍니다.

중앙에서 서비스 및 리소스 구성: AWS Organizations를 사용하면 모든 계정에 적용되는 AWS 서비스 를 구성할 수 있습니다. 예를 들어 AWS CloudTrail, 회원 계정이 로깅을 비활성화하는 것을 방지합니다. 또한 정의한 규칙에 대해 AWS Config를 사용하여 중앙에서 데이터를 취합할 수 있으므로 워크로드를 감사하여 규정 준수 여부를 확인하고, 변화에 신속하게 대응하도록 지원할 수 있습니다. AWS CloudFormation StackSets 를 사용하면 조직 내의 여러 계정 및 OU에 걸쳐 AWS CloudFormation 스택을 중앙에서 관리할 수 있습니다. 이렇게 하면 보안 요구 사항에 부합하도록 새 계정을 자동으로 프로비저닝할 수 있습니다.

보안 서비스의 위임된 관리 기능을 사용하여 조직의 결제(관리) 계정에서 관리에 사용되는 계정을 분리하세요. GuardDuty, Security Hub, AWS Config 등의 여러 AWS 서비스에서는 관리 기능을 위해 특정 계정을 지정하는 등 AWS Organizations와의 통합을 지원합니다.

모범 사례

- SEC01-BP01 계정을 사용하여 워크로드 분리
- SEC01-BP02 계정 루트 사용자 및 속성 보호

SEC01-BP01 계정을 사용하여 워크로드 분리

다중 계정 전략을 통해 환경(예: 프로덕션, 개발 및 테스트)과 워크로드 간에 일반적인 가드레일 및 격리를 설정합니다. 계정 수준의 분리는 보안, 청구 및 액세스에 대한 강력한 격리 경계를 제공하므로 강력하게 권장됩니다.

원하는 결과: 클라우드 운영, 관련 없는 워크로드 및 환경을 별도의 계정으로 격리하여 클라우드 인프라 전반의 보안을 강화하는 계정 구조입니다.

일반적인 안티 패턴:

- 데이터 민감도 수준이 서로 다른 관련 없는 여러 워크로드를 동일한 계정에 배치합니다.
- 잘못 정의된 조직 단위(OU) 구조입니다.

이 모범 사례 확립의 이점:

- 워크로드가 의도치 않게 액세스되는 경우 영향 범위 감소

- AWS 서비스, 리소스 및 리전에 대한 액세스 권한의 중앙 거버넌스
- 보안 서비스의 정책 및 중앙 집중식 관리를 통해 클라우드 인프라의 보안 유지 관리
- 자동화된 계정 생성 및 유지 관리 프로세스
- 규정 준수 및 규제 요구 사항에 대한 인프라의 중앙 집중식 감사

이 모범 사례를 따르지 않을 경우 노출 위험도: 높음

구현 가이드

AWS 계정는 서로 다른 민감도 수준에서 작동하는 워크로드 또는 리소스 간에 보안 격리 경계를 제공합니다. AWS는 다중 계정 전략을 통해 대규모로 클라우드 워크로드를 관리할 수 있는 도구를 제공하여 이 격리 경계를 활용할 수 있습니다. AWS에 대한 다중 계정 전략의 개념, 패턴 및 구현에 대한 지침은 여러 계정을 사용하여 AWS 환경 구성을 참조하세요.

중앙 관리 하에 여러 AWS 계정가 있는 경우 조직 단위(OU) 계층으로 정의된 계층 구조로 계정을 구성해야 합니다. 그런 다음 보안 제어를 구성하고 OU 및 멤버 계정에 적용하여 조직의 멤버 계정에 일관된 예방적인 제어를 설정할 수 있습니다. 보안 제어는 상속되므로 OU 계층 구조의 하위 수준에 있는 멤버 계정이 사용 가능한 권한을 필터링할 수 있습니다. 우수한 설계는 이 상속을 활용하여 각 멤버 계정에 대해 원하는 보안 제어를 달성하는 데 필요한 보안 정책의 수와 복잡성을 줄입니다.

AWS Organizations 및 AWS Control Tower는 AWS 환경에서 이 다중 계정 구조를 구현하고 관리하는 데 사용할 수 있는 두 가지 서비스입니다. AWS Organizations를 사용하면 하나 이상의 OU 계층으로 정의된 계층 구조로 계정을 구성할 수 있습니다. 각 OU에는 여러 멤버 계정이 포함되어 있습니다. 서비스 제어 정책(SCP)을 사용하면 조직 관리자가 멤버 계정에 대한 세분화된 예방 제어를 설정할 수 있으며, AWS Config를 사용하면 멤버 계정에 대한 사전 예방 및 탐지 제어를 설정할 수 있습니다. 많은 AWS 서비스가 AWS Organizations와 통합되어 위임된 관리 제어를 제공하고 조직의 모든 멤버 계정 전체의 서비스별 작업을 수행합니다.

AWS Organizations를 기반의 AWS Control Tower는 랜딩 존이 있는 다중 계정 AWS 환경에 대한 원클릭 모범 사례 설정을 제공합니다. 랜딩 존은 Control Tower가 구축한 다중 계정 환경의 진입점입니다. Control Tower는 AWS Organizations에 비해 몇 가지 이점을 제공합니다. 개선된 계정 거버넌스를 제공하는 세 가지 이점은 다음과 같습니다.

- 조직에 참여하도록 승인된 계정에 자동으로 적용되는 통합 필수 보안 가드레일
- 주어진 OU 세트에 대해 활성화 또는 비활성화할 수 있는 선택적 가드레일
- AWS Control Tower Account Factory는 조직 내에서 사전 승인된 기준 및 구성 옵션이 포함된 계정의 자동 배포를 제공합니다.

구현 단계

1. 조직 단위 구조 설계: 적절하게 설계된 조직 단위 구조는 서비스 제어 정책 및 기타 보안 제어를 생성하고 유지 관리하는 데 필요한 관리 부담을 줄여줍니다. 조직 단위 구조는 비즈니스 요구 사항, 데이터 민감도 및 워크로드 구조에 맞춰 조정해야 합니다.
2. 다중 계정 환경을 위한 랜딩 존 생성: 랜딩 존은 조직이 워크로드를 신속하게 개발, 실행 및 배포할 수 있는 일관된 보안 및 인프라 기반을 제공합니다. 맞춤형으로 구축된 랜딩 존 또는 AWS Control Tower를 사용하여 환경을 오케스트레이션할 수 있습니다.
3. 가드레일 설정: 랜딩 존을 통해 환경에 일관된 보안 가드레일을 구현합니다. AWS Control Tower는 배포할 수 있는 필수 및 선택적 제어 목록을 제공합니다. 필수 제어는 Control Tower를 구현할 때 자동으로 배포됩니다. 적극 권장되는 선택적 제어 목록을 검토하고 요구 사항에 적합한 제어를 구현합니다.
4. 새로 추가된 리전에 대한 액세스 권한 제한: 새 AWS 리전의 경우 사용자 및 역할과 같은 IAM 리소스는 사용자가 지정하는 리전으로만 전파됩니다. 이 작업은 Control Tower를 사용할 때 콘솔을 통해 수행하거나 AWS Organizations에서 IAM 권한 정책을 조정하여 수행할 수 있습니다.
5. AWS CloudFormation StackSets 고려: StackSets를 사용하면 IAM 정책, 역할, 그룹을 포함한 리소스를 승인된 템플릿에서 다양한 AWS 계정과 리전에 배포할 수 있습니다.

리소스

관련 모범 사례:

- SEC02-BP04 중앙 집중식 자격 증명 공급자 사용

관련 문서:

- AWS Control Tower
- AWS 보안 감사 지침
- IAM 모범 사례
- CloudFormation StackSets를 사용하여 여러 AWS 계정 및 리전에 리소스 프로비저닝
- Organizations FAQ
- AWS Organizations 용어 및 개념
- AWS Organizations 다중 계정 환경의 서비스 제어 정책 모범 사례
- AWS 계정 관리 참조 가이드

- 여러 계정을 사용하여 AWS 환경 구성

관련 동영상:

- Enable AWS adoption at scale with automation and governance(자동화 및 거버넌스를 통해 대규모 AWS 도입 지원)
- Security Best Practices the Well-Architected Way(Well-Architected 방식의 보안 모범 사례)
- Building and Governing Multiple Accounts using AWS Control Tower(AWS Control Tower를 사용하여 여러 계정 구축 및 관리)
- Enable Control Tower for Existing Organizations(기존 조직에 Control Tower 활성화)

관련 워크숍:

- Control Tower Immersion Day

SEC01-BP02 계정 루트 사용자 및 속성 보호

루트 사용자는 계정 내의 모든 리소스에 대한 전체 관리 액세스 권한이 있는 AWS 계정에서 가장 권한이 높은 사용자이며 경우에 따라 보안 정책의 제약을 받을 수 없습니다. 루트 사용자에 대한 프로그래밍 방식 액세스 권한을 비활성화하고, 루트 사용자에 대한 적절한 제어를 설정하며, 루트 사용자의 일상적인 사용을 방지하면 루트 보안 인증 정보가 의도치 않게 노출되어 클라우드 환경이 손상될 위험을 줄일 수 있습니다.

원하는 결과: 루트 사용자를 보호하면 루트 사용자 보안 인증 정보의 남용으로 인해 우발적이거나 의도적인 손상이 발생할 가능성을 줄일 수 있습니다. 탐지 제어를 설정하면 루트 사용자를 사용하여 작업을 수행할 때 적절한 담당자에게 알릴 수도 있습니다.

일반적인 안티 패턴:

- 루트 사용자 보안 인증 정보가 필요한 몇 가지 작업 이외의 작업에 루트 사용자를 사용합니다.
- 긴급 상황에서의 중요한 인프라, 프로세스 및 인력 기능을 확인하기 위한 비상 계획을 정기적으로 테스트하는 데 소홀합니다.
- 일반적인 계정 로그인 흐름만 고려하고 대체 계정 복구 방법을 고려하거나 테스트하는 데 소홀합니다.
- DNS, 이메일 서버 및 전화 공급자가 계정 복구 흐름에서 사용되므로 중요한 보안 경계의 일부로 처리하지 않습니다.

이 모범 사례 확립의 이점: 루트 사용자에 대한 액세스를 보호하면 계정의 작업이 제어되고 감사된다는 확신을 얻을 수 있습니다.

이 모범 사례를 따르지 않을 경우 노출 위험도: 높음

구현 가이드

AWS는 계정을 보호하는 데 도움이 되는 다양한 도구를 제공합니다. 그러나 이러한 조치는 대부분 기본적으로 활성화되어 있지 않으므로 이를 구현하기 위해서는 직접적인 조치를 취해야 합니다. AWS 계정 보호를 위한 기본 단계로 다음 권장 사항을 고려해 보세요. 이러한 단계를 구현할 때 보안 제어를 지속적으로 평가하고 모니터링하는 프로세스를 구축하는 것이 중요합니다.

AWS 계정를 처음 생성할 때 계정의 모든 AWS 서비스 및 리소스에 대한 완전한 액세스 권한이 있는 단일 자격 증명을 생성하는 것으로 시작합니다. 이때 생성하는 자격 증명을 AWS 계정 루트 사용자라고 합니다. 계정을 생성할 때 사용한 이메일 주소와 암호를 사용하여 루트 사용자로 로그인할 수 있습니다. AWS 루트 사용자에게 부여된 높은 액세스 권한으로 인해 특별히 이를 필요로 하는 작업을 수행하려면 AWS 루트 사용자의 사용을 제한해야 합니다. 루트 사용자 로그인 보안 인증 정보는 철저하게 보호되어야 하며 AWS 계정 루트 사용자에 대해 항상 다중 인증(MFA)을 활성화해야 합니다.

사용자 이름, 암호 및 다중 인증(MFA) 디바이스를 사용하여 루트 사용자에 로그인하는 일반적인 인증 흐름 외에도 이메일 주소 및 계정과 연결된 전화번호에 대한 액세스 권한이 부여된 AWS 계정 루트 사용자에 로그인하는 계정 복구 흐름이 있습니다. 따라서 복구 이메일이 전송되는 루트 사용자 이메일 계정을 보호하는 것과 계정과 연결된 전화번호를 보호하는 것이 동일하게 중요합니다. 또한 루트 사용자와 연결된 이메일 주소가 동일한 AWS 계정의 이메일 서버 또는 도메인 이름 서비스(DNS) 리소스에서 호스팅되는 잠재적인 순환 종속성도 고려하세요.

AWS Organizations를 사용하는 경우 각각 루트 사용자가 있는 여러 AWS 계정가 있습니다. 하나의 계정이 관리 계정으로 지정되면 여러 계층의 멤버 계정을 관리 계정 아래 추가할 수 있습니다. 관리 계정의 루트 사용자 보호에 우선순위를 지정한 다음 멤버 계정 루트 사용자의 주소를 기재합니다. 관리 계정의 루트 사용자를 보호하기 위한 전략은 멤버 계정 루트 사용자와 다를 수 있으며, 멤버 계정 루트 사용자에게 예방 보안 제어를 적용할 수 있습니다.

구현 단계

루트 사용자에 대한 제어를 설정하려면 다음 구현 단계를 수행하는 것이 좋습니다. 해당하는 경우 권장 사항은 CIS AWS Foundations 벤치마크 버전 1.4.0과 상호 참조됩니다. 이러한 단계 외에도 AWS 계정 및 리소스 보호에 대한 AWS 모범 사례 지침을 참조하세요.

예방 제어

1. 계정에 대한 정확한 연락처 정보를 설정합니다.
    a. 이 정보는 분실한 암호 복구 흐름, 분실한 MFA 디바이스 계정 복구 흐름 및 팀과의 중요한 보안 관련 커뮤니케이션에 사용됩니다.
    b. 회사 도메인에서 호스팅하는 이메일 주소(배포 목록 선호)를 루트 사용자의 이메일 주소로 사용합니다. 개인의 이메일 계정이 아닌 배포 목록을 사용하면 장기간에 걸쳐 루트 계정에 액세스할 수 있는 추가 중복성과 연속성이 제공됩니다.
    c. 연락처 정보에 표시된 전화번호는 이 목적을 위한 보안 전용 전화여야 합니다. 전화번호를 표시하거나 다름 사람과 공유해서는 안 됩니다.
2. 루트 사용자의 액세스 키를 생성하지 않습니다. 액세스 키가 있으면 제거합니다(CIS 1.4).
    a. 루트 사용자에 대한 수명이 긴 프로그래밍 방식 보안 인증 정보(액세스 및 보안 암호 키)를 제거합니다.
    b. 루트 사용자 액세스 키가 이미 있는 경우 해당 키를 사용하여 AWS Identity and Access Management(IAM) 역할에서 임시 액세스 키를 사용하도록 프로세스를 전환한 다음 루트 사용자 액세스 키를 삭제해야 합니다.
3. 루트 사용자의 보안 인증 정보를 저장해야 하는지 여부를 결정합니다.
    a. AWS Organizations를 사용하여 새 멤버 계정을 생성하는 경우 새 멤버 계정에 대한 루트 사용자의 초기 암호가 사용자에게 노출되지 않는 임의의 값으로 설정됩니다. 필요한 경우 AWS 조직 관리 계정의 암호 재설정 흐름을 사용하여 멤버 계정에 대한 액세스 권한을 얻는 것이 좋습니다.
    b. 독립 실행형 AWS 계정 또는 관리 AWS 조직 계정의 경우 루트 사용자에 대한 보안 인증 정보를 생성하고 안전하게 저장하는 것이 좋습니다. 루트 사용자에 대해 MFA를 활성화합니다.
4. AWS 다중 계정 환경에서 멤버 계정 루트 사용자에 대한 예방 제어를 활성화합니다.
    a. 멤버 계정에 대해 루트 사용자의 루트 액세스 키 생성 차단 예방 가드레일을 활성화하는 것이 좋습니다.
    b. 멤버 계정에 대해 루트 사용자로서의 작업 차단 예방 가드레일을 활성화하는 것이 좋습니다.
5. 루트 사용자에 대한 보안 인증 정보가 필요한 경우:
    a. 복잡한 암호를 사용합니다.
    b. 루트 사용자, 특히 AWS Organizations 관리(지급인) 계정에 대해 다중 인증(MFA)을 활성화합니다(CIS 1.5).
    c. 단일 사용 디바이스는 MFA 코드가 포함된 디바이스가 다른 용도로 재사용될 가능성을 줄일 수 있으므로 복원력 및 보안을 위해 하드웨어 MFA 디바이스를 고려합니다. 배터리로 구동되는 하드웨어 MFA 디바이스가 정기적으로 대체되는지 확인합니다. (CIS 1.6)
        - 루트 사용자에 대해 MFA를 구성하려면 가상 MFA 또는 하드웨어 MFA 디바이스를 활성화하는 지침을 따르세요.

d. 백업을 위해 여러 MFA 디바이스를 등록하는 것이 좋습니다. 계정당 최대 8개의 MFA 디바이스가 허용됩니다.
   - 루트 사용자에 대해 둘 이상의 MFA 디바이스를 등록하면 MFA 디바이스가 손실된 경우 계정을 복구하는 흐름이 자동으로 비활성화됩니다.

e. 암호를 안전하게 저장하고, 암호를 전자적으로 저장하는 경우 순환 종속성을 고려합니다. 암호를 획득하는 데 동일한 AWS 계정에 대한 액세스 권한이 필요한 방식으로 암호를 저장하지 않습니다.

6. 선택 사항: 루트 사용자에 대한 주기적인 암호 교체 일정을 설정하는 것이 좋습니다.
   - 보안 인증 정보 관리 모범 사례는 규정 및 정책 요구 사항에 따라 다릅니다. MFA로 보호되는 루트 사용자는 단일 인증 요소로 암호에 의존하지 않습니다.
   - 주기적으로 루트 사용자 암호를 변경하면 의도치 않게 노출된 암호가 남용될 위험이 줄어듭니다.

탐지 제어

- 루트 보안 인증 정보의 사용을 감지하는 경보를 생성합니다(CIS 1.7). Amazon GuardDuty를 활성화하면 RootCredentialUsage 결과를 통해 루트 사용자 API 보안 인증 정보 사용을 모니터링하고 알립니다.
- AWS Config용 AWS Well-Architected 보안 원칙 규정 준수 팩에 포함된 탐지 제어를 평가 및 구현하거나 AWS Control Tower를 사용하는 경우 Control Tower 내에서 사용 가능한 강력하게 권장되는 제어를 평가 및 구현합니다.

운영 지침

- 조직에서 루트 사용자 보안 인증 정보에 대한 액세스 권한을 갖는 담당자를 결정합니다.
  - 루트 사용자 액세스 권한을 획득하는 데 필요한 모든 보안 인증 정보 및 MFA에 대한 액세스 권한을 한 명의 개인이 갖지 않도록 2인 규칙을 사용합니다.
  - 한 명의 개인이 아닌 조직에서 계정과 연결된 전화번호 및 이메일 별칭(암호 재설정 및 MFA 재설정 흐름에 사용됨)에 대한 제어 권한을 유지 관리하는지 확인합니다.
- 루트 사용자는 예외적으로만 사용합니다(CIS 1.7).
  - AWS 루트 사용자는 관리 작업이라 하더라도 일상 작업에는 사용하면 안 됩니다. 루트 사용자가 필요한 AWS 작업을 수행하려면 루트 사용자로만 로그인합니다.. 다른 모든 작업은 적절한 역할이 있는 다른 사용자가 수행해야 합니다.
- 루트 사용자 보안 인증 정보를 사용해야 하는 긴급 상황이 발생하기 전에 절차를 테스트할 수 있도록 루트 사용자에 대한 액세스 권한이 작동하는지 주기적으로 확인합니다.

- 계정과 연결된 이메일 주소와 대체 연락처에 나열된 이메일 주소가 작동하는지 주기적으로 확인합니다. 다음에서 수신된 보안 알림이 있는지 이러한 이메일 받은 편지함을 <abuse@amazon.com>모니터링합니다. 또한 계정과 연결된 모든 전화번호가 작동하는지 확인합니다.
- 루트 계정 남용에 대응하기 위한 인시던트 대응 절차를 준비합니다. AWS 계정에 대한 인시던트 대응 전략을 구축하는 방법에 대한 자세한 내용은 AWS 보안 인시던트 대응 가이드 및 보안 원칙 백서의 인시던트 대응 섹션에 있는 모범 사례를 참조하세요.

리소스

관련 모범 사례:

- SEC01-BP01 계정을 사용하여 워크로드 분리
- SEC02-BP01 강력한 로그인 메커니즘 사용
- SEC03-BP02 최소 권한 액세스 부여
- SEC03-BP03 긴급 액세스 프로세스 설정
- SEC10-BP05 액세스 권한 사전 프로비저닝

관련 문서:

- AWS Control Tower
- AWS 보안 감사 지침
- IAM 모범 사례
- Amazon GuardDuty – 루트 보안 인증 정보 사용 알림
- CloudTrail을 통한 루트 보안 인증 정보 사용 모니터링에 대한 단계별 지침
- AWS에서 사용하도록 승인된 MFA 토큰
- AWS에서 브레이크 글라스 액세스 구현
- AWS 계정에서 개선해야 할 10가지 보안 항목
- AWS 계정에서 무단 활동이 발견되면 어떻게 해야 하나요?

관련 동영상:

- Enable AWS adoption at scale with automation and governance(자동화 및 거버넌스를 통해 대규모 AWS 도입 지원)
- Security Best Practices the Well-Architected Way(Well-Architected 방식의 보안 모범 사례)

- Limiting use of AWS root credentials from AWS re:inforce 2022 – Security best practices with AWS IAM(AWS re:inforce 2022 – AWS IAM의 보안 모범 사례의 AWS 루트 보안 인증 정보 사용 제한)

관련 예시 및 실습:

- 실습: AWS 계정 and root user(AWS 계정 및 루트 사용자)

워크로드를 안전하게 운영

안전한 워크로드 운영에는 설계에서 빌드, 실행, 지속적 개선까지 워크로드의 전체 수명 주기가 포함됩니다. 클라우드에서 안전하게 운영하는 역량을 향상하는 한 가지 방법은 거버넌스에 조직적 접근법을 취하는 것입니다. 거버넌스는 관여된 사람의 선의의 판단에 전적으로 의존하지 않고 일관되게 의사 결정을 이끄는 방식입니다. 거버넌스 모델과 프로세스는 '특정 워크로드에 대한 제어 목표가 충족되며 제어 목표가 해당 워크로드에 적합한지 어떻게 알 수 있는가?'라는 질문에 대답을 제시합니다. 의사 결정에 일관적인 접근법을 마련하면 워크로드 배포의 속도가 빨라지고 조직 내 보안 역량의 기준을 높이는 데 도움이 됩니다.

워크로드를 안전하게 운영하려면 모든 보안 영역에 포괄적 모범 사례를 적용해야 합니다. 조직 및 워크로드 수준에서 운영 우수성에 정의된 요구 사항과 프로세스를 가져와 모든 영역에 적용합니다. AWS 및 업계 권장 사항 및 위협 인텔리전스를 최신 상태로 유지하면 위협 모델 및 제어 목표를 발전시키는 데 도움이 됩니다. 보안 프로세스, 테스트 및 검증을 자동화하면 보안 작업을 확장하는 데 도움이 됩니다.

자동화를 통해 프로세스의 일관성과 반복 가능성을 확보할 수 있습니다. 사람들은 많은 것에 두각을 나타내지만 실수 없이 같은 일을 일관적으로 반복하는 것은 사람이 잘하는 일이 아닙니다. 런북이 잘 작성되었더라도 사람들이 반복적인 작업을 일관적으로 수행하지 않을 것이라는 위험이 있습니다. 사람들의 책임이 서로 다르며 익숙하지 않은 알림에 대응해야 할 때는 그 위험이 특히 더 큽니다. 하지만 자동화는 매번 같은 방식으로 반응합니다. 애플리케이션을 배포하는 가장 좋은 방법은 자동화를 사용하는 것입니다. 배포를 실행하는 코드를 테스트한 후 배포를 수행하는 데 사용할 수 있습니다. 그러면 변경 프로세스에 대한 신뢰도가 높아지고 실패한 변경에 대한 위험이 낮아집니다.

구성이 제어 목표에 부합하는지 확인하기 위해 자동화와 배포된 애플리케이션을 비프로덕션 환경에서 먼저 테스트하세요. 그러면 자동화가 모든 단계를 올바르게 수행했는지 입증할 수 있습니다. 또한 개발 및 배포 주기의 초기에 피드백을 얻어 재작업을 줄일 수 있습니다. 배포 오류의 가능성을 줄이려면 사람이 아닌 코드를 사용하여 구성을 변경하세요. 애플리케이션을 다시 배포해야 한다면 자동화를 사용하는 경우 배포가 훨씬 쉽습니다. 추가 제어 목표를 정의할 때 모든 워크로드에 적용되도록 자동화에 쉽게 추가하면 됩니다.

개별 워크로드 담당자가 워크로드별 보안을 위해 노력하게 하는 대신 공동의 기능과 공유 구성 요소를 사용하여 시간을 절약하세요. 여러 팀이 사용할 수 있는 서비스의 예로는 AWS 계정 생성 프로세스, 인력을 위한 중앙 집중화된 자격 증명, 일반적인 로깅 구성, AMI 및 컨테이너 기반 이미지 생성이 있습니다. 이 접근법을 사용하면 빌더가 워크로드 주기 시간을 개선하고 일관적으로 보안 제어 목표를 달성하는 데 도움이 됩니다. 팀의 일관성이 높아지면 제어 목표를 검증하고 이해 관계자에게 제어 태세와 위험 포지션을 더 효과적으로 보고할 수 있습니다.

모범 사례

- SEC01-BP03 제어 목표 파악 및 검증
- SEC01-BP04 최신 보안 위협 정보 파악
- SEC01-BP05 최신 보안 권장 사항 파악
- SEC01-BP06 파이프라인에서 보안 제어의 테스트 및 검증 자동화
- SEC01-BP07 위협 모델을 사용하여 위협 식별 및 완화 조치의 우선순위 지정
- SEC01-BP08 새로운 보안 서비스 및 기능을 정기적으로 평가 및 구현

SEC01-BP03 제어 목표 파악 및 검증

위협 모델에서 식별된 규정 준수 요구 사항 및 위험을 기준으로, 워크로드에 적용해야 하는 제어 목표와 제어 항목을 도출하고 검증합니다. 제어 목표 및 제어에 대한 지속적인 검증은 위험 완화의 효과를 측정하는 데 도움이 됩니다.

이 모범 사례가 수립되지 않을 경우 노출되는 위험의 수준: 높음

구현 가이드

- 규정 준수 요구 사항 파악: 워크로드가 준수해야 하는 조직/법적/규정 준수 요구 사항을 파악합니다.
- AWS 규정 준수 리소스 파악: 규정 준수를 지원하기 위해 AWS에서 제공하는 리소스를 파악합니다.
  - https://aws.amazon.com/compliance/
  - https://aws.amazon.com/artifact/

리소스

관련 문서:

- AWS 보안 감사 지침

- 보안 공지

관련 동영상:

- AWS Security Hub: 보안 알림 관리 및 규정 준수 자동화
- Well-Architected 방식의 보안 모범 사례

SEC01-BP04 최신 보안 위협 정보 파악

적절한 제어 수단을 정의하고 구현하는 데 도움이 되도록 최신 보안 위협 정보를 바탕으로 공격 벡터를 파악합니다. AWS Managed Services를 사용하면 AWS 계정에서 예기치 않거나 일반적이지 않은 동작에 대한 알림을 더 쉽게 받을 수 있습니다. 보안 정보 흐름의 일부로 AWS 파트너 도구 또는 서드 파티 위협 정보 피드를 사용하여 조사합니다. 이 일반적인 취약점 및 노출(CVE) 목록 에는 최신 정보를 얻기 위해 사용할 수 있는 공개된 사이버 보안 취약점이 수록되어 있습니다.

이 모범 사례가 수립되지 않을 경우 노출되는 위험의 수준: 높음

구현 가이드

- 위협 정보 출처 구독: 워크로드에 사용된 기술과 관련이 있는 여러 출처의 위협 정보를 정기적으로 검토합니다.
  - 일반적인 취약점 및 노출 목록
- 실시간으로 워크로드를 모니터링하고, 보안 문제를 식별하고, 근본 원인 분석 및 수정을 신속하게 처리할 수 있도록 AWS Shield Advanced 서비스: 인터넷을 통해 워크로드에 액세스할 수 있는 경우 인텔리전스 소스에 대한 실시간에 가까운 가시성을 제공합니다.

리소스

관련 문서:

- AWS 보안 감사 지침
- AWS Shield
- 보안 공지

관련 동영상:

- Well-Architected 방식의 보안 모범 사례

SEC01-BP05 최신 보안 권장 사항 파악

워크로드의 보안 태세를 강화하기 위해 최신 AWS 및 업계 보안 권장 사항을 모두 파악합니다. AWS 보안 공지 에는 보안 및 개인정보 알림에 대한 중요한 정보가 포함되어 있습니다.

이 모범 사례를 정립하지 않을 경우 노출되는 위험의 수준: 높음

구현 가이드

- AWS 업데이트 팔로우: 새로운 권장 사항, 팁 및 요령을 구독하거나 정기적으로 확인합니다.
  - AWS Well-Architected 실습
  - AWS 보안 블로그
  - AWS 서비스 설명서
- 업계 뉴스 구독: 워크로드에 사용된 기술과 관련이 있는 여러 출처의 뉴스 피드를 정기적으로 검토합니다.
  - 예시: 일반적인 취약점 및 노출 목록

리소스

관련 문서:

- 보안 공지

관련 동영상:

- Well-Architected 방식의 보안 모범 사례

SEC01-BP06 파이프라인에서 보안 제어의 테스트 및 검증 자동화

빌드, 파이프라인 및 프로세스의 일부로서 테스트 및 검증되는 보안 메커니즘에 대한 보안 기준과 템플릿을 설정합니다. 도구와 자동화를 사용하여 모든 보안 제어를 지속적으로 테스트하고 검증합니다. 예를 들어 시스템 이미지와 인프라 같은 항목을 코드 템플릿으로 삼아 단계마다 설정된 기준에 따라 보안 취약성, 불규칙성, 드리프트를 검사합니다. AWS CloudFormation Guard는 CloudFormation 템플릿이 안전하고 시간을 절약해 주며 구성 오류의 위험을 줄여주는지 확인하는 데 도움이 됩니다.

프로덕션 환경에 적용되는 잘못된 보안 구성의 수를 줄여야 하므로, 빌드 프로세스에서 품질 관리와 결함 감소 과정을 많이 수행할수록 좋습니다. 가능한 경우 항상 보안 문제를 테스트하도록 지속적 통합

및 지속적 배포(CI/CD) 파이프라인을 설계합니다. CI/CD 파이프라인은 빌드 및 전달의 각 단계에서 보안을 강화할 기회를 제공합니다. 새로운 위협을 완화하기 위해 CI/CD 보안 도구도 업데이트해야 합니다.

워크로드 구성의 변경 사항을 추적하여 규정 준수 감사, 관리 변경, 적용 가능한 조사에 도움을 받습니다. AWS Config를 사용하여 AWS 및 서드 파티 리소스를 기록하고 평가할 수 있습니다. 이를 통해 전체적인 규정 준수를 규칙 및 규정 준수 팩(해결 조치가 포함된 규칙의 모음)으로 지속적으로 감사 및 평가할 수 있습니다.

변경 추적에는 조직의 변화 제어 프로세스에 포함된 정기적 변경 사항(MACD - 이동/추가/변경/삭제라고도 함), 계획하지 않은 변경 사항 또는 사고 등 예기치 않은 변경 사항도 포함됩니다. 인프라에서 변경이 이루어지기도 하지만 코드 리포지토리 변경, 머신 이미지 및 애플리케이션 인벤토리 변경, 프로세스 및 정책 변경 또는 문서 변경 등 다른 카테고리와 관련된 경우도 있을 수 있습니다.

이 모범 사례를 정립하지 않을 경우 노출되는 위험의 수준: 보통

구현 가이드

- 구성 관리 자동화: 구성 관리 서비스 또는 도구를 사용하여 보안 구성을 자동으로 적용하고 확보합니다.
  - AWS Systems Manager
  - AWS CloudFormation
  - AWS에서 CI/CD 파이프라인 설정

리소스

관련 문서:

- AWS Organization의 여러 계정에서 서비스 제어 정책을 사용해 권한 가드레일을 설정하는 방법

관련 동영상:

- AWS Organizations를 사용하여 다중 계정 AWS 환경 관리
- Well-Architected 방식의 보안 모범 사례

SEC01-BP07 위협 모델을 사용하여 위협 식별 및 완화 조치의 우선순위 지정

위협 모델링을 수행하여 워크로드에 대한 잠재적 위협 및 관련 완화 조치의 최신 등록을 식별하고 유지 관리합니다. 보안 위협 우선순위를 지정하고 보안 제어 완화 조치를 조정하여 방지, 감지 및 대응합니다. 워크로드와 진화하는 보안 환경에 맞춰 이를 보완하고 유지 관리합니다.

이 모범 사례를 따르지 않을 경우 노출 위험도: 높음

구현 가이드

위협 모델링이란 무엇인가요?

“위협 모델링은 가치 있는 것을 보호한다는 맥락에서 위협과 완화 조치를 식별, 전달 및 이해하기 위해 작동합니다.” – Open Web Application Security Project(OWASP) 애플리케이션 위협 모델링

위협 모델을 사용해야 하는 이유는 무엇인가요?

시스템은 복잡하고 시간이 지남에 따라 점점 더 복잡해지고 기능이 향상되어 더 많은 비즈니스 가치를 제공하고 고객 만족도와 참여도를 향상시킵니다. 즉, IT 설계를 결정할 때는 계속해서 증가하는 사용 사례를 고려해야 합니다. 이러한 복잡성과 사용 사례 순열의 수는 일반적으로 위협을 찾고 완화하는 데 구조화되지 않은 접근 방식을 비효율적으로 만듭니다. 대신 시스템에 대한 잠재적인 위협을 열거할 뿐만 아니라 조직의 제한된 리소스가 시스템의 전체 보안 태세를 개선하는 데 최대한 영향을 미칠 수 있도록 완화 조치를 고안하고 우선순위를 지정하는 체계적인 접근 방식이 필요합니다.

위협 모델링은 수명 주기 후반에 비해 상대적으로 완화 조치 관련 비용과 노력이 적게 필요한 설계 프로세스 초기에 문제를 찾아 해결하는 것을 목표로 이러한 체계적인 접근 방식을 제공하도록 설계되었습니다. 이 접근 방식은 shift-left 보안이라는 업계 원칙에 부합합니다. 궁극적으로 위협 모델링은 조직의 위험 관리 프로세스와 통합되며 위협 기반 접근 방식을 사용하여 구현할 제어에 대한 결정을 내리는 데 도움이 됩니다.

위협 모델링은 언제 수행해야 하나요?

워크로드의 수명 주기에서 가능한 한 빠르게 위협 모델링을 시작합니다. 그러면 식별한 위협에 대해 유연성을 높일 수 있습니다. 소프트웨어 버그와 마찬가지로 위협을 조기에 식별할수록 위협을 해결하는 것이 더 비용 효율적입니다. 위협 모델은 최신 상태를 유지해야 하는 문서로 워크로드가 변경됨에 따라 계속 진화해야 합니다. 주요 변경 사항이 있거나 위협 환경이 변경되거나 새로운 기능 또는 서비스를 채택하는 경우를 포함하여 시간이 지남에 따라 위협 모델을 보완합니다.

구현 단계

위협 모델링을 수행하려면 어떻게 해야 하나요?

위협 모델링을 수행하는 방법에는 여러 가지가 있습니다. 프로그래밍 언어와 마찬가지로 각각 장단점이 있으므로 자신에게 가장 적합한 방법을 선택해야 합니다. 한 가지 접근 방식은 위협 모델링 연습에 구조를 제공하기 위해 개방형 질문을 제시하는 위협 모델링에 대한 Shostack의 4가지 질문 프레임으로 시작하는 것입니다.

1. 어떤 작업을 하고 있나요?

   이 질문의 목적은 구축 중인 시스템과 보안과 관련된 해당 시스템에 대한 세부 정보를 이해하고 동의하는 데 도움을 주기 위한 것입니다. 모델이나 다이어그램을 생성하는 것은 데이터 흐름 다이어그램을 사용하여 구축 항목을 시각화하는 데 도움이 되므로 이 질문에 답하는 가장 일반적인 방법입니다. 시스템에 대한 권한 수임과 중요한 세부 정보를 작성하는 것도 범위를 정의하는 데 도움이 됩니다. 이를 통해 위협 모델에 기여하는 모든 담당자가 동일한 작업에 집중하고 범위 이외의 주제(시스템의 오래된 버전 포함)로 벗어나 시간을 허비하는 것을 방지할 수 있습니다. 예를 들어 웹 애플리케이션을 구축하는 경우, 사용자 설계를 통해 영향을 미칠 수 없기 때문에 브라우저 클라이언트에 대한 운영 체제의 신뢰할 수 있는 부팅 시퀀스에 대해 위협 모델링을 수행할 필요가 없습니다.

2. 잘못될 일이 뭐가 있을까요?

   이 질문을 통해 시스템에 대한 위협을 식별합니다. 위협은 원치 않는 영향을 미치고 시스템의 보안에 영향을 미칠 수 있는 우발적이거나 의도적인 작업 또는 이벤트입니다. 무엇이 잘못될 수 있는지에 대한 명확한 이해 없이는 위협에 대해 대응할 수 있는 방법이 아무 것도 없습니다.

   무엇이 잘못될 수 있는지에 대한 표준 목록은 없습니다. 이 목록을 작성하려면 팀 내의 모든 구성원과 위협 모델링 수행에 관여하는 관련 대상자 간의 브레인스토밍과 협업이 필요합니다. STRIDE와 같은 위협 식별 모델을 사용하여 브레인스토밍을 지원할 수 있습니다. 이 모델은 스푸핑, 변조, 거부, 정보 공개, 서비스 거부 및 권한 상승과 같이 평가할 다양한 범주를 제안합니다. 또한 OWASP Top 10, HiTrust Threat Catalog, 조직의 자체 위협 카탈로그를 포함하여 아이디어를 얻을 수 있는 기존 목록과 연구를 검토하여 브레인스토밍을 지원할 수 있습니다.

3. 이에 대해 무엇을 할 수 있나요?

   이전 질문의 경우와 마찬가지로 가능한 모든 완화 조치에 대한 표준 목록은 없습니다. 이 단계에 대한 입력은 이전 단계에서 식별된 위협, 행위자 및 개선 영역입니다.

   보안 및 규정 준수는 사용자와 AWS의 공동 책임입니다. '이에 대해 무엇을 할 수 있나요?'라고 질문할 때 '이에 대한 책임은 누구에게 있나요?'라고 질문하는 것임을 이해하는 것이 중요합니다. 사용자와 AWS 간의 책임 균형을 이해하면 위협 모델링 수행의 범위를 관리되는 완화 조치로 지정하는 데 도움이 됩니다. 이러한 완화 조치는 일반적으로 AWS 서비스 구성 옵션과 자체 시스템별 완화 조치의 조합으로 이루어집니다.

AWS 부분의 공동 책임에 대한 경우, AWS 서비스가 많은 규정 준수 프로그램의 범위에 속해 있다는 것을 알 수 있습니다. 이러한 프로그램을 통해 클라우드의 보안 및 규정 준수를 유지하기 위해 AWS에 마련된 강력한 제어 기능을 이해할 수 있습니다. 이러한 프로그램의 감사 보고서는 AWS 고객이 AWS Artifact에서 다운로드할 수 있습니다.

사용 중인 AWS 서비스에 관계없이 항상 고객 책임의 요소가 있으며 이러한 책임에 맞는 완화 조치가 위협 모델에 포함되어야 합니다. AWS 서비스 자체에 대한 보안 제어 완화 조치를 위해 자격 증명 및 액세스 관리(인증 및 권한 부여), 데이터 보호(저장 데이터 및 전송 중 데이터), 인프라 보안, 로깅, 모니터링과 같은 도메인을 포함한 도메인 전반에서 보안 제어 구현을 고려해야 합니다. 각 AWS 서비스에 대한 설명서에는 완화 조치로 고려할 보안 제어에 대한 지침을 제공하는 보안 전용 섹션이 있습니다. 작성 중인 코드와 해당 코드 종속성을 고려하고 이러한 위협을 해결하기 위해 마련할 수 있는 제어에 대해 생각하는 것이 중요합니다. 이러한 제어는 입력 검증, 세션 처리 및 경계 처리와 같은 것일 수 있습니다. 대부분의 취약성은 사용자 지정 코드에서 발생하는 경우가 많으므로 이 영역에 집중하세요.

4. 잘 수행했나요?

목표는 팀과 조직이 위협 모델의 품질과 시간이 지남에 따라 위협 모델링을 수행하는 속도를 모두 개선하는 것입니다. 이러한 개선은 연습, 학습, 지도, 검토의 조합에서 비롯됩니다. 더 자세히 알아보고 실습하려면 사용자와 팀이 Threat modeling the right way for builders(빌더에게 적합한 위협 모델링) 교육 과정 또는 워크숍을 완료하는 것이 좋습니다. 또한 위협 모델링을 조직의 애플리케이션 개발 수명 주기에 통합하는 방법에 대한 지침은 AWS 보안 블로그에서 How to approach threat modeling(위협 모델링 접근 방식) 게시물을 참조하세요.

Threat Composer

위협 모델링을 수행하는 데 도움과 안내를 받으려면 위협 모델링 시 가치 창출 시간을 단축하는 것을 목표로 하는 Threat Composer 도구를 사용하는 것이 좋습니다. 이 도구는 다음과 같은 작업을 수행하는 데 도움이 됩니다.

- 자연스러운 비선형 워크플로우에서 작동하는 threat grammar에 맞춰 유용한 위협 문을 작성합니다.
- 사람이 읽을 수 있는 위협 모델을 생성합니다.
- 기계가 읽을 수 있는 위협 모델을 생성하여 위협 모델을 코드로 취급할 수 있습니다.
- Insights Dashboard를 사용하여 품질 및 커버리지 개선 영역을 빠르게 식별할 수 있도록 도와줍니다.

자세한 내용을 보려면 Threat Composer를 방문하여 시스템 정의 Example Workspace로 전환하세요.

리소스

관련 모범 사례:

- SEC01-BP03 제어 목표 파악 및 검증
- SEC01-BP04 최신 보안 위협 정보 파악
- SEC01-BP05 최신 보안 권장 사항 파악
- SEC01-BP08 새로운 보안 서비스 및 기능을 정기적으로 평가 및 구현

관련 문서:

- 위협 모델링 접근 방식(AWS 보안 블로그)
- NIST: 데이터 중심 시스템 위협 모델링을 위한 가이드

관련 동영상:

- AWS Summit ANZ 2021 - How to approach threat modelling
- AWS Summit ANZ 2022 - Scaling security – Optimise for fast and secure delivery

관련 교육:

- Threat modeling the right way for builders(빌더에게 적합한 위협 모델링) – AWS Skill Builder 자습형 온라인 교육
- Threat modeling the right way for builders(빌더에게 적합한 위협 모델링) – AWS 워크숍

관련 도구:

- Threat Composer

SEC01-BP08 새로운 보안 서비스 및 기능을 정기적으로 평가 및 구현

워크로드의 보안 상태를 개선할 수 있는 AWS 및 AWS 파트너의 보안 서비스와 기능을 평가하고 구현합니다. AWS 보안 블로그에서는 새로운 AWS 서비스 및 기능, 구현 가이드, 일반적인 보안 가이드를 강조합니다. AWS의 새로운 소식은 모든 신규 AWS 기능, 서비스, 공지 사항을 파악하는 유용한 방법입니다.

이 모범 사례를 정립하지 않을 경우 노출되는 위험의 수준: 낮음

구현 가이드

- 일반 검토 계획: 규정 준수 요구 사항, 새 AWS 보안 기능/서비스 평가, 업계 최신 소식 확인 등의 검토 활동 일정을 생성합니다.
- AWS 서비스 및 기능 발견: 사용 중인 서비스에 적용 가능한 보안 기능을 검색하고 새로 릴리스되는 기능을 검토합니다.
  - AWS 보안 블로그
  - AWS 보안 공지
  - AWS 서비스 설명서
- AWS 서비스 온보딩 프로세스 정의: 새로운 AWS 서비스를 온보딩하는 프로세스를 정의합니다. 이 과정에서 새 AWS 서비스의 기능과 워크로드의 규정 준수 요구 사항을 평가할 방법도 정의합니다.
- 신규 서비스 및 기능 테스트: 프로덕션 환경을 거의 동일하게 복제한 비프로덕션 환경에서, 새로 릴리스하는 서비스 및 기능을 테스트합니다.
- 다른 방어 메커니즘 구현: 워크로드를 방어하기 위한 자동화된 메커니즘을 구현하고 사용 가능한 옵션을 살펴봅니다.
  - AWS Config 규칙에 따른 규정 미준수 AWS 리소스 문제 해결

리소스

관련 동영상:

- Well-Architected 방식의 보안 모범 사례

자격 증명 및 액세스 관리

AWS 서비스를 사용하려면 사용자와 애플리케이션에 AWS 계정 내 리소스에 대한 액세스 권한을 부여해야 합니다. AWS에서 더 많은 워크로드를 실행할 경우 적절한 사람이 적절한 조건에서 적절한 리소스에 액세스할 수 있도록 강력한 자격 증명 관리 및 권한이 필요합니다. AWS(은)는 인력 및 시스템 자격 증명과 그 권한을 관리하는 데 도움이 되는 다양한 기능을 제공합니다. 이러한 기능에 대한 모범 사례는 2가지 주요 영역으로 나뉩니다.

주제

- 자격 증명 관리
- 권한 관리

자격 증명 관리

안전한 AWS 워크로드 운영에 접근할 때, 관리해야 하는 2가지 유형의 자격 증명이 있습니다.

- 인적 자격 증명: 관리자, 개발자, 운영자 및 애플리케이션 소비자가 AWS 환경 및 애플리케이션에 액세스하려면 자격 증명이 필요합니다. 이들은 조직의 구성원이거나 협업하는 외부 사용자일 수 있으며, 웹 브라우저, 클라이언트 애플리케이션, 모바일 앱 또는 대화형 명령줄 도구를 통해 AWS 리소스와 상호 작용합니다.
- 시스템 자격 증명: 워크로드 애플리케이션, 운영 도구 및 구성 요소에서 AWS 서비스에 요청을 보내려면(예: 데이터 읽기) 자격 증명이 필요합니다. 이러한 자격 증명에는 Amazon EC2 인스턴스 또는 AWS Lambda 함수와 같이 AWS 환경에서 실행되는 시스템이 포함됩니다. 액세스가 필요한 외부 당사자의 시스템 자격 증명을 관리할 수도 있습니다. 또한 AWS 환경에 액세스해야 하는 시스템이 AWS 외부에 있을 수도 있습니다.

모범 사례

- SEC02-BP01 강력한 로그인 메커니즘 사용
- SEC02-BP02 임시 보안 인증 정보 사용
- SEC02-BP03 안전하게 보안 암호 저장 및 사용
- SEC02-BP04 중앙 집중식 자격 증명 공급자 사용
- SEC02-BP05 정기적으로 보안 인증 정보 감사 및 교체
- SEC02-BP06 사용자 그룹 및 속성 활용

SEC02-BP01 강력한 로그인 메커니즘 사용

로그인(로그인 보안 인증 정보를 사용한 인증)은 다중 인증(MFA)과 같은 메커니즘을 사용하지 않을 때, 특히 로그인 보안 인증 정보가 의도치 않게 공개되었거나 쉽게 추측되는 상황에서 위험을 초래할 수 있습니다. 강력한 로그인 메커니즘을 사용하면 MFA 및 강력한 암호 정책을 요구하여 이러한 위험을 줄일 수 있습니다.

원하는 결과: AWS Identity and Access Management(IAM) 사용자, AWS 계정 루트 사용자, AWS IAM Identity Center(AWS Single Sign-On의 후속 서비스), 타사 자격 증명 공급자에 대한 강력한 로그인 메커니즘을 사용하여 AWS의 보안 인증 정보에 대한 의도치 않은 액세스 위험을 줄입니다. 즉, MFA를 요구하고 강력한 암호 정책을 적용하며 비정상적인 로그인 동작을 감지합니다.

일반적인 안티 패턴:

- 복잡한 암호 및 MFA를 포함하여 자격 증명에 대한 강력한 암호 정책을 적용하지 않습니다.
- 다른 사용자 간에 동일한 보안 인증 정보를 공유합니다.
- 의심스러운 로그인에 대한 탐지 제어를 사용하지 않습니다.

이 모범 사례를 따르지 않을 경우 노출 위험도: 높음

구현 가이드

인적 자격 증명으로 AWS에 로그인하는 방법에는 여러 가지가 있습니다. AWS로 인증할 때 페더레이션(직접 페더레이션 또는 AWS IAM Identity Center 사용)을 사용하는 중앙 집중식 자격 증명 공급자를 사용하는 것이 AWS 모범 사례입니다. 이 경우 자격 증명 공급자 또는 Microsoft Active Directory를 사용하여 보안 로그인 프로세스를 설정해야 합니다.

AWS 계정를 처음 열면 AWS 계정 루트 사용자로 시작합니다. 루트 사용자 계정은 사용자(및 루트 사용자가 필요한 작업)에 대한 액세스 권한을 설정할 때만 사용해야 합니다. AWS 계정를 연 직후에는 계정 루트 사용자에 대해 MFA를 활성화하고 AWS 모범 사례 가이드를 사용하여 루트 사용자를 보호하는 것이 중요합니다.

AWS IAM Identity Center에서 사용자를 생성하는 경우 해당 서비스의 로그인 프로세스를 보호하세요. 소비자 자격 증명의 경우 Amazon Cognito user pools를 사용하고 해당 서비스에서 로그인 프로세스를 보호하거나 Amazon Cognito user pools가 지원하는 자격 증명 공급자 중 하나를 사용할 수 있습니다.

AWS Identity and Access Management(IAM) 사용자를 사용하는 경우 IAM을 사용하여 로그인 프로세스를 보호합니다.

로그인 방법에 관계없이 강력한 로그인 정책을 적용하는 것이 중요합니다.

구현 단계

다음은 일반적인 강력한 로그인 권장 사항입니다. 구성하는 실제 설정은 회사 정책에 따라 설정하거나 NIST 800-63과 같은 표준을 사용해야 합니다.

- MFA가 필요합니다. 인적 자격 증명 및 워크로드에 대해 MFA를 요구하는 것이 IAM 모범 사례입니다. MFA를 활성화하면 사용자가 로그인 보안 인증 정보와 일회용 암호(OTP) 또는 하드웨어 디바이스에서 암호로 확인 및 생성된 문자열을 제공해야 하는 추가 보안 계층이 제공됩니다.
- 암호 강도의 기본 요소인 최소 암호 길이를 적용합니다.
- 암호 복잡성을 적용하여 암호를 추측하기 어렵게 만듭니다.
- 사용자가 자신의 암호를 변경할 수 있도록 허용합니다.
- 공유 보안 인증 정보 대신 개별 자격 증명을 생성합니다. 개별 자격 증명을 생성하여 각 사용자에게 고유한 보안 보안 인증 정보를 제공할 수 있습니다. 개별 사용자는 각 사용자의 활동을 감사할 수 있는 기능을 제공합니다.

IAM Identity Center 권장 사항:

- IAM Identity Center는 암호 길이, 복잡성 및 재사용 요구 사항을 설정하는 기본 디렉터리를 사용할 때 사전 정의된 암호 정책을 제공합니다.
- MFA를 활성화하고 자격 증명 소스가 기본 디렉터리, AWS Managed Microsoft AD 또는 AD Connector인 경우 MFA에 대한 컨텍스트 인식 또는 상시 설정을 구성합니다.
- 사용자가 자신의 MFA 디바이스를 등록하도록 허용합니다.

Amazon Cognito user pools 디렉터리 권장 사항:

- 암호 강도 설정을 구성합니다.
- 사용자에게 MFA 설정을 요구합니다.
- 의심스러운 로그인을 차단할 수 있는 적응형 인증과 같은 기능에 대해 Amazon Cognito user pools 고급 보안 설정을 사용합니다.

IAM 사용자 권장 사항:

- IAM Identity Center 또는 직접 페더레이션을 사용하는 것이 좋습니다. 그러나 IAM 사용자가 필요할 수 있습니다. 이 경우 IAM 사용자에 대해 암호 정책을 설정합니다. 암호 정책을 사용하여 최소 길이 또는 알파벳 이외 문자 포함 여부 등과 같은 요구 사항을 정의할 수 있습니다.

- 사용자가 자신의 암호화 MFA 디바이스를 관리할 수 있도록 MFA 로그인을 적용하는 IAM 정책을 생성합니다.

리소스

관련 모범 사례:

- SEC02-BP03 안전하게 보안 암호 저장 및 사용
- SEC02-BP04 중앙 집중식 자격 증명 공급자 사용
- SEC03-BP08 안전하게 조직과 리소스 공유

관련 문서:

- AWS IAM Identity Center(AWS Single Sign-On의 후속 서비스) 암호 정책
- IAM 사용자 암호 정책
- AWS 계정 루트 사용자 암호 설정
- Amazon Cognito 암호 정책
- AWS 보안 인증 정보
- IAM 보안 모범 사례

관련 동영상:

- Managing user permissions at scale with AWS IAM Identity Center(AWS IAM Identity Center를 사용하여 대규모로 사용자 권한 관리)
- Mastering identity at every layer of the cake(케이크의 모든 계층에서 자격 증명 마스터링)

SEC02-BP02 임시 보안 인증 정보 사용

모든 유형의 인증을 수행할 때 보안 인증 정보가 의도치 않게 공개, 공유 또는 도난 당하는 위험을 줄이거나 제거하기 위해 장기 보안 인증 정보 대신 임시 보안 인증 정보를 사용하는 것이 가장 좋습니다.

원하는 결과: 장기 보안 인증 정보의 위험을 줄이려면 가능한 한 인적 자격 증명과 시스템 자격 증명 모두에 대해 임시 보안 인증 정보를 사용합니다. 장기 보안 인증 정보는 많은 위험을 초래합니다. 예를 들어 퍼블릭 GitHub 리포지토리에 코드를 업로드할 수 있습니다. 임시 보안 인증 정보를 사용하면 보안 인증 정보가 손상될 가능성이 크게 줄어듭니다.

일반적인 안티 패턴:

- 개발자가 페더레이션을 사용하여 CLI에서 임시 보안 인증 정보를 획득하는 대신 IAM users의 장기 액세스 키를 사용합니다.
- 개발자가 장기 액세스 키를 코드에 포함하고 해당 코드를 퍼블릭 Git 리포지토리에 업로드합니다.
- 개발자가 앱 스토어에서 사용할 수 있도록 장기 액세스 키를 모바일 앱에 포함합니다.
- 사용자가 다른 사용자와 장기 액세스 키를 공유하거나 직원이 장기 액세스 키를 소유한 상태로 퇴사합니다.
- 임시 보안 인증 정보를 사용할 수 있는 경우에도 시스템 자격 증명에 장기 액세스 키를 사용합니다.

이 모범 사례를 따르지 않을 경우 노출 위험도: 높음

구현 가이드

모든 AWS API 및 CLI 요청에 대해 장기 보안 인증 정보 대신 임시 보안 인증 정보를 사용합니다. AWS 서비스에 대한 API 및 CLI 요청은 거의 모든 경우에 AWS 액세스 키를 사용하여 서명해야 합니다. 이러한 요청은 임시 또는 장기 보안 인증 정보로 서명할 수 있습니다. 장기 액세스 키라고도 하는 장기 보안 인증 정보를 사용해야 하는 유일한 경우는 IAM 사용자 또는 AWS 계정 루트 사용자를 사용하는 경우입니다. 다른 방법을 통해 AWS에 페더레이션하거나 IAM 역할을 수임할 때 임시 보안 인증 정보가 생성됩니다. 로그인 보안 인증 정보를 사용하여 AWS Management Console에 액세스하는 경우에도 AWS 서비스를 호출할 수 있도록 임시 보안 인증 정보가 생성됩니다. 장기 보안 인증 정보가 필요한 경우는 거의 없으며 임시 보안 인증 정보를 사용하여 대부분의 작업을 수행할 수 있습니다.

임시 보안 인증 정보를 선호하여 장기 보안 인증 정보 사용을 피하는 것은 페더레이션 및 IAM 역할을 선호하여 IAM 사용자의 사용을 줄이는 전략과 함께 수행해야 합니다. 과거에는 IAM 사용자가 인적 자격 증명과 시스템 자격 증명 모두에 사용되었지만 이제는 장기 액세스 키 사용의 위험을 피하기 위해 사용하지 않는 것이 좋습니다.

구현 단계

직원, 관리자, 개발자, 운영자 및 고객과 같은 인적 자격 증명의 경우:

- 중앙 집중식 자격 증명 공급자를 사용해야 하며 인적 사용자가 임시 보안 인증 정보를 사용하여 AWS에 액세스하려면 자격 증명 공급자와의 페더레이션을 사용해야 합니다. 사용자에 대한 페더레이션은 각 AWS 계정에 대한 직접 페더레이션을 사용하거나 AWS IAM Identity Center(AWS IAM Identity Center의 후속 서비스) 및 선택한 자격 증명 공급자를 사용하여 수행할 수 있습니다. 페더레이션은 장기 보안 인증 정보를 제거하는 것 외에도 IAM 사용자를 사용하는 것보다 많은 이점을 제공

합니다. 사용자는 직접 페더레이션에 대한 명령줄에서 또는 IAM Identity Center를 사용하여 임시 보안 인증 정보를 요청할 수도 있습니다. 즉, IAM 사용자 또는 사용자에 대한 장기 보안 인증 정보가 필요한 사용 사례가 거의 없습니다.

- 서비스형 소프트웨어(SaaS) 공급자와 같은 타사에 AWS 계정의 리소스에 대한 액세스 권한을 부여할 때 크로스 계정 역할 및 리소스 기반 정책을 사용할 수 있습니다.
- 소비자 또는 고객에게 AWS 리소스에 대한 액세스 권한을 부여해야 하는 경우 Amazon Cognito 자격 증명 풀 또는 Amazon Cognito user pools을 사용하여 임시 보안 인증 정보를 제공할 수 있습니다. 보안 인증 정보에 대한 권한은 IAM 역할을 통해 구성됩니다. 또한 인증되지 않은 게스트 사용자의 경우, 권한이 제한된 별도의 IAM 역할을 정의할 수 있습니다.

시스템 자격 증명의 경우 장기 보안 인증 정보를 사용해야 할 수 있습니다. 이러한 경우 워크로드에서 AWS에 액세스하려면 IAM 역할과 함께 임시 보안 인증 정보를 사용해야 합니다.

- Amazon Elastic Compute Cloud(Amazon EC2)의 경우 Amazon EC2에 대한 역할을 사용할 수 있습니다.

- AWS Lambda를 사용하면 임시 보안 인증 정보를 사용하여 AWS 작업을 수행할 수 있는 서비스 권한을 부여하도록 Lambda 실행 역할을 구성할 수 있습니다. IAM 역할을 사용하여 임시 보안 인증 정보를 부여하는 AWS 서비스에 대한 다른 유사한 모델이 많이 있습니다.
- IoT 디바이스의 경우 AWS IoT Core 보안 인증 정보 공급자를 사용하여 임시 보안 인증 정보를 요청할 수 있습니다.
- 온프레미스 시스템 또는 AWS 리소스에 액세스해야 하는 AWS 외부에서 실행되는 시스템의 경우 IAM Roles Anywhere를 사용할 수 있습니다.

임시 보안 인증 정보를 사용할 수 없고 장기 보안 인증 정보를 사용해야 하는 시나리오가 있습니다. 이러한 상황에서는 보안 인증 정보를 주기적으로 감사 및 교체하고 장기 보안 인증 정보가 필요한 사용 사례를 위해 정기적으로 액세스 키를 교체합니다. 장기 보안 인증 정보가 필요할 수 있는 몇 가지 예로는 WordPress 플러그인 및 타사 AWS 클라이언트가 있습니다. 장기 보안 인증 정보를 사용해야 하는 상황이나 데이터베이스 로그인과 같이 AWS 액세스 키 이외의 보안 인증 정보에 대해 AWS Secrets Manager와 같은 보안 암호 관리를 처리하도록 설계된 서비스를 사용할 수 있습니다. Secrets Manager를 사용하면 지원되는 서비스를 사용하여 암호화된 보안 암호를 간단하게 관리 및 교체하고 안전하게 저장할 수 있습니다. 장기 보안 인증 정보 교체에 대한 자세한 내용은 액세스 키 교체를 참조하세요.

리소스

관련 모범 사례:

- SEC02-BP03 안전하게 보안 암호 저장 및 사용
- SEC02-BP04 중앙 집중식 자격 증명 공급자 사용
- SEC03-BP08 안전하게 조직과 리소스 공유

관련 문서:

- 임시 보안 인증 정보
- AWS 보안 인증 정보
- IAM 보안 모범 사례
- IAM 역할
- IAM Identity Center
- 자격 증명 공급자 및 페더레이션
- 액세스 키 교체
- 보안 파트너 솔루션: 액세스 및 액세스 제어
- AWS 계정 루트 사용자

관련 동영상:

- Managing user permissions at scale with AWS IAM Identity Center(successor to AWS IAM Identity Center)(AWS IAM Identity Center(AWS IAM Identity Center의 후속 버전)를 사용하여 대규모로 사용자 권한 관리)
- Mastering identity at every layer of the cake(케이크의 모든 계층에서 자격 증명 마스터링)

SEC02-BP03 안전하게 보안 암호 저장 및 사용

워크로드에는 데이터베이스, 리소스 및 타사 서비스에 대한 아이덴티티를 증명하는 자동화된 기능이 필요합니다. 이는 API 액세스 키, 암호, OAuth 토큰과 같은 보안 암호 액세스 보안 인증 정보를 사용하여 수행됩니다. 특별 제작된 서비스를 사용하여 이러한 보안 인증 정보를 저장, 관리 및 교체하면 해당 보안 인증 정보가 손상될 가능성을 줄이는 데 도움이 됩니다.

원하는 결과: 다음 목표를 달성하는 애플리케이션 보안 인증 정보를 안전하게 관리하기 위한 메커니즘을 구현합니다.

- 워크로드에 필요한 보안 암호를 식별합니다.

- 가능한 경우 장기 보안 인증 정보를 단기 보안 인증 정보로 대체하여 필요한 장기 보안 인증 정보의 수를 줄입니다.
- 나머지 장기 보안 인증 정보의 안전한 저장 및 자동 교체를 설정합니다.
- 워크로드에 존재하는 보안 암호에 대한 액세스 권한을 감사합니다.
- 개발 프로세스 중에 소스 코드에 보안 암호가 포함되어 있지 않은지 확인하기 위해 지속적으로 모니터링합니다.
- 보안 인증 정보가 의도치 않게 공개될 가능성을 줄입니다.

일반적인 안티 패턴:

- 자격 증명을 교체하지 않습니다.
- 소스 코드 또는 구성 파일에 장기 보안 인증 정보를 저장합니다.
- 보안 인증 정보를 저장 시 암호화되지 않은 상태로 저장합니다.

이 모범 사례 확립의 이점:

- 보안 암호는 저장 시 및 전송 중 암호화된 상태로 저장됩니다.
- 보안 인증 정보에 대한 액세스 권한은 API를 통해 제한됩니다(보안 인증 정보 벤딩 머신으로 간주).
- 보안 인증 정보에 대한 액세스 권한(읽기 및 쓰기 모두)은 감사되고 로깅됩니다.
- 우려 사항 분리: 보안 인증 정보 교체는 아키텍처의 나머지 부분과 분리될 수 있는 별도의 구성 요소에 의해 수행됩니다.
- 보안 암호는 필요에 따라 소프트웨어 구성 요소에 자동으로 배포되며 중앙 위치에서 교체가 수행됩니다.
- 보안 인증 정보에 대한 액세스 권한은 세분화된 방식으로 제어할 수 있습니다.

이 모범 사례를 따르지 않을 경우 노출 위험도: 높음

구현 가이드

과거에는 데이터베이스, 타사 API, 토큰 및 기타 보안 암호에 인증하는 데 사용되는 보안 인증 정보가 소스 코드나 환경 파일에 포함되었을 수 있습니다. AWS는 이러한 보안 인증 정보를 안전하게 저장하고 자동으로 교체하며 사용을 감사하는 여러 메커니즘을 제공합니다.

보안 암호 관리에 접근하는 가장 좋은 방법은 제거, 대체 및 교체 지침을 따르는 것입니다. 가장 안전한 보안 인증 정보는 저장, 관리 또는 처리할 필요가 없는 보안 인증 정보입니다. 안전하게 제거할 수 있는 워크로드 기능에 더 이상 필요하지 않은 보안 인증 정보가 있을 수 있습니다.

워크로드의 적절한 기능을 위해 여전히 필요한 보안 인증 정보의 경우 장기 보안 인증 정보를 임시 또는 단기 보안 인증 정보로 대체할 기회가 있을 수 있습니다. 예를 들어 AWS 비밀 액세스 키를 하드 코딩하는 대신 IAM 역할을 사용하여 해당 장기 보안 인증 정보를 임시 보안 인증 정보로 대체하는 것이 좋습니다.

일부 수명이 긴 보안 암호는 제거하거나 대체하지 못할 수 있습니다. 이러한 보안 암호는 AWS Secrets Manager와 같은 서비스에 저장될 수 있으며 정기적으로 중앙에서 저장, 관리 및 교체할 수 있습니다.

워크로드의 소스 코드 및 구성 파일을 감사하면 여러 유형의 보안 인증 정보를 확인할 수 있습니다. 다음 표에는 일반적인 유형의 보안 인증 정보를 처리하기 위한 전략이 요약되어 있습니다.

Credential type	Description	Suggested strategy
IAM access keys	AWS IAM access and secret keys used to assume IAM roles inside of a workload	Replace: Use IAM 역할 assigned to the compute instances (such as Amazon EC2 or AWS Lambda) instead. For interoperability with third parties that require access to resources in your AWS 계정, ask if they support AWS 크로스 계정 액세스. For mobile apps, consider using temporary credentials through Amazon Cognito 자격 증명 풀 (페더레이션 자격 증명). For workloads running outside of AWS, consider IAM Roles Anywhere or AWS Systems Manager 하이브리드 활성화.
SSH keys	Secure Shell private keys used to log into Linux EC2	Replace: Use AWS Systems Manager or EC2 Instance Connect to provide

Credential type	Description	Suggested strategy
	instances, manually or as part of an automated process	programmatic and human access to EC2 instances using IAM roles.
Application and database credentials	Passwords – plain text string	Rotate: Store credentials in AWS Secrets Manager and establish automated rotation if possible.
Amazon RDS and Aurora Admin Database credentials	Passwords – plain text string	Replace: Use the Amazon RDS와 Secrets Manager 통합 or Amazon Aurora. In addition, some RDS database types can use IAM roles instead of passwords for some use cases (for more detail, see IAM 데이터베이스 인증).
OAuth tokens	Secret tokens – plain text string	Rotate: Store tokens in AWS Secrets Manager and configure automated rotation.
API tokens and keys	Secret tokens – plain text string	Rotate: Store in AWS Secrets Manager and establish automated rotation if possible.

일반적인 안티 패턴은 소스 코드, 구성 파일 또는 모바일 앱 내부에 IAM 액세스 키를 포함하는 것입니다. AWS 서비스와 통신하기 위해 IAM 액세스 키가 필요한 경우 임시(단기) 보안 인증 정보를 사용합니다. 이러한 단기 보안 인증 정보는 EC2 인스턴스의 경우 IAM 역할, Lambda 함수의 경우 실행 역할, 모바일 사용자 액세스의 경우 Cognito IAM 역할 및 IoT 기기의 경우 IoT Core 정책을 통해 제공할 수 있습니다. 타사와 인터페이스할 때 IAM 사용자를 구성하고 타사에 해당 사용자의 비밀 액세스 키를 보내는 것보다 계정 리소스에 필수 액세스 권한이 있는 IAM 역할에 대한 액세스 권한을 위임하는 것이 좋습니다.

워크로드에 다른 서비스 및 리소스와 상호 운용하는 데 필요한 보안 암호를 저장해야 하는 경우가 많습니다. AWS Secrets Manager는 이러한 보안 인증 정보는 물론 API 토큰, 암호 및 기타 보안 인증 정보의 저장, 사용 및 교체를 안전하게 관리하기 위해 특별히 제작되었습니다.

AWS Secrets Manager는 민감한 보안 인증 정보의 안전한 저장 및 처리를 보장하는 5가지 주요 기능, 즉 저장 시 암호화, 전송 중 암호화, 종합적 감사, 세분화된 액세스 제어, 확장 가능한 보안 인증 정보 교체를 제공합니다. AWS 파트너의 기타 보안 암호 관리 서비스 또는 유사한 기능과 보증을 제공하는 현지 개발 솔루션도 허용됩니다.

구현 단계

1. Amazon CodeGuru 같은 자동화 도구를 사용하여 하드 코딩된 보안 인증 정보가 포함된 코드 경로를 식별합니다.
   - Amazon CodeGuru를 사용하여 코드 리포지토리를 스캔합니다. 검토가 완료되면 CodeGuru에서 `Type=Secrets`를 필터링하여 문제가 있는 코드 줄을 찾습니다.
2. 제거하거나 대체할 수 있는 보안 인증 정보를 식별합니다.
   a. 더 이상 필요하지 않은 보안 인증 정보를 식별하고 제거하도록 표시합니다.
   b. 소스 코드에 포함된 AWS 보안 암호 키의 경우 필요한 리소스와 연결된 IAM 역할로 대체합니다. 워크로드의 일부가 AWS 외부에 있지만 AWS 리소스에 액세스하기 위해 IAM 보안 인증 정보가 필요한 경우 IAM Roles Anywhere 또는 AWS Systems Manager 하이브리드 활성화를 고려합니다.
3. 교체 전략을 사용해야 하는 타사의 수명이 긴 보안 암호의 경우 Secrets Manager를 코드에 통합하여 런타임 시 타사 보안 암호를 검색합니다.
   a. CodeGuru 콘솔은 검색된 보안 인증 정보를 사용하여 자동으로 Secrets Manager에 보안 암호를 생성할 수 있습니다.
   b. Secrets Manager의 보안 암호 검색을 애플리케이션 코드에 통합합니다.
      - 서버리스 Lambda 함수는 언어에 구애받지 않는 Lambda 확장을 사용할 수 있습니다.
      - EC2 인스턴스 또는 컨테이너의 경우 AWS는 널리 사용되는 여러 프로그래밍 언어로 Secrets Manager에서 보안 암호를 검색하기 위한 클라이언트 측 코드 예를 제공합니다.
4. 주기적으로 코드 베이스를 검토하고 다시 스캔하여 코드에 추가된 새 보안 암호가 없는지 확인합니다.
   - git-secrets와 같은 도구를 사용하여 소스 코드 리포지토리에 새 보안 암호를 커밋되지 않도록 하는 것이 좋습니다.
5. Secrets Manager 활동을 모니터링하여 예상치 못한 사용, 부적절한 보안 암호 액세스 또는 보안 암호 삭제 시도가 있는지 확인합니다.

6. 보안 인증 정보에 대한 인적 노출을 줄입니다. 보안 인증 정보를 읽고 쓰고 수정할 수 있는 액세스 권한을 이 목적을 위한 전용 IAM 역할로 제한하고, 해당 역할을 수임할 수 있는 액세스 권한을 일부 운영 사용자에게만 제공합니다.

리소스

관련 모범 사례:

- SEC02-BP02 임시 보안 인증 정보 사용
- SEC02-BP05 정기적으로 보안 인증 정보 감사 및 교체

관련 문서:

- AWS Secrets Manager 시작하기
- 자격 증명 공급자 및 페더레이션
- Amazon CodeGuru에서 Secrets Detector 도입
- AWS Secrets Manager의 AWS Key Management Service 사용 방법
- Secrets Manager에서 보안 암호 암호화 및 복호화
- Secrets Manager 블로그 항목
- Amazon RDS에서 AWS Secrets Manager와의 통합 발표

관련 동영상:

- Best Practices for Managing, Retrieving, and Rotating Secrets at Scale(대규모 보안 암호 관리, 검색, 교체 모범 사례)
- Find Hard-Coded Secrets Using Amazon CodeGuru Secrets Detector(Amazon CodeGuru Secrets Detector를 사용하여 하드 코딩된 보안 암호 찾기)
- Securing Secrets for Hybrid Workloads Using AWS Secrets Manager(AWS Secrets Manager를 사용하여 하이브리드 워크로드에 대한 보안 암호 보호)

관련 워크숍:

- AWS Secrets Manager에서 민감한 보안 인증 정보 저장, 검색, 관리
- AWS Systems Manager 하이브리드 활성화

SEC02-BP04 중앙 집중식 자격 증명 공급자 사용

직원 ID(직원 및 계약업체)의 경우 중앙 위치에서 ID를 관리할 수 있는 ID 공급업체를 이용하세요. 이렇게 하면 단일 위치에서 액세스를 생성, 관리 및 취소하므로 여러 애플리케이션과 서비스에 대한 액세스를 더 쉽게 관리할 수 있습니다.

원하는 결과: 중앙 집중식 ID 공급자를 통해 직원 사용자, 인증 정책(예: MFA (Multi-Factor Authentication) 요구), 시스템 및 애플리케이션에 대한 권한 부여(예: 사용자의 그룹 구성원 자격 또는 특성에 따른 액세스 할당)를 중앙에서 관리할 수 있습니다. 직원 사용자는 중앙 ID 공급자에 로그인하고 내부 및 외부 애플리케이션에 페더레이션(Single Sign-On) 하므로 사용자가 여러 자격 증명을 기억할 필요가 없습니다. ID 공급자는 인사(HR) 시스템과 통합되므로 직원 변경 사항이 ID 공급자와 자동으로 동기화됩니다. 예를 들어, 누군가가 조직을 떠나는 경우 페더레이션된 응용 프로그램 및 시스템(AWS 포함)에 대한 액세스를 자동으로 취소할 수 있습니다. ID 공급자에서 세부 감사 로깅을 활성화했으며 이러한 로그를 모니터링하여 비정상적인 사용자 동작이 있는지 확인하고 있습니다.

일반적인 안티 패턴:

- 페더레이션 및 싱글 사인온은 사용하지 않습니다. 직원이 여러 애플리케이션과 시스템에서 별도의 사용자 계정과 자격 증명을 생성합니다.
- ID 공급자를 HR 시스템에 통합하는 등 직원 사용자의 ID 라이프사이클을 자동화하지 않습니다. 사용자가 조직을 떠나거나 역할을 변경하면 수동 프로세스에 따라 여러 애플리케이션 및 시스템에서 기록을 삭제하거나 업데이트합니다.

이 모범 사례 확립의 이점: 중앙 집중식 ID 공급자를 사용하면 직원 사용자 ID 및 정책을 한 곳에서 관리하고, 사용자와 그룹에 애플리케이션 액세스 권한을 할당하고, 사용자 로그인 활동을 모니터링할 수 있습니다. 인사(HR) 시스템과 통합하면 사용자가 역할을 변경하면 이러한 변경 내용이 ID 공급자와 동기화되고 할당된 애플리케이션 및 권한이 자동으로 업데이트됩니다. 사용자가 조직을 떠나면 ID 공급자에서 해당 ID가 자동으로 비활성화되어 페더레이션된 애플리케이션 및 시스템에 대한 액세스 권한이 취소됩니다.

이 모범 사례가 확립되지 않을 경우 노출되는 위험 수준:높음

구현 가이드

AWS에 액세스하는 직원을 위한 지침

조직의 직원 및 계약직과 같은 인력 사용자는 직무를 수행하기 위해 AWS Management Console 또는 AWS Command Line Interface(AWS CLI)를 사용하여 AWS에 대한 액세스 권한이 필요할 수 있습니

다. 중앙 집중식 ID 공급자를 두 가지 AWS 수준으로 페더레이션하여 직원 사용자에게 AWS 액세스 권한을 부여할 수 있습니다. 즉, AWS 조직에서 각 공급자에 대해 직접 페더레이션 AWS 계정 또는 여러 계정에 페더레이션하는 것입니다.

- 인력 사용자를 각 AWS 계정 사용자와 직접 페더레이션하려면 중앙 집중식 ID 공급자를 사용하여 그 계정에서 다음과 같이 AWS Identity and Access Management 페더레이션할 수 있습니다. IAM의 유연성을 통해 별도의 기능을 활성화할 수 있습니다. SAML 2.0 또는 오픈 ID 커넥트 (OIDC) 각각에 대한 ID AWS 계정 공급자이며 액세스 제어를 위한 통합 사용자 속성을 사용합니다. 직원은 웹 브라우저를 사용하여 자격 증명(예: 암호 및 MFA 토큰 코드)을 제공하여 ID 공급자에 로그인합니다. ID 공급자가 브라우저에 SAML 어설션을 발행하여 AWS Management Console 로그인 URL에 제출하여 사용자가 SSO에 싱글 사인온할 수 있도록 합니다. AWS Management Console역할을 맡음으로써 IAM. 또한 사용자는 ID 공급자로부터 SAML 어설션을 사용하여 IAM 역할을 가정하여 AWS STS에서 AWS CLI 또는 AWS SKD에서 사용할 임시 AWS API 자격 증명을 얻을 수 있습니다.
- AWS 조직에서 여러 계정을 가진 인력 사용자를 페더레이션하려면 AWS IAM Identity Center를 사용하여 AWS 계정 및 애플리케이션에 대한 인력 사용자의 액세스를 중앙에서 관리할 수 있습니다. 조직의 ID 센터를 활성화하고 ID 소스를 구성합니다.IAM Identity Center사용자 및 그룹을 관리하는 데 사용할 수 있는 기본 ID 소스 디렉터리를 제공합니다. 또는 SAML 2.0을 사용하여 외부 ID 공급자에 연결하고 SCIM을 사용하여 사용자 및 그룹을 자동으로 프로비저닝하거나 AWS Directory Service를 사용하여 Microsoft AD 디렉터리에 연결하여 외부 ID 소스를 선택할 수도 있습니다. ID 소스가 구성되면 권한 집합에 최소 권한 정책을 정의하여 사용자 및 그룹에 AWS 계정에 대한 액세스 권한을 할당할 수 있습니다. 직원 사용자는 중앙 ID 공급자를 통해 인증하여 다음에 로그인할 수 있습니다. AWS 액세스 포털 그리고 그들에게 할당된 클라우드 애플리케이션에도 싱글 AWS 계정 사인온을 할 수 있습니다. 사용자는 다음을 구성할 수 있습니다. AWS CLI v2: ID 센터를 통해 인증하고 AWS CLI 명령을 실행하기 위한 자격 증명을 얻을 수 있습니다. 또한ID 센터를 사용하면 AWS 애플리케이션에 싱글 사인온 액세스할 수 있습니다. 예시로는 Amazon SageMaker Studio 및 AWS IoT Sitewise Monitor 포털들이 있습니다.

위의 지침을 따른 후에는 작업자가 AWS 워크로드를 관리할 때 정상적인 작업을 위해 더 이상 IAM users와 그룹을 사용할 필요가 없습니다. 대신 사용자와 그룹은 외부에서 AWS 관리되며 사용자는 연동 자격 증명과 같이 AWS 리소스에 액세스할 수 있습니다. 페더레이션 ID는 중앙 ID 공급자가 정의한 그룹을 사용합니다. AWS 계정에서 더 이상 필요하지 않은 IAM 그룹, IAM users 그리고 수명이 긴 사용자 자격 증명(암호 및 액세스 키)을 식별하고 제거해야 합니다. IAM 자격 증명 보고서를 사용하여 사용하지 않는 자격 증명을 찾고, 해당 IAM users를 및 IAM 그룹을 삭제할 수 있습니다. 조직에 서비스 제어 정책(SCP)를 적용하여 새로운 IAM users 및 그룹 생성을 방지하고 페더레이션된 ID를 통해 AWS에 액세스하도록 강제할 수 있습니다.

애플리케이션 사용자를 위한 지침

모바일 앱과 같은 애플리케이션 사용자의 ID를 중앙 ID 공급자로 Amazon Cognito를 사용하여 관리할 수 있습니다. Amazon Cognito은 웹 및 모바일 앱에 대한 인증, 권한 부여 및 사용자 관리를 지원합니다. Amazon Cognito는 수백만 명의 사용자로 확장 가능한 ID 저장소를 제공하고, 소셜 및 엔터프라이즈 ID 페더레이션을 지원하며, 고급 보안 기능을 제공하여 사용자와 비즈니스를 보호할 수 있도록 지원합니다. 사용자 지정 웹 또는 모바일 애플리케이션을 Amazon Cognito 통합하여 몇 분 만에 애플리케이션에 사용자 인증 및 액세스 제어를 추가할 수 있습니다. SAML 및 Open ID Connect (OIDC) 와 같은 개방형 ID 표준을 기반으로 하며 다양한 규정 준수 규정을 Amazon Cognito 지원하고 프런트엔드 및 백엔드 개발 리소스와 통합됩니다.

구현 단계

직원 AWS 액세스 단계

- 다음 접근 방식 중 하나인 AWS 사용하여 직원 사용자를 중앙 집중식 ID 공급자를 사용하도록 통합하세요.
  - ID 공급자와 IAM Identity Center 페더레이션하여 AWS 조직 AWS 계정 내 여러 명이 Single Sign-On을 사용할 수 있도록 하는 데 사용합니다.
  - ID 공급자를 AWS 계정 각각에 직접 IAM 연결하여 통합 세분화된 액세스를 가능하게 하는 데 사용합니다.
- 페더레이션 ID로 대체되는 항목을 식별 IAM users 및 제거하고 그룹화합니다.

애플리케이션 사용자를 위한 단계

- Amazon Cognito를 애플리케이션에 대한 중앙 집중식 ID 공급자로 사용하세요.
- OpenID Connect 및 Amazon Cognito OAuth를 사용하여 사용자 지정 애플리케이션을 통합하세요. 인증과 같은 Amazon Cognito 다양한 AWS 서비스와 통합할 수 있는 간단한 인터페이스를 제공하는 Amplify 라이브러리를 사용하여 사용자 지정 애플리케이션을 개발할 수 있습니다.

리소스

관련 Well-Architected 모범 사례:

- SEC02-BP06 사용자 그룹 및 속성 활용
- SEC03-BP02 최소 권한 액세스 부여
- SEC03-BP06 수명 주기에 따라 액세스 관리

관련 문서:

- AWS의 아이덴티티 페더레이션
- IAM의 보안 모범 사례
- AWS Identity and Access Management 모범 사례
- IAM Identity Center위임 관리 시작하기
- 고급 사용 사례에서 IAM Identity Center 고객 관리형 정책을 사용하는 방법
- AWS CLIv2: IAM Identity Center 자격 증명 공급자

관련 동영상:

- AWS re:Inforce 2022 - AWS Identity and Access Management (IAM) deep dive
- AWS re:Invent 2022 - Simplify your existing workforce access with IAM Identity Center
- AWS re:Invent 2018: Mastering identity at every layer of the cake

관련 예시:

- 워크숍: 강력한 ID 관리를 위해 AWS IAM Identity Center 사용
- 워크숍: 서버리스 ID

관련 도구:

- AWS 보안 컴피턴시 파트너: ID 및 액세스 관리
- saml2aws

SEC02-BP05 정기적으로 보안 인증 정보 감사 및 교체

보안 인증 정보를 주기적으로 감사하고 교체하여 리소스에 액세스하는 데 보안 인증 정보를 사용할 수 있는 기간을 제한합니다. 장기 보안 인증 정보는 많은 위험을 초래하며 이러한 위험은 장기 보안 인증 정보를 정기적으로 교체하여 줄일 수 있습니다.

원하는 결과: 장기 보안 인증 정보 사용과 관련된 위험을 줄이는 데 도움이 되는 보안 인증 정보 교체를 구현합니다. 보안 인증 정보 교체 정책 미준수를 정기적으로 감사하고 개선합니다.

일반적인 안티 패턴:

- 보안 인증 정보 사용을 감사하지 않습니다.

- 장기 보안 인증 정보를 불필요하게 사용합니다.
- 장기 보안 인증 정보를 사용하고 정기적으로 교체하지 않습니다.

이 모범 사례를 따르지 않을 경우 노출 위험도: 중간

구현 가이드

임시 보안 인증 정보를 사용할 수 없으며 장기 보안 인증 정보가 필요한 경우 보안 인증 정보를 감사하여 정의된 제어(예: 다중 인증(MFA))가 적용되고 정기적으로 교체되며 적절한 액세스 수준을 보유하는지 확인합니다.

올바른 제어 기능이 적용되는지 확인하려면 주기적인 검증(가능한 자동화된 도구 사용)을 실시해야 합니다. 인적 자격 증명의 경우, 사용자가 주기적으로 암호를 변경하고 액세스 키 사용을 중지하며 그 대신 임시 보안 인증 정보를 사용하도록 규정해야 합니다. AWS Identity and Access Management(IAM) 사용자에서 중앙 집중식 자격 증명으로 이동할 때 보안 인증 정보 보고서를 생성하여 사용자를 감사할 수 있습니다.

또한 자격 증명 공급자에서 MFA를 적용하고 모니터링하는 것이 좋습니다. AWS Config 규칙를 설정하거나 AWS Security Hub 보안 표준을 사용하여 사용자가 MFA를 활성화했는지 모니터링할 수 있습니다. 시스템 자격 증명에 대한 임시 보안 인증 정보를 제공하려면 IAM Roles Anywhere를 사용하는 것이 좋습니다. IAM 역할 및 임시 보안 인증 정보를 사용할 수 없는 상황에서는 빈번한 감사 및 교체 액세스 키가 필요합니다.

구현 단계

- 정기적으로 보안 인증 정보 감사: 자격 증명 공급자 및 IAM에 구성된 보안 인증 정보를 감사하면 권한이 부여된 자격 증명만 워크로드에 액세스할 수 있는지 확인할 수 있습니다. 이러한 자격 증명에는 IAM 사용자, AWS IAM Identity Center 사용자, Active Directory 사용자 또는 다른 업스트림 자격 증명 공급자의 사용자가 포함될 수 있지만 이에 국한되지 않습니다. 예를 들어 퇴사자의 계정을 제거하고 더 이상 필요하지 않은 크로스 계정 역할을 제거합니다. IAM 엔터티가 액세스하는 서비스에 대한 권한을 정기적으로 감사하는 프로세스가 있어야 합니다. 이렇게 하면 사용되지 않는 권한을 제거하기 위해 수정해야 하는 정책을 식별하는 데 도움이 됩니다. 보안 인증 정보 보고서 및 AWS Identity and Access Management Access Analyzer를 사용하여 IAM 자격 증명 및 권한을 감사합니다. Amazon CloudWatch를 사용하여 AWS 환경 내에서 호출된 특정 API 호출에 대한 경보를 설정할 수 있습니다. Amazon GuardDuty는 예상치 못한 활동에 대해 알릴 수도 있습니다. 이는 IAM 보안 인증 정보에 대한 지나치게 관대한 액세스 또는 의도치 않은 액세스를 나타낼 수 있습니다.
- 정기적으로 보안 인증 정보 교체: 임시 보안 인증 정보를 사용할 수 없는 경우 장기 IAM 액세스 키를 정기적으로 교체합니다(최대 90일마다). 자신도 모르게 액세스 키가 의도치 않게 공개된 경우 보안

인증 정보를 사용하여 리소스에 액세스할 수 있는 기간이 제한됩니다. IAM 사용자의 액세스 키 교체에 대한 자세한 내용은 액세스 키 교체를 참조하세요.

- IAM 권한 검토: AWS 계정의 보안을 강화하기 위해 각 IAM 정책을 정기적으로 검토하고 모니터링합니다. 정책이 최소 권한 원칙을 준수하는지 확인합니다.
- IAM 리소스 생성 및 업데이트 자동화 고려: IAM Identity Center는 역할 및 정책 관리와 같은 많은 IAM 작업을 자동화합니다. 또는 AWS CloudFormation을 사용하면 템플릿을 확인하고 버전을 제어할 수 있으므로, 역할 및 정책을 포함한 IAM 리소스 배포를 자동화하여 인적 오류가 발생할 가능성을 줄일 수 있습니다.
- IAM Roles Anywhere를 사용하여 시스템 자격 증명의 IAM 사용자 대체: IAM Roles Anywhere를 사용하면 온프레미스 서버와 같이 기존에는 사용할 수 없었던 영역에서 역할을 사용할 수 있습니다. IAM Roles Anywhere는 신뢰할 수 있는 X.509 인증서를 사용하여 AWS에 인증하고 임시 보안 인증 정보를 받습니다. IAM Roles Anywhere를 사용하면 장기 보안 인증 정보가 온프레미스 환경에 더 이상 저장되지 않으므로 이러한 보안 인증 정보를 교체할 필요가 없습니다. 만료가 가까워지면 X.509 인증서를 모니터링하고 교체해야 합니다.

리소스

관련 모범 사례:

- SEC02-BP02 임시 보안 인증 정보 사용
- SEC02-BP03 안전하게 보안 암호 저장 및 사용

관련 문서:

- AWS Secrets Manager 시작하기
- IAM 모범 사례
- 자격 증명 공급자 및 페더레이션
- 보안 파트너 솔루션: 액세스 및 액세스 제어
- 임시 보안 인증 정보
- AWS 계정의 보안 인증 정보 보고서 가져오기

관련 동영상:

- Best Practices for Managing, Retrieving, and Rotating Secrets at Scale(대규모 보안 암호 관리, 검색, 교체 모범 사례)

- Managing user permissions at scale with AWS IAM Identity Center(AWS IAM Identity Center를 사용하여 대규모로 사용자 권한 관리)
- Mastering identity at every layer of the cake(케이크의 모든 계층에서 자격 증명 마스터링)

관련 예시:

- Well-Architected Lab - Automated IAM User Cleanup(자동화된 IAM 사용자 정리)
- Well-Architected Lab - Automated Deployment of IAM Groups and Roles(IAM 그룹 및 역할의 자동 배포)

SEC02-BP06 사용자 그룹 및 속성 활용

관리하는 사용자 수가 늘어나면서 사용자를 체계적으로 정리하여 대규모로 관리할 방법을 결정해야 합니다. 공통 보안 요구 사항이 있는 사용자들을 자격 증명 공급자가 정의한 그룹에 배치하고, 액세스 제어에 사용할 수 있는 사용자 속성(예: 부서 또는 위치)이 정확하게 업데이트되었는지 확인하는 메커니즘을 적절히 설정합니다. 액세스를 제어할 때는 개별적인 사용자가 아니라 이러한 그룹과 속성을 사용합니다. 이를 통해 사용자의 액세스 권한을 변경해야 할 때 여러 개별 정책을 업데이트하는 대신 권한 세트를 사용해 사용자의 그룹 멤버십 또는 속성을 한 번 변경하여 중앙에서 액세스를 관리할 수 있습니다. AWS IAM Identity Center(IAM Identity Center)를 사용하여 사용자 그룹 및 속성을 관리할 수 있습니다. IAM Identity Center는 가장 보편적으로 사용되는 속성을 지원합니다. 사용자 생성 중에 수동으로 입력한 것이든, 동기화 엔진을 사용하여 자동으로 프로비저닝된 것(예: System for Cross-Domain Identity Management(SCIM) 사양에 정의된 대로)이든 마찬가지입니다.

공통 보안 요구 사항이 있는 사용자들을 자격 증명 공급자가 정의한 그룹에 배치하고, 액세스 제어에 사용할 수 있는 사용자 속성(예: 부서 또는 위치)이 정확하게 업데이트되었는지 확인하는 메커니즘을 적절히 설정합니다. 개별 사용자가 아닌 이러한 그룹 및 속성을 사용하여 액세스를 제어합니다. 이를 통해 사용자의 액세스 권한을 변경해야 할 때 여러 개별 정책을 업데이트하는 대신 사용자의 그룹 멤버십 또는 속성을 한 번 변경하여 중앙에서 액세스를 관리할 수 있습니다.

이 모범 사례가 수립되지 않을 경우 노출되는 위험의 수준: 낮음

구현 가이드

- AWS IAM Identity Center(IAM Identity Center)를 사용하는 경우 그룹 구성: IAM Identity Center는 사용자 그룹을 구성하고 그룹에 원하는 수준의 권한을 할당할 수 있는 기능을 제공합니다.
  - AWS Single Sign-On - 자격 증명 관리

- 속성 기반 액세스 제어(ABAC) 자세히 알아보기: ABAC는 속성을 기반으로 권한을 정의하는 권한 부여 전략입니다.
  - AWS용 ABAC란 무엇입니까?
  - 실습: EC2에 대한 IAM 태그 기반 액세스 제어

리소스

관련 문서:

- AWS Secrets Manager 시작하기
- IAM 모범 사례
- 자격 증명 공급자 및 연동
- AWS 계정 루트 사용자

관련 동영상:

- Best Practices for Managing, Retrieving, and Rotating Secrets at Scale(대규모 보안 암호 관리, 검색, 교체 모범 사례)
- Managing user permissions at scale with AWS IAM Identity Center(AWS SSO를 사용하여 대규모로 사용자 권한 관리)
- Mastering identity at every layer of the cake

관련 예시:

- 실습: EC2에 대한 IAM 태그 기반 액세스 제어

권한 관리

AWS 및 워크로드에 액세스해야 하는 인적 자격 증명 및 시스템 자격 증명에 대한 액세스를 제어하는 권한을 관리합니다. 권한은 누가 어떤 조건에서 무엇에 액세스할 수 있는지를 제어합니다. 특정 리소스의 특정 서비스 작업에 대한 액세스 권한을 부여하려면 구체적인 인적 및 시스템 자격 증명에 권한을 설정합니다. 또한 액세스 권한을 부여하려면 '참'이어야 하는 조건을 지정합니다. 예를 들어 개발자에게 새 Lambda 함수를 생성하도록 허용하되, 특정 리전에서만 가능하도록 제한할 수 있습니다. 대규모로 AWS 환경을 관리할 때는 다음과 같은 모범 사례를 준수하여 자격 증명에 필요한 액세스만 부여하도록 해야 합니다.

서로 다른 유형의 리소스에 액세스 권한을 부여하는 방법에는 여러 가지가 있습니다. 그중 하나는 서로 다른 정책 유형을 사용하는 것입니다.

IAM의 자격 증명 기반 정책은 관리형이거나 인라인 방식이며, 사용자나 그룹, 역할을 포함한 IAM 자격 증명에 연결됩니다. 이러한 정책을 사용하면 해당 자격 증명이 할 수 있는 일 및 권한을 지정할 수 있습니다. 자격 증명 기반 정책은 하위 범주로 분류할 수 있습니다.

관리형 정책 - 독립형 자격 증명 기반 정책으로, AWS 계정 내의 여러 사용자, 그룹, 역할에 연결할 수 있습니다. 관리형 정책에는 다음과 같이 2가지 유형이 있습니다.

- AWS 관리형 정책 - AWS에서 생성하고 관리하는 관리형 정책입니다.
- 고객 관리형 정책 - AWS 계정에서 생성하고 관리하는 관리형 정책입니다. 고객 관리형 정책은 AWS 관리형 정책에 비해 정책을 더 세세하게 제어할 수 있습니다.

관리형 정책은 권한 적용에 선호되는 방법입니다. 그러나 하나의 사용자, 그룹 또는 역할에 직접 추가하는 인라인 정책을 사용할 수도 있습니다. 인라인 정책은 정책과 자격 증명 사이에 엄격한 1대1 관계를 유지합니다. 자격 증명을 삭제하면 인라인 정책도 삭제됩니다.

대부분의 경우 최소 권한 원칙에 따라 자체적으로 고객 관리형 정책을 생성해야 합니다.

리소스 기반 정책은 리소스에 연결됩니다. 리소스 기반 정책의 예로는 S3 버킷 정책이 있습니다. 이 정책은 리소스와 같거나 다른 계정에 있을 수 있는 보안 주체에 권한을 부여합니다. 리소스 기반 정책을 지원하는 서비스 목록을 확인하려면 IAM과 함께 작동하는 AWS 서비스를 참조하세요.

권한 경계는 관리형 정책을 사용하여 관리자가 설정할 수 있는 최대 권한을 지정합니다. 이렇게 하면 IAM 역할 생성과 같은 권한을 생성하고 관리할 수 있는 능력을 개발자에게 위임할 수 있지만, 개발자가 생성한 권한을 사용하여 권한을 에스컬레이션할 수 없도록 제한할 수 있습니다.

속성 기반 액세스 제어(ABAC)를 사용하면 속성을 기반으로 권한을 부여할 수 있습니다. AWS에서는 이를 태그라고 합니다. 태그는 IAM 보안 주체(사용자 또는 역할) 및 AWS 리소스에 연결될 수 있습니다. 관리자는 IAM 정책을 사용하여 IAM 보안 주체의 속성을 기반으로 권한을 적용하는, 재사용 가능한 정책을 만들 수 있습니다. 예를 들어, 관리자는 조직의 개발자에게 개발자의 프로젝트 태그와 일치하는 AWS 리소스에 대한 액세스 권한을 부여하는 단일 IAM 정책을 사용할 수 있습니다. 개발자 팀이 프로젝트에 리소스를 추가하면 속성에 따라 권한이 자동으로 적용됩니다. 따라서 새 리소스가 생길 때마다 정책을 업데이트하지 않아도 됩니다.

조직 서비스 통제 정책(SCP)을 사용하여 조직 또는 조직 단위(OU)의 계정 멤버에 대한 최대 권한을 정의합니다. SCP는 자격 증명 기반 정책 또는 리소스 기반 정책이 계정 내의 엔터티(사용자 또는 역할)에 부여하는 권한을 제한할 뿐 권한을 부여하지는 않습니다.

세션 정책은 역할 또는 페더레이션 사용자를 수임합니다. AWS CLI 또는 AWS API를 사용할 때 세션 정책을 전달합니다. 세션 정책은 역할 또는 사용자의 자격 증명 기반 정책이 세션에 부여하는 권한을 제한합니다. 이러한 정책은 생성된 세션에 대한 권한을 제한할 뿐 권한을 부여하지는 않습니다. 자세한 내용은 세션 정책을 참조하세요.

모범 사례

- SEC03-BP01 액세스 요구 사항 정의
- SEC03-BP02 최소 권한 액세스 부여
- SEC03-BP03 긴급 액세스 프로세스 설정
- SEC03-BP04 지속적으로 권한 축소
- SEC03-BP05 조직에 대한 권한 가드레일 정의
- SEC03-BP06 수명 주기에 따라 액세스 관리
- SEC03-BP07 퍼블릭 및 크로스 계정 액세스 분석
- SEC03-BP08 안전하게 조직과 리소스 공유
- SEC03-BP09 안전하게 제3자와 리소스 공유

SEC03-BP01 액세스 요구 사항 정의

관리자, 최종 사용자 또는 기타 구성 요소는 워크로드의 각 구성 요소 또는 리소스에 액세스해야 합니다. 누가 혹은 무엇이 각 구성 요소에 액세스할 수 있는지 명확하게 정의한 다음 적절한 자격 증명 유형과 인증 및 권한 부여 방법을 선택해야 합니다.

일반적인 안티 패턴:

- 애플리케이션에 보안 암호를 하드 코딩 또는 저장합니다.
- 각 사용자에 대한 사용자 지정 권한을 부여합니다.
- 수명이 긴 보안 인증을 사용합니다.

이 모범 사례가 확립되지 않을 경우 노출되는 위험의 수준: 높음

구현 가이드

관리자, 최종 사용자 또는 기타 구성 요소는 워크로드의 각 구성 요소 또는 리소스에 액세스해야 합니다. 누가 혹은 무엇이 각 구성 요소에 액세스할 수 있는지 명확하게 정의한 다음 적절한 자격 증명 유형과 인증 및 권한 부여 방법을 선택해야 합니다.

조직 내 AWS 계정에 대한 정기적인 액세스를 제공하려면 페더레이션 액세스 또는 중앙 집중식 자격 증명 공급자를 사용해야 합니다. 또한 자격 증명 관리를 중앙 집중화하고, 직원 액세스 수명 주기에 대한 AWS 액세스를 통합하기 위해 정립된 사례가 있는지 확인해야 합니다. 예를 들어, 직원이 다른 액세스 수준의 직무로 변경할 경우 해당 그룹 멤버십 또한 변경하여 새로운 액세스 요구 사항을 반영해야 합니다.

비 인적 자격 증명에 대한 액세스 요구 사항을 정의할 경우 어떤 애플리케이션 및 구성 요소가 액세스해야 하는지, 그리고 어떻게 권한이 부여되는지 결정합니다. 권장되는 접근 방식은 최소 권한 액세스 모델을 통해 구축된 IAM 역할을 사용하는 것입니다. AWS 관리형 정책은 대부분의 일반적인 사용 사례를 다루는 사전 정의된 IAM 정책을 제공합니다.

AWS 서비스, 즉 AWS Secrets Manager 및 AWS Systems Manager Parameter Store는IAM 역할 사용이 실현 불가능한 경우 애플리케이션 또는 워크로드로부터 보안 암호를 안전하게 분리할 수 있도록 지원합니다. Secrets Manager에서 보안 인증에 대한 자동 교체를 설정할 수 있습니다. Systems Manager를 사용하여 스크립트, 명령, SSM 문서, 구성 및 자동화 워크플로의 파라미터를 참조할 수 있으며 이 경우 파라미터를 생성할 때 지정한 고유한 이름을 사용합니다.

AWS Identity and Access Management Roles Anywhere를 사용하여 IAM 외부에서 실행되는 워크로드에 대한 AWS의 임시 보안 자격 증명을 얻을 수 있습니다. 워크로드에서 사용하는 IAM 정책 및 IAM 역할은 AWS 애플리케이션에서 AWS 리소스에 액세스하는 데 사용하는 것과 동일한 것일 수 있습니다.

가능한 경우, 장기적이고 정적인 보안 인증보다는 단기적이고 임시적인 보안 인증을 사용하는 것이 좋습니다. 프로그래밍 방식 액세스 및 장기 보안 인증을 사용하는 IAM 사용자가 필요한 경우 마지막으로 사용된 액세스 키 정보를 사용하여 액세스 키를 교체 및 제거합니다.

리소스

관련 문서:

- 속성 기반 액세스 제어(ABAC)
- AWS IAM Identity Center
- IAM Roles Anywhere
- IAM Identity Center의 AWS 관리형 정책
- AWS IAM 정책 조건
- IAM 사용 사례
- 불필요한 보안 인증 제거

- 정책 사용
- AWS 계정, OU 또는 조직을 기준으로 AWS 리소스에 대한 액세스를 제어하는 방법
- AWS Secrets Manager의 향상된 검색을 사용하여 보안 암호를 쉽게 식별, 정렬 및 관리

관련 동영상:

- 60분 이내에 IAM 정책 마스터하기
- 업무 분리, 최소 권한, 위임 및 CI/CD
- 혁신을 위한 자격 증명 및 액세스 관리 간소화

SEC03-BP02 최소 권한 액세스 부여

구체적인 조건에서 특정 리소스에 대해 일정한 작업을 수행하기 위해 자격 증명이 필요한 액세스 권한만 부여하는 것이 좋습니다. 개별 사용자에 대한 권한을 정의하는 대신, 그룹 및 자격 증명 속성을 사용하여 대규모로 권한을 동적으로 설정합니다. 예를 들어 한 개발자 그룹에 자체 프로젝트에 대한 리소스만 관리하도록 액세스 권한을 허용할 수 있습니다. 이렇게 하면 특정 개발자가 프로젝트에서 빠지게 될 경우 기본 액세스 정책을 변경하지 않고도 해당 개발자의 액세스 권한이 자동으로 해지됩니다.

원하는 결과: 사용자는 작업을 수행하는 데 필요한 권한만 가지고 있어야 합니다. 사용자는 제한된 시간 안에 특정 작업을 수행할 수 있는 프로덕션 환경에 액세스할 권한만 부여받아야 하며, 해당 작업이 완료된 후에는 액세스 권한이 해지되어야 합니다. 사용자가 다른 프로젝트에 착수하거나 직무를 옮기는 등 액세스 권한이 더 이상 필요하지 않으면 권한은 해지되어야 합니다. 관리자 권한은 신뢰할 수 있는 소수의 관리자 그룹에만 주어져야 합니다. 권한을 정기적으로 검토하여 권한이 잘못 부여되어 있는 상황이 없도록 해야 합니다. 시스템 또는 시스템 계정에는 작업을 완료하는 데 필요한 최소 권한 집합이 부여되어야 합니다.

일반적인 안티 패턴:

- 사용자에게 기본적으로 관리자 권한을 부여합니다.
- 일상적인 활동에 루트 사용자를 이용합니다.
- 전체 관리자 권한은 아니지만 과도하게 허용적인 정책을 생성합니다.
- 최소 권한 액세스를 허용하는지를 확인하기 위해 권한을 검토하지 않습니다.

이 모범 사례를 따르지 않을 경우 노출되는 위험의 수준: 높음

구현 가이드

최소 권한의 원칙은 자격 증명을 부여할 때 특정 작업을 완수하는 데 필요한 최소한의 활동만 수행하도록 부여해야 한다고 규정합니다. 이를 통해 사용 편의성, 효율성 및 보안의 균형을 이룰 수 있습니다. 이 원칙에 따라 운영하면 의도하지 않은 액세스를 제한하고 누가 어떤 리소스에 액세스했는지 추적하는 데 도움이 됩니다. IAM 사용자 및 역할에는 기본적으로 권한이 없습니다. 루트 사용자는 기본적으로 전체 액세스 권한을 가지므로, 루트 사용자는 루트 액세스가 필요한 작업에만 엄격하게 제어, 모니터링 및 사용되어야 합니다.

IAM 정책은 IAM 역할 또는 특정 리소스에 대한 권한을 명시적으로 부여하는 데 사용됩니다. 예를 들어, 자격 증명 기반 정책은 IAM 그룹에 연결하는 한편 S3 버킷은 리소스 기반 정책으로 제어할 수 있습니다.

IAM 정책을 생성할 때 AWS에서 액세스를 허용하거나 거부하려면 '참'이어야 하는 서비스 작업, 리소스 및 조건을 지정할 수 있습니다. AWS에서는 액세스 범위를 줄일 수 있도록 다양한 조건을 지원합니다. 예를 들어, `PrincipalOrgID` 조건 키를 사용하면 요청자가 AWS 조직에 속하지 않는 경우 요청자의 작업을 거부할 수 있습니다.

또한 `CalledVia` 조건 키를 사용하여 AWS Lambda 함수를 생성하는 AWS CloudFormation(와)과 같이, AWS 서비스가 사용자를 대신하여 수행하는 요청을 제어할 수 있습니다. 다양한 정책 유형을 계층화하여 심층 방어를 설정하고 사용자의 권한 전반을 제한해야 합니다. 나아가 어떤 조건에서 어떤 권한을 허용할지도 제한할 수도 있습니다. 예를 들어 애플리케이션 팀이 구축하는 시스템에 대해 자체 IAM 정책을 만들도록 허용하되, 권한 경계를 적용하여 시스템이 수신할 수 있는 최대 권한을 제한할 수 있습니다.

구현 단계

- 최소 권한 정책 구현: IAM 그룹 및 역할에 최소 권한 액세스 정책을 적용하여 사용자별로 정의한 역할 또는 기능을 반영합니다.
  - API 사용에 대한 기본 정책: AWS CloudTrail 로그를 검토하여 필요한 권한을 결정합니다. 이 검토를 통해 사용자가 AWS 내에서 실제로 수행하는 작업에 맞게 조정된 권한을 만들 수 있습니다. IAM Access Analyzer는 활동을 기반으로 IAM 정책을 자동으로 생성할 수 있습니다. 조직 또는 계정 수준에서 IAM Access Advisor를 사용하여 특정 정책에 대해 마지막으로 액세스한 정보를 추적할 수 있습니다.
- 직무 역할에 따른 AWS 관리형 정책 사용을 고려해 보세요. 세분화된 권한 정책을 어디서부터 만들어야 할지 알기 어려울 수 있습니다. AWS에서는 결제 관리자, 데이터베이스 관리자 및 데이터 사이언티스트와 같은 일반적인 작업 역할에 대한 관리형 정책을 관리합니다. 이러한 정책은 최소 권한 정책을 구현하는 방법을 결정하는 동시에 사용자의 액세스 범위를 좁히는 데 도움이 될 수 있습니다.

- 불필요한 권한 제거: 필요하지 않은 권한을 제거하고 과도하게 허용적인 정책을 축소합니다. IAM Access Analyzer 정책 생성을 사용하면 권한 정책을 세밀하게 조정할 수 있습니다.
- 사용자의 프로덕션 환경 액세스 제한: 사용자는 유효한 사용 사례가 있는 프로덕션 환경에만 액세스할 수 있어야 합니다. 사용자가 프로덕션 액세스 권한이 필요한 특정 작업을 수행한 후에는 액세스 권한을 해지해야 합니다. 프로덕션 환경에 대한 액세스를 제한하면 예기치 않게 프로덕션에 영향을 미치는 이벤트를 방지하고 의도하지 않은 액세스의 영향 범위를 줄일 수 있습니다.
- 권한 경계 고려: 권한 경계는 자격 증명 기반 정책을 통해 IAM 엔터티에 부여할 수 있는 최대 권한을 설정하는 관리형 정책을 사용하는 기능입니다. 엔터티의 권한 경계는 자격 증명 기반 정책과 권한 경계 모두에서 허용되는 작업만 수행하도록 허용합니다.
- 권한 리소스 태그 고려: 리소스 태그를 사용하는 속성 기반 액세스 제어 모델을 활용하면 리소스 목적, 소유자, 환경 또는 기타 기준에 따라 액세스 권한을 부여할 수 있습니다. 예를 들어, 리소스 태그를 사용하여 개발 환경과 프로덕션 환경을 구분할 수 있습니다. 이러한 태그를 사용하여 개발자의 권한을 개발 환경으로 제한할 수 있습니다. 태그 지정 정책과 권한 정책을 결합하면 모든 직무에 대해 복잡한 사용자 지정 정책을 정의할 필요 없이 세분화된 리소스 액세스 제어가 가능합니다.
- AWS Organizations 서비스 제어 정책을 사용해 보세요. 서비스 제어 정책은 조직의 멤버 계정에 대해 사용 가능한 최대 권한을 중앙에서 제어합니다. 중요한 점은 서비스 제어 정책을 사용하여 멤버 계정의 루트 사용자 권한을 제한할 수 있다는 사실입니다. AWS Organizations(을)를 보다 풍부하게 활용할 수 있도록 권장 관리 제어 기능을 제공하는 AWS Control Tower(을)를 사용하는 것도 고려해 보세요. Control Tower 내에서 자체 제어 기능을 정의할 수도 있습니다.
- 조직의 사용자 수명 주기 정책 설정: 사용자 수명 주기 정책은 사용자가 AWS에 온보드되어 있을 때, 작업 역할 또는 범위를 변경했을 때, 또는 AWS에 더 이상 액세스할 필요가 없을 때 각각 수행할 작업을 정의합니다. 사용자 수명 주기의 각 단계에서 권한 검토를 수행하여 권한이 적절하게 제한되는지 확인하고 권한이 잘못 부여되어 있는 상황이 없도록 해야 합니다.
- 권한을 검토하고 불필요한 권한을 제거하기 위한 정기 예약 설정: 사용자 액세스를 정기적으로 검토하여 사용자에게 과도하게 허용되는 액세스 권한이 없는지 확인해야 합니다. AWS Config 및 IAM Access Analyzer는 사용자 권한을 감사할 때 유용합니다.
- 작업 역할 매트릭스 설정: 작업 역할 매트릭스는 AWS 기반 내에서 필요한 여러 역할 및 액세스 수준을 시각화합니다. 작업 역할 매트릭스를 바탕으로 조직 내 사용자 책임에 따라 권한을 정의하고 분리할 수 있습니다. 개별 사용자 또는 역할에 직접 권한을 적용하는 대신 그룹을 사용하세요.

리소스

관련 문서:

- 최소 권한 부여

- IAM 엔터티의 권한 경계
- 최소 권한 IAM 정책 작성 기법
- IAM Access Analyzer를 통해 액세스 활동에 기반한 IAM 정책을 생성하여 보다 쉽게 최소 권한을 구현
- IAM 권한 경계를 사용하여 개발자에게 권한 관리 위임
- 마지막 액세스 정보를 사용하여 권한 수정
- IAM 정책 유형과 사용 시기
- IAM 정책 시뮬레이터로 IAM 정책 테스트
- AWS Control Tower의 가드레일
- 제로 트러스트 아키텍처: AWS의 철학
- CloudFormation StackSets로 최소 권한의 원칙을 구현하는 방법
- 속성 기반 액세스 제어(ABAC)
- 사용자 활동 보기를 통한 정책 범위 축소
- 역할 액세스 보기
- 태그를 지정하여 환경 구성 및 책임 추진
- AWS 태그 지정 전략
- AWS 리소스 태그 지정

관련 동영상:

- 차세대 권한 관리
- 제로 트러스트: AWS의 철학
- 권한 경계를 사용하여 사용자 및 역할을 제한하고 권한 에스컬레이션을 방지하려면 어떻게 해야 하나요?

관련 예시:

- 실습: 역할 생성을 위임하는 IAM 권한 경계
- 실습: EC2에 대한 IAM 태그 기반 액세스 제어

SEC03-BP03 긴급 액세스 프로세스 설정

중앙 집중식 ID 공급자에 문제가 발생할 경우 예상치 못한 상황에서 워크로드에 긴급 액세스할 수 있는 프로세스를 만드세요.

비상 상황을 초래할 수 있는 다양한 장애 모드에 대한 프로세스를 설계해야 합니다. 예를 들어, 일반적인 상황에서는 직원 사용자가 중앙 집중식 ID 공급자를 사용하여 클라우드로 페더레이션합니다(SEC02-BP04)를 사용하여 워크로드를 관리합니다. 그러나 중앙 집중식 ID 공급자에 장애가 발생하거나 클라우드에서의 페더레이션 구성이 수정되면 직원 사용자가 클라우드로 페더레이션하지 못할 수 있습니다. 긴급 액세스 프로세스를 통해 권한 있는 관리자는 대체 수단(예: 대체 형태의 페더레이션 또는 직접 사용자 액세스)을 통해 클라우드 리소스에 액세스하여 페더레이션 구성 또는 워크로드 관련 문제를 해결할 수 있습니다. 비상 액세스 프로세스는 일반 페더레이션 메커니즘이 복원될 때까지 사용됩니다.

원하는 결과:

- 비상 상황으로 간주되는 장애 모드를 정의하고 문서화했습니다. 일반적인 상황과 사용자가 워크로드를 관리하기 위해 사용하는 시스템을 고려하세요. 이러한 각 종속성이 어떻게 실패하여 긴급 상황을 초래할 수 있는지 생각해 보세요. 질문과 모범 사례는 안정성 원칙에서 찾을 수 있습니다. 장애 모드를 식별하고 장애 가능성을 최소화하기 위해 보다 탄력적인 시스템을 설계하는 데 유용합니다.
- 장애를 긴급 상황으로 확인하기 위해 따라야 하는 단계를 문서화했습니다. 예를 들어 ID 관리자에게 기본 및 대기 ID 제공자의 상태를 확인하고 둘 다 사용할 수 없는 경우 ID 제공자 장애에 대한 긴급 이벤트를 선언하도록 요청할 수 있습니다.
- 각 유형의 비상 또는 장애 모드에 맞는 긴급 액세스 프로세스를 정의했습니다. 구체적으로 설명하면 모든 유형의 긴급 상황에서 일반 프로세스를 과도하게 사용하려는 사용자의 유혹을 줄일 수 있습니다. 긴급 액세스 프로세스는 각 프로세스를 사용해야 하는 상황과 반대로 프로세스를 사용하지 않아야 하는 상황을 설명하고 적용될 수 있는 대체 프로세스를 가리킵니다.
- 프로세스는 빠르고 효율적으로 따를 수 있는 상세한 지침과 플레이북과 함께 잘 문서화되어 있습니다. 긴급 상황은 사용자에게 스트레스를 주는 시간이 될 수 있고 사용자들이 극심한 시간 압박을 받을 수 있다는 점을 기억하세요. 따라서 프로세스를 최대한 단순하게 설계하세요.

일반적인 안티 패턴:

- 비상 액세스 절차가 제대로 문서화되고 테스트되지 않습니다. 사용자는 비상 상황에 대비하지 않고 긴급 상황 발생 시 즉흥적인 프로세스를 따릅니다.

- 긴급 액세스 프로세스는 일반 액세스 메커니즘과 동일한 시스템(예: 중앙 집중식 ID 공급자)을 기반으로 합니다. 즉, 이러한 시스템에 장애가 발생하면 정상 액세스 메커니즘과 긴급 액세스 메커니즘에 모두 영향을 미치고 장애 복구 능력이 저하될 수 있습니다.
- 비상 액세스 프로세스는 비응급 상황에서 사용됩니다. 예를 들어, 사용자는 파이프라인을 통해 변경 사항을 제출하는 것보다 직접 변경하는 것이 더 쉽기 때문에 긴급 액세스 프로세스를 자주 오용합니다.
- 긴급 액세스 프로세스는 프로세스를 감사하기에 충분한 로그를 생성하지 않거나 프로세스의 오용 가능성에 대해 경고할 수 있도록 로그를 모니터링하지 않습니다.

이 모범 사례 확립의 이점:

- 잘 문서화되고 테스트를 거친 긴급 액세스 프로세스를 갖추면 사용자가 긴급 상황에 대응하고 해결하는 데 걸리는 시간을 줄일 수 있습니다. 이를 통해 가동 중지 시간이 줄어들고 고객에게 제공하는 서비스의 가용성이 향상될 수 있습니다.
- 각 긴급 액세스 요청을 추적하고, 프로세스를 긴급 상황이 아닌 이벤트에 악용하려는 무단 시도를 감지하고 경고를 보낼 수 있습니다.

이 모범 사례가 확립되지 않을 경우 노출되는 위험 수준:보통

구현 가이드

이 섹션에서는 모든 장애 모드에 적용되는 공통 지침부터 시작하여 장애 모드 유형에 따른 구체적인 지침에 이어 AWS에 배포된 워크로드와 관련된 여러 장애 모드에 대한 비상 액세스 프로세스를 만드는 방법에 대한 지침을 제공합니다.

모든 장애 모드에 대한 공통 지침

장애 모드에 대한 긴급 액세스 프로세스를 설계할 때는 다음 사항을 고려하세요.

- 프로세스의 사전 조건과 전제 조건, 즉 프로세스를 사용해야 하는 시기와 사용하지 말아야 하는 경우를 문서화하세요. 장애 모드를 자세히 설명하고 다른 관련 시스템의 상태와 같은 가정을 문서화하는 데 도움이 됩니다. 예를 들어 장애 모드 2의 프로세스에서는 ID 제공자를 사용할 수 있지만 설정된 AWS 구성이 수정되었거나 만료된 것으로 가정합니다.
- 긴급 액세스 프로세스에 필요한 리소스를 사전에 생성합니다(SEC10-BP05). 예를 들어, 모든 워크로드 계정에서 IAM users 및 AWS 계정 역할과 계정 간 IAM 역할을 사용하여 긴급 액세스를 미리 생성하세요. 이를 통해 긴급 상황 발생 시 이러한 리소스가 준비되어 있고 사용할 수 있는지 확인할 수 있습니다. 리소스를 미리 만들면 긴급 상황에서 사용할 수 없는 AWS 컨트롤 플레인 API(AWS 리소스

생성 및 수정에 사용됨)에 대한 종속성이 없습니다. 또한 IAM 리소스를 미리 생성하면 최종 일관성으로 인한 잠재적 지연을 고려할 필요가 없습니다.

- 사고 관리 계획의 일부로 비상 액세스 프로세스를 포함하세요(SEC10-BP02). 비상 상황이 어떻게 추적되고 동료 팀, 경영진, 그리고 해당하는 경우 외부 고객 및 비즈니스 파트너와 같은 조직 내 다른 사람에게 전달되는지 문서화하세요.
- 기존 서비스 요청 워크플로 시스템 (있는 경우) 에서 긴급 액세스 요청 프로세스를 정의하세요. 일반적으로 이러한 워크플로우 시스템에서는 접수 양식을 만들어 요청에 대한 정보를 수집하고, 워크플로의 각 단계를 통해 요청을 추적하고, 자동 승인 단계와 수동 승인 단계를 모두 추가할 수 있습니다. 각 요청을 사고 관리 시스템에서 추적되는 해당 비상 이벤트와 연관시키세요. 비상 액세스를 위한 통일된 시스템을 구축하면 단일 시스템에서 이러한 요청을 추적하고, 사용 추세를 분석하고, 프로세스를 개선할 수 있습니다.
- 긴급 액세스 프로세스는 승인된 사용자만 시작할 수 있고 필요에 따라 사용자의 동료 또는 경영진의 승인이 필요한지 확인하세요. 승인 절차는 업무 시간 내외에서 모두 효과적으로 운영되어야 합니다. 1차 승인자를 사용할 수 없고 승인될 때까지 관리망에 에스컬레이션되는 경우 승인 요청을 2차 승인자가 허용할 수 있는 방법을 정의합니다.
- 프로세스에서 긴급 액세스 시도 성공 및 실패 모두에 대한 자세한 감사 로그 및 이벤트를 생성하는지 확인합니다. 요청 프로세스와 긴급 액세스 메커니즘을 모두 모니터링하여 오용 또는 무단 액세스를 탐지합니다. 활동을 사고 관리 시스템의 진행 중인 비상 이벤트와 연관시키고 예상 시간 외에 조치가 발생할 경우 경보를 보냅니다. 예를 들어 정상 운영 환경에서는 절대 사용해서는 안 되므로 비상 액세스 AWS 계정 활동을 모니터링하고 경고해야 합니다.
- 긴급 액세스 프로세스를 정기적으로 테스트하여 단계가 명확한지 확인하고 올바른 액세스 수준을 빠르고 효율적으로 부여하세요. 비상 액세스 프로세스는 사고 대응 시뮬레이션의 일부로 테스트해야 합니다(SEC10-BP07) 및 재해 복구 테스트 (REL13-BP03).

장애 모드 1: AWS으로 페더레이션에 사용된 ID 제공자를 사용할 수 없음

다음과 설명된 대로 SEC02-BP04, 중앙 집중식 ID 공급자를 통해 직원 사용자를 페데레이션하여 AWS 계정에 액세스 권한을 부여하는 것이 좋습니다. IAM을 사용하여 AWS 조직 AWS 계정 내 여러 곳에 페더레이션하거나 IAM Identity Center를 사용하여 개별적으로 AWS 계정 페더레이션할 수 있습니다. 두 경우 모두, 직원 사용자는 SSO(Single Sign-On)를 위해 AWS 로그인 엔드포인트로 리디렉션되기 전에 중앙 집중식 ID 공급자를 통해 인증합니다.

드문 경우지만 중앙 집중식 ID 공급자를 사용할 수 없는 경우 직원 사용자는 AWS 계정에 페더레이션하거나 워크로드를 관리할 수 없습니다. 이 긴급 상황에서 중앙 집중식 ID 제공자가 다시 온라인 상태가 될 때까지 기다릴 수 없는 중요한 작업을 수행할 수 AWS 계정 있도록 소수의 관리자가 액세스할 수 있는 긴급 액세스 프로세스를 제공할 수 있습니다. 예를 들어 ID 공급자를 4시간 동안 사용할 수 없는

경우, 예상치 못한 고객 트래픽 급증에 대처하려면 Production 계정의 Amazon EC2 Auto Scaling 그룹 상한선을 수정해야 합니다. 비상 관리자는 긴급 액세스 프로세스에 따라 특정 프로덕션 AWS 계정에 대한 액세스 권한을 얻고 필요한 사항을 변경해야 합니다.

긴급 액세스 프로세스는 긴급 액세스에만 사용되고 긴급 액세스 AWS 계정 프로세스를 지원하는 AWS 리소스(예: IAM 역할 및IAM users)가 있는 사전 생성된 긴급 액세스를 기반으로 합니다. 정상적으로 운영되는 동안에는 아무도 긴급 액세스 계정에 접속해서는 안 되며 이 계정의 오용을 모니터링하고 경고해야 합니다 (자세한 내용은 이전 공통 지침 섹션 참조).

긴급 액세스 계정에는 긴급 액세스가 필요한 계정 간 IAM 역할을 맡을 수 있는 권한이 AWS 계정 있는 긴급 액세스 역할이 있습니다. 이러한 IAM 역할은 긴급 계정의 IAM 역할을 신뢰하는 신뢰 정책으로 미리 생성되고 구성됩니다.

긴급 액세스 프로세스는 다음 방법 중 하나를 사용할 수 있습니다.

- 연결된 강력한 비밀번호 및 MFA 토큰을 사용하여 비상 액세스 계정에서 비상 관리자를 위한 IAM users 세트를 미리 생성할 수 있습니다. 이 IAM users은 IAM 역할을 맡아 긴급 액세스가 필요한 AWS 계정에 계정 간 액세스를 허용하는 권한을 갖게 됩니다. 가능한 한 적은 수의 사용자를 생성하고 각 사용자를 한 명의 비상 관리자에게 할당하는 것이 좋습니다. 긴급 상황 발생 시 긴급 관리자 사용자는 암호와 MFA 토큰 코드를 사용하여 긴급 액세스 계정에 로그인하고, 긴급 계정에서 긴급 액세스 IAM 역할로 전환하고, 마지막으로 워크로드 계정의 긴급 액세스 IAM 역할로 전환하여 긴급 변경 작업을 수행합니다. 이 접근 방식의 장점은 각 IAM user가 한 명의 비상 관리자에게 할당되며 CloudTrail 이벤트를 검토하여 어떤 사용자가 로그인했는지 알 수 있다는 것입니다. 단점은 수명이 긴 암호와 MFA 토큰을 사용하여 IAM users 여러 개를 유지 관리해야 한다는 것입니다.
- 긴급 액세스 루트 사용자를 사용하여 긴급 액세스 계정에 로그인하고, 긴급 액세스 IAM 역할을 맡고, 워크로드 계정에서 교차 계정 역할을 맡을 수 있습니다. 루트 사용자에게는 강력한 암호와 여러 MFA 토큰을 설정하는 것이 좋습니다. 또한 강력한 인증 및 권한 부여를 시행하는 안전한 엔터프라이즈 자격 증명 보관소에 암호와 MFA 토큰을 저장하는 것이 좋습니다. 암호 및 MFA 토큰 재설정 요소를 보호해야 합니다. 계정의 이메일 주소를 클라우드 보안 관리자가 모니터링하는 이메일 배포 목록으로 설정하고 계정의 전화번호를 보안 관리자가 모니터링하는 공유 전화번호로 설정합니다. 이 접근 방식의 장점은 한 세트의 루트 사용자 자격 증명을 관리할 수 있다는 것입니다. 단점은 공유 사용자이기 때문에 여러 관리자가 루트 사용자로 로그인할 수 있다는 것입니다. 엔터프라이즈 볼트 로그 이벤트를 감사하여 루트 사용자 암호를 체크아웃한 관리자를 식별해야 합니다.

장애 모드 2: AWS의 ID 제공자 구성이 수정되었거나 만료됨

인력 사용자가 페더레이션할 수 있도록 하려면 외부 ID 공급자를 사용하여 구성하거나 ID 제공자를 생성할 수 있습니다 (AWS 계정IAM Identity CenterIAMSEC02-BP04). 일반적으로 ID 공급자가 제공

한 SAML 메타데이터 XML 문서를 가져와서 이를 구성합니다. 메타데이터 XML 문서에는 ID 제공자가 SAML 어설션에 서명하는 데 사용하는 개인 키에 해당하는 X.509 인증서가 포함되어 있습니다.

AWS-side의 이러한 구성은 관리자가 실수로 수정하거나 삭제할 수 있습니다. 또 다른 시나리오에서는 AWS로 가져온 X.509 인증서가 만료되고 새 인증서가 포함된 새 메타데이터 XML을 AWS에 아직 가져오지 않은 경우가 있습니다. 두 시나리오 모두 인력 사용자에 대한 AWS 페더레이션을 중단하여 긴급 상황을 초래할 수 있습니다.

이러한 긴급 상황의 경우 ID 관리자에게 페더레이션 문제를 해결할 수 있는 AWS 액세스 권한을 제공할 수 있습니다. 예를 들어, ID 관리자는 긴급 액세스 프로세스를 사용하여 긴급 액세스에 로그인하고 AWS 계정, Identity Center 관리자 계정의 역할로 전환하고, 페더레이션을 다시 활성화하기 위해 ID 공급자로부터 최신 SAML 메타데이터 XML 문서를 가져와서 외부 ID 제공자 구성을 업데이트합니다. 페더레이션이 수정되면 인력 사용자는 계속해서 일반 운영 프로세스를 사용하여 워크로드 계정에 페더레이션합니다.

이전 장애 모드 1에 설명된 접근 방식에 따라 긴급 액세스 프로세스를 만들 수 있습니다. ID 관리자에게 Identity Center 관리자 계정에만 액세스하고 해당 계정의 Identity Center에서 작업을 수행할 수 있는 최소 권한 권한을 부여할 수 있습니다.

장애 모드 3: ID 센터 중단

예상치 못한 IAM Identity Center 상황이나 AWS 리전 중단이 발생할 경우 AWS Management Console에 임시 액세스를 제공하는 데 사용할 수 있는 구성을 설정하는 것이 좋습니다.

긴급 액세스 프로세스는 ID 공급자로부터 긴급 계정으로의 IAM 직접 페더레이션을 사용합니다. 프로세스 및 설계 고려 사항에 대한 자세한 내용은 을 참조하세요. AWS Management Console에 대한 비상 액세스 설정.

구현 단계

모든 장애 모드의 공통 단계

- 비상 액세스 프로세스 AWS 계정 전용 프로세스를 생성합니다. 계정에 필요한 IAM 리소스(예: IAM 역할 또는 IAM users IAM ID 공급자)를 미리 생성합니다. 또한 긴급 액세스 계정의 해당 IAM AWS 계정 IAM 역할과의 신뢰 관계를 사용하여 워크로드에 계정 간 역할을 미리 생성하세요. 이때 AWS Organizations와 AWS CloudFormation StackSets으로 조직의 구성원 계정에 이러한 리소스를 만들 수 있습니다.

- 서비스 제어 정책 AWS Organizations 생성 (SCP) - AWS 계정의 구성원의 교차 계정 IAM 역할 삭제 및 수정을 거부합니다.

- 비상 액세스를 CloudTrail AWS 계정 활성화하고 로그 컬렉션 AWS 계정의 중앙 S3 버킷으로 트레일 이벤트를 전송하세요. AWS 계정를 AWS Control Tower 사용하여 AWS 다중 계정 환경을 설정하고 관리하는 경우 AWS Control Tower 사용하거나 등록한 모든 계정이 기본적으로 AWS Control Tower CloudTrail 활성화되어 전용 로그 아카이브의 S3 버킷으로 전송됩니다.
- 비상 IAM 역할별로 콘솔 로그인 및 API 활동과 일치하는 EventBridge 규칙을 생성하여 긴급 액세스 계정의 활동을 모니터링합니다. 사고 관리 시스템에서 추적되는 진행 중인 비상 이벤트 외부에서 활동이 발생하는 경우 보안 운영 센터에 알림을 보냅니다.

장애 모드 1에 대한 추가 단계: 페더레이션에 사용된 ID 제공자를 사용할 수 없으며 장애 모드 2: ID 제공자 AWS 구성이 수정되었거나 만료됨 AWS

- 긴급 액세스를 위해 선택한 메커니즘에 따라 리소스를 미리 생성하세요.
  - IAM users 사용 강력한 암호 및 관련 MFA 디바이스를 사용하여 IAM users를 미리 생성하세요.
  - 긴급 계정 루트 사용자 사용: 강력한 암호로 루트 사용자를 구성하고 엔터프라이즈 자격 증명 저장소에 암호를 저장합니다. 여러 물리적 MFA 디바이스를 루트 사용자와 연결하고 비상 관리자 팀 구성원이 빠르게 액세스할 수 있는 위치에 디바이스를 저장합니다.

장애 모드 3에 대한 추가 단계: ID 센터 중단

- AWS Management Console에 대한 비상 액세스 설정 에 자세히 설명된 대로긴급 AWS 계정 액세스에서 ID 공급자를 생성하여 IAM ID 공급자로부터 직접 SAML 페더레이션을 활성화하세요.
- IdP에 구성원 없이 비상 운영 그룹을 만드세요.
- 긴급 액세스 계정에서 비상 운영 그룹에 해당하는 IAM 역할을 생성합니다.

리소스

관련 Well-Architected 모범 사례:

- SEC02-BP04 중앙 집중식 자격 증명 공급자 사용
- SEC03-BP02 최소 권한 액세스 부여
- SEC10-BP02 인시던트 관리 계획 개발
- SEC10-BP07 게임 데이 진행

관련 문서:

- AWS Management Console에 대한 비상 액세스 설정
- SAML 2.0 페더레이션 사용자가 AWS Management Console에 액세스할 수 있도록 지원
- 브레이크 글라스 액세스

관련 동영상:

- AWS re:Invent 2022 - Simplify your existing workforce access with IAM Identity Center
- AWS re:Inforce 2022 - AWS Identity and Access Management (IAM) deep dive

관련 예시:

- AWS브레이크 글래스 역할
- AWS 고객 플레이북 프레임워크
- AWS 인시던트 대응 플레이북 샘플

SEC03-BP04 지속적으로 권한 축소

팀에서 필요한 액세스 권한을 결정할 때 불필요한 권한을 제거하고 최소 권한을 부여하기 위한 검토 프로세스를 수립합니다. 인적 액세스와 시스템 액세스 모두에 대해 사용되지 않는 자격 증명과 권한을 지속적으로 모니터링하고 제거합니다.

원하는 결과: 권한 정책은 최소 권한 원칙을 준수해야 합니다. 직무와 역할이 더 잘 정의됨에 따라 권한 정책을 검토하여 불필요한 권한을 제거해야 합니다. 이 접근 방식은 보안 인증 정보가 의도치 않게 노출되거나 권한 부여 없이 액세스되는 경우 영향 범위를 줄입니다.

일반적인 안티 패턴:

- 기본적으로 사용자에게 관리자 권한을 부여합니다.
- 지나치게 관대하지만 전체 관리자 권한이 없는 정책을 생성합니다.
- 더 이상 필요하지 않은 권한 정책을 유지합니다.

이 모범 사례를 따르지 않을 경우 노출 위험도: 중간

구현 가이드

팀과 프로젝트가 이제 막 시작되었으므로 허용 권한 정책을 사용하여 혁신과 민첩성을 확보할 수 있습니다. 예를 들어 개발 또는 테스트 환경에서 개발자에게 광범위한 AWS 서비스에 대한 액세스 권한을

부여할 수 있습니다. 액세스 권한을 지속적으로 평가하고 현재 작업을 완료하는 데 필요한 서비스 및 서비스 작업으로만 액세스 권한을 제한하는 것이 좋습니다. 인적 자격 증명과 시스템 자격 증명 모두에 대해 이 평가가 권장됩니다. 시스템 또는 서비스 계정이라고도 하는 시스템 자격 증명은 AWS에 애플리케이션 또는 서버에 대한 액세스 권한을 부여하는 자격 증명입니다. 지나친 허용 권한은 광범위한 영향을 미치고 잠재적으로 고객 데이터를 노출시킬 수 있으므로 이 액세스 권한은 프로덕션 환경에서 특히 중요합니다.

AWS는 사용되지 않는 사용자, 역할, 권한 및 보안 인증 정보를 식별하는 데 도움이 되는 여러 방법을 제공합니다. AWS는 또한 연결된 액세스 키와 Amazon S3 버킷의 객체와 같은 AWS 리소스에 대한 액세스 권한을 포함하여 IAM 사용자 및 역할의 액세스 활동을 분석하는 데 도움이 될 수 있습니다. AWS Identity and Access Management Access Analyzer 정책 생성은 보안 주체가 상호 작용하는 실제 서비스 및 작업을 기반으로 제한적 권한 정책을 생성하는 데 도움이 될 수 있습니다. 속성 기반 액세스 제어(ABAC)는 권한 정책을 각 사용자에게 직접 연결하는 대신 속성을 사용하여 사용자에게 권한을 제공할 수 있으므로 권한 관리를 간소화하는 데 도움이 됩니다.

구현 단계

- AWS Identity and Access Management Access Analyzer 사용: IAM Access Analyzer는 조직 및 계정에서 Amazon Simple Storage Service(Amazon S3) 버킷 또는 IAM 역할과 같이 외부 엔터티와 공유되는 리소스를 식별하는 데 도움이 됩니다.
- IAM Access Analyzer 정책 생성 사용: IAM Access Analyzer 정책 생성은 IAM 사용자 또는 역할의 액세스 활동을 기반으로 세분화된 권한 정책을 생성하는 데 도움이 됩니다.
- IAM 사용자 및 역할에 대해 허용되는 기간 및 사용 정책 결정: 마지막으로 액세스한 타임스탬프를 사용하여 사용되지 않는 사용자 및 역할을 식별하고 제거합니다. 마지막으로 액세스한 서비스 및 작업 정보를 검토하여 특정 사용자 및 역할에 대한 권한을 식별하고 범위를 지정합니다. 예를 들어 마지막으로 액세스한 정보를 사용하면 애플리케이션 역할에 필요한 특정 Amazon S3 작업을 식별하여 그러한 작업으로만 역할의 액세스 권한을 제한할 수 있습니다. 마지막으로 액세스한 정보 기능은 AWS Management Console에서 프로그램 방식으로 제공되므로 인프라 워크플로 및 자동화된 도구에 손쉽게 통합할 수 있습니다.
- AWS CloudTrail에서 데이터 이벤트 로깅 고려: 기본적으로 CloudTrail은 Amazon S3 객체 수준 활동(예: `GetObject` 및 `DeleteObject`) 또는 Amazon DynamoDB 테이블 활동(예: `PutItem` 및 `DeleteItem`)과 같은 데이터 이벤트를 로깅하지 않습니다. 특정 Amazon S3 객체 또는 DynamoDB 테이블 항목에 액세스해야 하는 사용자 및 역할을 결정하려면 이러한 이벤트에 대한 로깅을 활성화하는 것이 좋습니다.

리소스

관련 문서:

- 최소 권한 부여
- 불필요한 보안 인증 정보 삭제
- AWS CloudTrail란 무엇인가요?
- 정책 사용
- DynamoDB 로깅 및 모니터링
- Amazon S3 버킷 및 객체에 대해 CloudTrail 이벤트 로깅 활성화
- AWS 계정의 보안 인증 정보 보고서 가져오기

관련 동영상:

- Become an IAM Policy Master in 60 Minutes or Less(60분 이내에 IAM 정책 마스터하기)
- Separation of Duties, Least Privilege, Delegation, and CI/CD(업무 분리, 최소 권한, 위임 및 CI/CD)
- AWS re:Inforce 2022 - AWS Identity and Access Management (IAM) deep dive(AWS Identity and Access Management(IAM) 심층 분석)

SEC03-BP05 조직에 대한 권한 가드레일 정의

조직의 모든 자격 증명에 대한 액세스를 제한하는 공통 제어를 설정합니다. 예를 들어, 특정 AWS 리전에 대한 액세스를 제한하거나 중앙 보안 팀에 사용되는 IAM 역할과 같은 공통 리소스를 운영자가 삭제하지 못하게 할 수 있습니다.

일반적인 안티 패턴:

- 조직의 관리자 계정에서 워크로드를 실행합니다.
- 프로덕션과 비 프로덕션 워크로드를 동일한 계정에서 실행합니다.

이 모범 사례가 확립되지 않을 경우 노출되는 위험의 수준: 보통

구현 가이드

AWS에서 추가 워크로드를 확장하고 관리하면서, 계정을 사용하여 이러한 워크로드를 분리하고 AWS Organizations를 사용하여 해당 계정을 관리해야 합니다. 이 경우 조직 내 모든 자격 증명에 대한 액세

스를 제한하는 공통 권한 가드레일을 설정하는 것이 좋습니다. 예를 들어, 특정 AWS 리전에 대한 액세스를 제한하거나 중앙 보안팀이 사용하는 IAM 역할과 같은 공통 리소스를 팀에서 삭제하지 못하게 할 수 있습니다.

사용자가 주요 서비스를 비활성화하지 못하도록 방지하는 등 서비스 제어 정책 예시를 구현하는 것부터 시작할 수 있습니다. SCP는 IAM 정책 언어를 사용하여 모든 IAM 보안 주체(사용자 및 역할)가 준수하는 제어를 설정할 수 있습니다. 특정 서비스 작업과 리소스에 대한 액세스를 제한하거나, 조직의 액세스 제어 요구 사항에 부합하도록 특정 조건을 기반으로 액세스를 제한할 수 있습니다. 필요한 경우, 가드레일에 예외를 정의할 수 있습니다. 예를 들어, 주어진 계정에서 특정 관리자 역할을 제외한 모든 IAM 엔터티에 대해 서비스 작업을 제한할 수 있습니다.

관리 계정에서 워크로드를 실행하지 않는 것이 좋습니다. 관리 계정은 멤버 계정에 영향을 미치는 보안 가드레일을 관리 및 배포하는 데 사용해야 합니다. 일부 AWS 서비스는 위임된 관리자 계정의 사용을 지원합니다. 가능한 경우, 이 위임된 계정을 관리 계정 대신 사용해야 합니다. 조직의 관리자 계정에 대한 액세스는 강력하게 제한해야 합니다.

다중 계정 전략을 사용하면 워크로드에 가드레일을 훨씬 더 유연하게 적용할 수 있습니다. AWS Security Reference Architecture는 계정 구조 설계 방법에 대한 권장 가이드를 제공합니다. AWS Control Tower와 같은 AWS 서비스는 조직 전체에 대한 사전 예방 제어와 탐지 제어 모두를 중앙에서 관리할 수 있는 기능을 제공합니다. 조직 내 각 계정 또는 OU에 대한 명확한 목적을 정의하고 해당 목적에 따라 제어를 제한합니다.

리소스

관련 문서:

- AWS Organizations
- 서비스 제어 정책(SCP)
- 다중 계정 환경에서 서비스 제어 정책 최대한 활용
- AWS Security Reference Architecture(AWS SRA)

관련 동영상:

- 서비스 제어 정책을 사용한 예방적 가드레일 적용
- AWS Control Tower를 통해 대규모 거버넌스 구축
- AWS Identity and Access Management 심층 분석

SEC03-BP06 수명 주기에 따라 액세스 관리

액세스 제어를 운영자 및 애플리케이션 수명 주기/중앙 집중식 페더레이션 공급자와 통합합니다. 예를 들어 조직에서 나가거나 역할이 변경될 때 사용자의 액세스 권한을 제거합니다.

워크로드를 관리할 때 별도의 여러 계정을 사용하다 보면, 해당 계정 간에 리소스를 공유해야 하는 경우가 있습니다. 리소스를 공유할 때는 AWS Resource Access Manager(AWS RAM)를 사용하는 것이 좋습니다.. 이 서비스를 사용하면 AWS Organizations 및 조직 단위 내에서 AWS 리소스를 쉽고 안전하게 공유할 수 있습니다. AWS RAM을 사용하면 리소스를 공유하는 조직이나 조직 단위에서의 계정 포함 여부에 따라 공유 리소스에 대한 액세스 권한이 자동으로 부여되거나 취소됩니다. 이렇게 하면 리소스를 원하는 계정끼리만 공유하도록 할 수 있습니다.

이 모범 사례가 수립되지 않을 경우 노출되는 위험의 수준: 낮음

구현 가이드

사용자 액세스 수명 주기 현재 사용자만 액세스할 수 있도록 시설의 신규 참여 사용자, 직무 기능 변경, 퇴거 사용자에 적용할 사용자 액세스 수명 주기 정책을 구현합니다.

리소스

관련 문서:

- ABAC(속성 기반 액세스 제어)
- 최소 권한 부여
- IAM Access Analyzer
- 불필요한 자격 증명 삭제
- 정책 사용

관련 동영상:

- 60분 이내에 IAM 정책 마스터하기
- 업무 분리, 최소 권한, 위임 및 CI/CD

SEC03-BP07 퍼블릭 및 크로스 계정 액세스 분석

퍼블릭 및 크로스 계정 액세스를 강조하는 조사 결과를 지속적으로 모니터링합니다. 이 액세스 권한이 필요한 특정 리소스에 대해서만 퍼블릭 액세스 및 크로스 계정 액세스를 줄입니다.

원하는 결과: AWS 리소스 중 어떤 리소스가 누구와 공유되는지 파악합니다. 공유 리소스를 지속적으로 모니터링하고 감사하여 권한이 부여된 보안 주체와만 공유되는지 확인합니다.

일반적인 안티 패턴:

- 공유 리소스의 인벤토리를 유지하지 않습니다.
- 크로스 계정 또는 리소스에 대한 퍼블릭 액세스를 승인하는 프로세스를 따르지 않습니다.

이 모범 사례를 따르지 않을 경우 노출 위험도: 낮음

구현 가이드

계정이 AWS Organizations에 있는 경우 전체 조직, 특정 조직 단위 또는 개별 계정에 리소스에 대한 액세스 권한을 부여할 수 있습니다. 계정이 조직의 멤버가 아닌 경우 개별 계정과 리소스를 공유할 수 있습니다. 리소스 기반 정책(예: Amazon Simple Storage Service(Amazon S3) 버킷 정책)을 사용하거나 다른 계정의 보안 주체가 사용자 계정의 IAM 역할을 수임하도록 허용하여 직접 크로스 계정 액세스 권한을 부여할 수 있습니다. 리소스 정책을 사용할 때 권한이 부여된 보안 주체에게만 액세스 권한이 부여되는지 확인합니다. 공개적으로 사용 가능해야 하는 모든 리소스를 승인하는 프로세스를 정의합니다.

AWS Identity and Access Management Access Analyzer는 증명 가능한 보안을 사용하여 계정 외부의 리소스에 대한 모든 액세스 경로를 식별합니다. 리소스 정책을 지속적으로 검토하고, 퍼블릭 또는 크로스 계정 액세스의 조사 결과를 보고하여 잠재적으로 광범위한 액세스를 간단하게 분석할 수 있습니다. 모든 계정에 대한 가시성을 확보했는지 확인하려면 AWS Organizations로 IAM Access Analyzer를 구성하는 것이 좋습니다. IAM Access Analyzer를 사용하면 리소스 권한을 배포하기 전에 결과를 미리 볼 수 있습니다. 따라서 정책 변경 사항이 리소스에 대해 의도한 퍼블릭 및 크로스 계정 액세스만 부여하는지 확인할 수 있습니다. 다중 계정 액세스를 위해 설계할 때 신뢰 정책을 사용하여 역할을 수임할 수 있는 경우를 제어할 수 있습니다. 예를 들어 `PrincipalOrgId` 조건 키를 사용하여 AWS Organizations 외부에서 역할을 수임하려는 시도를 거부할 수 있습니다.

AWS Config는 잘못 구성된 리소스를 보고할 수 있으며 AWS Config 정책 확인을 통해 퍼블릭 액세스가 구성된 리소스를 감지할 수 있습니다. AWS Control Tower 및 AWS Security Hub 같은 서비스는 공개적으로 노출된 리소스를 식별하고 개선하기 위해 AWS Organizations 전체에 탐지 제어 및 가드레일 배포를 간소화합니다. 예를 들어 AWS Control Tower에는 Amazon EBS 스냅샷이 AWS 계정에 의해 복원 가능한지 감지할 수 있는 관리형 가드레일이 있습니다.

구현 단계

- AWS Organizations에 대해 AWS Config 활성화 고려: AWS Config를 사용하면 AWS Organizations 내의 여러 계정에서 찾은 조사 결과를 위임된 관리자 계정으로 집계할 수 있습니다. 이는 포괄적인 보기를 제공하고 계정 전체에 AWS Config 규칙를 배포하여 공개적으로 액세스 가능한 리소스를 감지할 수 있습니다.
- AWS Identity and Access Management Access Analyzer 구성: IAM Access Analyzer는 조직 및 계정에서 Amazon S3 버킷 또는 IAM 역할과 같이 외부 엔터티와 공유되는 리소스를 식별하는 데 도움이 됩니다.
- AWS Config에서 자동 개선조치를 사용하여 Amazon S3 버킷의 퍼블릭 액세스 구성 변경에 대응: Amazon S3 버킷에 대한 퍼블릭 액세스 차단 설정을 자동으로 재활성화할 수 있습니다.
- 모니터링 및 알림을 구현하여 Amazon S3 버킷이 공개되었는지 확인: Amazon S3 퍼블릭 액세스 차단이 비활성화된 시점과 Amazon S3 버킷이 공개되었는지 확인하기 위해 모니터링 및 알림을 구현해야 합니다. 또한 AWS Organizations를 사용하는 경우 Amazon S3 퍼블릭 액세스 정책의 변경을 방지하는 서비스 제어 정책을 생성할 수 있습니다. AWS Trusted Advisor는 퍼블릭 액세스 권한이 있는 Amazon S3 버킷을 확인합니다. 모든 사용자에게 업로드 또는 삭제 액세스 권한을 부여하는 버킷 권한은 모든 사용자가 버킷 항목을 추가하거나, 수정하거나, 제거할 수 있도록 허용하여 잠재적 보안 문제가 발생하는 원인이 됩니다. Trusted Advisor 점검 항목은 명시적인 버킷 권한뿐 아니라 버킷 권한을 재정의할 수 있는 관련 버킷 정책도 확인합니다. 또한 AWS Config를 사용하여 퍼블릭 액세스를 위해 Amazon S3 버킷을 모니터링할 수 있습니다. 자세한 내용은 퍼블릭 액세스를 허용하는 Amazon S3 버킷을 모니터링하고 대응하기 위해 AWS Config를 사용하는 방법을 참조하세요. 액세스 권한을 검토하는 동안 Amazon S3 버킷에 포함된 데이터 유형을 고려하는 것이 중요합니다. Amazon Macie은 개인 키 또는 AWS 키와 같은 보안 인증 정보, PII, PHI와 같은 민감한 데이터를 검색하고 보호하는 데 도움이 됩니다.

리소스

관련 문서:

- AWS Identity and Access Management Access Analyzer 사용
- AWS Control Tower 제어 라이브러리
- AWS 기초 보안 모범 사례 표준
- AWS Config 관리형 규칙
- AWS Trusted Advisor 점검 참조
- Amazon EventBridge를 사용하여 AWS Trusted Advisor 점검 결과 모니터링
- 조직의 모든 계정에서 AWS Config 규칙 관리

- AWS Config 및 AWS Organizations

관련 동영상:

- Best Practices for securing your multi-account environment(다중 계정 환경 보안 모범 사례)
- Dive Deep into IAM Access Analyzer(IAM Access Analyzer 심층 분석)

SEC03-BP08 안전하게 조직과 리소스 공유

워크로드 수가 증가함에 따라 해당 워크로드의 리소스에 대한 액세스 권한을 공유하거나 여러 계정에서 리소스를 여러 번 프로비저닝해야 할 수 있습니다. 개발, 테스트 및 프로덕션 환경과 같이 환경을 분류하는 구성이 있을 수 있습니다. 그러나 분리 구성이 있다고 해서 안전하게 공유하는 것이 제한되지는 않습니다. 겹치는 구성 요소를 공유하면 운영 오버헤드를 줄일 수 있고 동일한 리소스를 여러 번 생성하는 동안 누락된 부분을 추측하지 않고도 일관된 경험을 제공할 수 있습니다.

원하는 결과: 안전한 방법을 사용하여 조직 내에서 리소스를 공유하고 데이터 손실 방지 이니셔티브를 지원하여 의도치 않은 액세스를 최소화합니다. 개별 구성 요소를 관리하는 것에 비해 운영 오버헤드를 줄이고 동일한 구성 요소를 수동으로 여러 번 생성할 때 발생하는 오류를 줄이며 워크로드의 확장성을 높입니다. 다중 장애 지점 시나리오에서 해결 시간을 단축할 수 있고 구성 요소가 더 이상 필요하지 않은 시기를 결정할 때 신뢰성을 높일 수 있습니다. 외부에서 공유되는 리소스 분석에 대한 권장 가이드는 SEC03-BP07 퍼블릭 및 크로스 계정 액세스 분석를 참조하세요.

일반적인 안티 패턴:

- 예상치 못한 외부 공유를 지속적으로 모니터링하고 자동으로 알리는 프로세스가 부족합니다.
- 공유해야 할 것과 공유하지 말아야 할 것에 대한 기준이 부족합니다.
- 필요할 때 명시적으로 공유하는 대신 광범위한 공개 정책을 기본으로 설정합니다.
- 필요할 때 겹치는 기본 리소스를 수동으로 생성합니다.

이 모범 사례를 따르지 않을 경우 노출 위험도: 중간

구현 가이드

액세스 제어 및 패턴을 설계하여 공유 리소스의 소비를 신뢰할 수 있는 엔터티로만 안전하게 관리합니다. 공유 리소스를 모니터링하고 공유 리소스 액세스를 지속적으로 검토하고 부적절하거나 예상치 못한 공유에 대한 알림을 받습니다. 퍼블릭 및 크로스 계정 액세스 분석을 검토하면 필요한 리소스에 대

한 외부 액세스 권한을 줄이도록 거버넌스를 설정하고 지속적으로 모니터링하고 자동으로 알리는 프로세스를 설정하는 데 도움이 됩니다.

AWS Organizations 내 크로스 계정 공유는 AWS Security Hub, Amazon GuardDuty 및 AWS Backup과 같은 여러 AWS 서비스에서 지원됩니다. 이러한 서비스를 통해 데이터를 중앙 계정과 공유하거나, 중앙 계정에서 액세스하거나, 중앙 계정에서 리소스 및 데이터를 관리할 수 있습니다. 예를 들어 AWS Security Hub는 조사 결과를 개별 계정에서 모든 조사 결과를 볼 수 있는 중앙 계정으로 전송할 수 있습니다. AWS Backup은 리소스를 백업하고 계정 간에 공유할 수 있습니다. AWS Resource Access Manager(AWS RAM)를 사용하여 VPC 서브넷 및 Transit Gateway 첨부 파일, AWS Network Firewall, Amazon SageMaker 파이프라인과 같은 다른 공통 리소스를 공유할 수 있습니다.

조직 내에서만 리소스를 공유하도록 계정을 제한하려면 서비스 제어 정책(SCP)을 사용하여 외부 보안 주체에 대한 액세스 권한을 방지합니다. 리소스를 공유할 때 자격 증명 기반 제어와 네트워크 제어를 결합하여 조직의 데이터 경계를 생성하여 의도치 않은 액세스로부터 보호합니다. 데이터 경계는 신뢰할 수 있는 자격 증명만 예상 네트워크의 신뢰할 수 있는 리소스에 액세스하고 있는지 확인하는 데 도움이 되는 예방 가드레일입니다. 이러한 제어를 통해 어떤 리소스를 공유할 수 있는지에 대한 적절한 제한을 설정하고, 리소스의 허용되지 않은 공유 또는 노출을 방지할 수 있습니다. 예를 들어 데이터 경계의 일부로 VPC 엔드포인트 정책과 `AWS:PrincipalOrgId` 조건을 사용하여 Amazon S3 버킷에 액세스하는 자격 증명이 조직에 속하는지 확인할 수 있습니다. SCP는 서비스 연결 역할(LSR) 또는 AWS 서비스 보안 주체에 적용되지 않는다는 점에 유의해야 합니다.

Amazon S3를 사용하는 경우 Amazon S3 버킷에 대한 ACL을 비활성화하고 IAM 정책을 사용하여 액세스 제어를 정의합니다. Amazon CloudFront에서 Amazon S3 오리진에 대한 액세스 권한을 제한하려면 오리진 액세스 ID(OAI)에서 AWS Key Management Service를 통한 서버 측 암호화를 비롯한 추가 기능을 지원하는 오리진 액세스 제어(OAC)로 마이그레이션합니다.

경우에 따라 조직 외부에서 리소스 공유를 허용하거나 리소스에 대한 타사 액세스 권한을 부여할 수 있습니다. 외부에서 리소스를 공유하기 위한 권한 관리에 대한 권장 가이드는 권한 관리를 참조하세요.

구현 단계

1. AWS Organizations를 사용합니다.

   AWS Organizations는 여러 AWS 계정를 사용자가 생성하고 중앙에서 관리하는 조직으로 통합할 수 있는 계정 관리 서비스입니다. 계정을 조직 단위(OU)로 그룹화하고 각 OU에 서로 다른 정책을 연결하여 예산, 보안 및 규정 준수 요구 사항을 충족할 수 있습니다. 또한 AWS 인공 지능(AI) 및 기계 학습(ML) 서비스가 데이터를 수집 및 저장하는 방법을 제어하고 Organizations와 통합된 AWS 서비스의 다중 계정 관리를 사용할 수 있습니다.
2. AWS Organizations를 AWS 서비스와 통합합니다.

조직의 멤버 계정에서 사용자를 대신하여 작업을 수행하도록 AWS서비스를 활성화하면 AWS Organizations는 각 멤버 계정에서 해당 서비스에 대한 IAM 서비스 연결 역할을 생성합니다. AWS Management Console, AWS API 또는 AWS CLI를 사용하여 신뢰할 수 있는 액세스를 관리해야 합니다. 신뢰할 수 있는 액세스 활성화에 대한 권장 가이드는 다른 AWS 서비스와 함께 AWS Organizations 사용 및 Organizations와 함께 사용할 수 있는 AWS 서비스를 참조하세요.

3. 데이터 경계를 설정합니다.

   AWS 경계는 일반적으로 AWS Organizations에서 관리하는 조직으로 표시됩니다. 온프레미스 네트워크 및 시스템과 함께 AWS 리소스에 액세스하는 것은 내 AWS의 경계로 간주되는 경우가 많습니다. 경계의 목표는 자격 증명을 신뢰할 수 있고 리소스를 신뢰할 수 있으며 네트워크가 예상되는 경우 액세스가 허용되는지 확인하는 것입니다.

   a. 경계를 정의하고 구현합니다.

      각 권한 부여 조건은 AWS에 경계 구축 백서의 경계 구현에 설명된 단계를 따르세요. 네트워크 계층 보호에 대한 권장 가이드는 네트워크 보호를 참조하세요.

   b. 지속적으로 모니터링하고 알립니다.

      AWS Identity and Access Management Access Analyzer는 조직 및 계정에서 외부 엔터티와 공유되는 리소스를 식별하는 데 도움이 됩니다. IAM Access Analyzer를 AWS Security Hub와 통합하여 IAM Access Analyzer에서 Security Hub로 리소스에 대한 조사 결과를 전송 및 집계하여 환경의 보안 태세를 분석할 수 있습니다. 통합을 활성화하려면 각 계정의 각 리전에서 IAM Access Analyzer 및 Security Hub를 모두 활성화합니다. 또한 AWS Config 규칙를 사용하여 구성을 감사하고 AWS Chatbot을 AWS Security Hub와 함께 사용하여 적절한 당사자에게 알릴 수 있습니다. 그런 다음 AWS Systems Manager 자동화 문서를 사용하여 규정 미준수 리소스를 개선할 수 있습니다.

   c. 외부에서 공유되는 리소스에 대한 지속적인 모니터링 및 알림에 대한 권장 가이드는 퍼블릭 및 크로스 계정 액세스 분석을 참조하세요.

4. AWS 서비스에서 리소스 공유를 사용하고 그에 따라 제한합니다.

   Amazon Machine Image(AMI) 및 AWS Resource Access Manager(AWS RAM)와 같은 많은 AWS 서비스를 통해 다른 계정과 리소스를 공유하거나 다른 계정의 리소스를 대상으로 지정할 수 있습니다. AMI를 공유할 신뢰할 수 있는 계정을 지정하도록 `ModifyImageAttribute` API를 제한합니다. 신뢰할 수 없는 자격 증명의 액세스를 방지하기 위해 AWS RAM을 사용할 때 `ram:RequestedAllowsExternalPrincipals` 조건을 지정하여 조직으로만 공유를 제한합니다. 권장 가이드 및 고려 사항은 리소스 공유 및 외부 대상을 참조하세요.

5. AWS RAM을 사용하여 계정에서 또는 다른 AWS 계정와 안전하게 공유합니다.

AWS RAM은 사용자가 생성한 리소스를 계정의 역할 및 사용자와 다른 AWS 계정와 안전하게 공유하는 데 도움이 됩니다. 다중 계정 환경에서 AWS RAM을 사용하면 리소스를 한 번 생성하여 다른 계정과 공유할 수 있습니다. 이 접근 방식은 크로스 계정 액세스를 사용할 때는 받지 못하는 Amazon CloudWatch 및 AWS CloudTrail과의 통합을 통해 일관성, 가시성 및 감사 가능성을 제공하는 동시에 운영 오버헤드를 줄이는 데 도움이 됩니다.

이전에 리소스 기반 정책을 사용하여 공유한 리소스가 있는 경우 `PromoteResourceShareCreatedFromPolicy API` 또는 이에 상응하는 것을 사용하여 리소스 공유를 전체 AWS RAM 리소스 공유로 승격할 수 있습니다.

경우에 따라 리소스를 공유하기 위해 추가 단계를 수행해야 할 수도 있습니다. 예를 들어 암호화된 스냅샷을 공유하려면 AWS KMS 키를 공유해야 합니다.

리소스

관련 모범 사례:

- SEC03-BP07 퍼블릭 및 크로스 계정 액세스 분석
- SEC03-BP09 안전하게 제3자와 리소스 공유
- SEC05-BP01 네트워크 계층 생성

관련 문서:

- 버킷 소유자가 소유하지 않은 객체에 크로스 계정 권한 부여
- IAM을 통한 신뢰 정책 사용 방법
- AWS에 데이터 경계 구축
- AWS 리소스에 대한 타사 액세스 권한을 부여할 때 외부 자격 증명을 사용하는 방법
- AWS Organizations과 함께 사용할 수 있는 AWS 서비스
- AWS에서 데이터 경계 설정: 신뢰할 수 있는 자격 증명만 회사 데이터에 액세스할 수 있도록 허용

관련 동영상:

- Granular Access with AWS Resource Access Manager(AWS Resource Access Manager를 통한 세분화된 액세스)
- Securing your data perimeter with VPC endpoints(VPC 엔드포인트를 통한 데이터 경계 보호)

- Establishing a data perimeter on AWS(AWS에 데이터 경계 설정)

관련 도구:

- 데이터 경계 정책 예시

SEC03-BP09 안전하게 제3자와 리소스 공유

클라우드 환경의 보안은 조직에 국한되지 않습니다. 조직은 타사를 이용하여 데이터의 일부를 관리할 수 있습니다. 타사 관리형 시스템에 대한 권한 관리는 임시 보안 인증 정보로 최소 권한 원칙을 사용하여 적시 액세스 방식을 따라야 합니다. 타사와 긴밀히 협력하면 영향 범위와 의도치 않은 액세스의 위험을 함께 줄일 수 있습니다.

원하는 결과: 사용자와 연결된 장기 AWS Identity and Access Management(IAM) 보안 인증 정보, IAM 액세스 키 및 보안 암호 키는 보안 인증 정보가 유효하고 활성 상태라면 누구나 사용할 수 있습니다. IAM 역할 및 임시 보안 인증 정보를 사용하면 이러한 민감한 세부 정보의 관리 및 운영 오버헤드를 포함하여 장기 보안 인증 정보를 유지하기 위한 노력을 줄여 전반적인 보안 태세를 개선할 수 있습니다. IAM 신뢰 정책의 외부 자격 증명에 대해 UUID(Universally Unique Identifier)를 사용하고 IAM 역할에 연결된 IAM 정책을 제어하면 타사에 부여된 액세스 권한이 너무 많이 허용되지 않았는지 감사하고 확인할 수 있습니다. 외부에서 공유되는 리소스 분석에 대한 권장 가이드는 SEC03-BP07 퍼블릭 및 크로스 계정 액세스 분석를 참조하세요.

일반적인 안티 패턴:

- 조건 없이 기본 IAM 신뢰 정책을 사용합니다.
- 장기 IAM 보안 인증 정보 및 액세스 키를 사용합니다.
- 외부 ID를 재사용합니다.

이 모범 사례를 따르지 않을 경우 노출 위험도: 중간

구현 가이드

AWS Organizations 외부에서 리소스 공유를 허용하거나 타사에 계정에 대한 액세스 권한을 부여할 수 있습니다. 예를 들어, 타사가 계정 내 리소스에 액세스해야 하는 모니터링 솔루션을 제공할 수 있습니다. 이 경우, 타사에게만 필요한 권한이 포함된 IAM 크로스 계정 역할을 생성합니다. 또한 외부 ID 조건을 사용하여 신뢰 정책을 정의합니다. 외부 ID를 사용하는 경우 사용자 또는 타사는 각 고객, 타사 또는 테넌시용으로 고유한 ID를 생성할 수 있습니다. 고유한 ID는 생성된 후에는 사용자 외에는 누구도

제어할 수 없습니다. 타사는 안전하고 감사 가능하며 재현 가능한 방식으로 외부 ID를 고객과 연결하는 프로세스를 구현해야 합니다.

또한 IAM Roles Anywhere를 사용하여 AWS API를 사용하는 AWS 외부의 애플리케이션에 대한 IAM 역할을 관리할 수 있습니다.

타사가 더 이상 환경에 액세스할 필요가 없으면 역할을 제거합니다. 장기 보안 인증 정보를 타사에 제공하지 않아야 합니다. 공유를 지원하는 다른 AWS 서비스를 계속 인지하고 있어야 합니다. 예를 들어 AWS Well-Architected Tool을 사용하면 다른 AWS 계정와 워크로드를 공유할 수 있고 AWS Resource Access Manager를 사용하면 소유한 AWS 리소스를 다른 계정과 안전하게 공유할 수 있습니다.

구현 단계

1. 크로스 계정 역할을 사용하여 외부 계정에 대한 액세스 권한을 제공합니다.

   크로스 계정 역할은 고객에게 서비스를 제공하기 위해 외부 계정 및 타사에서 저장하는 민감한 정보의 양을 줄입니다. 크로스 계정 역할을 사용하면 AWS Partner 또는 조직의 다른 계정과 같은 타사에 계정의 AWS 리소스에 대한 액세스 권한을 안전하게 부여하는 동시에 해당 액세스를 관리 및 감사하는 기능을 유지할 수 있습니다.

   타사는 하이브리드 인프라에서 서비스를 제공하거나 오프사이트 위치로 데이터를 가져올 수 있습니다. IAM Roles Anywhere는 타사 워크로드가 AWS 워크로드와 안전하게 상호 작용하고 장기 보안 인증 정보의 필요성을 줄이는 데 도움이 됩니다.

   외부 계정 액세스를 제공하기 위해 장기 보안 인증 정보 또는 사용자와 연결된 액세스 키를 사용해서는 안 됩니다. 대신 크로스 계정 역할을 사용하여 크로스 계정 액세스를 제공합니다.
2. 타사의 외부 ID를 사용합니다.

   외부 ID를 사용하면 IAM 신뢰 정책에서 역할을 수임할 수 있는 사람을 지정할 수 있습니다. 신뢰 정책은 역할을 수임하는 사용자가 작동 중인 조건 및 대상을 어설션하도록 요구할 수 있습니다. 또한 계정 소유자가 특정 상황에서만 역할을 수임하도록 허용할 수도 있습니다. 외부 ID의 기본 기능은 혼동된 대리인(confused deputy) 문제를 해결하고 방지하는 것입니다.

   AWS 계정 소유자이고 사용자 외에 다른 AWS 계정에 액세스하는 타사의 역할을 구성했거나 다른 고객을 대신하여 역할을 수임하는 위치에 있는 경우 외부 ID를 사용합니다. 타사 또는 AWS Partner와 협력하여 IAM 신뢰 정책에 포함할 외부 ID 조건을 설정합니다.
3. 범용적으로 고유한 외부 ID를 사용합니다.

   UUID(Universally Unique Identifier)와 같은 외부 ID에 대해 임의의 고유한 값을 생성하는 프로세스를 구현합니다. 고객 A가 중복된 외부 ID와 함께 고객 B의 역할 ARN을 사용하여 고객 B의 데이터

를 볼 수 있으므로, 서로 다른 고객 간에 외부 ID를 재사용하는 타사의 경우 혼동된 대리인(confused deputy) 문제가 해결되지 않습니다. 타사가 서로 다른 AWS 계정로 여러 고객을 지원하는 다중 테넌트 환경에서 타사는 각 AWS 계정에 대한 외부 ID로 서로 다른 고유한 ID를 사용해야 합니다. 타사는 중복된 외부 ID를 감지하고 각 고객을 해당 외부 ID에 안전하게 매핑하는 업무를 담당합니다. 타사는 외부 ID를 지정할 때만 역할을 수임할 수 있는지 확인하기 위해 테스트해야 합니다. 타사는 외부 ID가 필요하기 전에는 고객 역할 ARN 및 외부 ID를 저장하지 않아야 합니다.

외부 ID는 보안 암호로 취급되지는 않지만 전화번호, 이름, 계정 ID와 같이 쉽게 추측할 수 있는 값이 아니어야 합니다. 설정을 가장할 목적으로 외부 ID를 변경할 수 없도록 외부 ID를 읽기 전용 필드로 만듭니다.

사용자 또는 타사가 외부 ID를 생성할 수 있습니다. ID 생성 담당자를 결정하는 프로세스를 정의합니다. 외부 ID를 생성하는 엔터티에 관계없이 타사는 고객 간에 고유성과 형식을 일관되게 적용합니다.

4. 고객이 제공한 장기 보안 인증 정보를 지원 중단합니다.

   장기 보안 인증 정보 사용을 중단하고 크로스 계정 역할 또는 IAM Roles Anywhere를 사용합니다. 장기 보안 인증 정보를 사용해야 하는 경우 역할 기반 액세스로의 마이그레이션 계획을 수립합니다. 키 관리에 대한 자세한 내용은 자격 증명 관리를 참조하세요. 또한 AWS 계정 팀 및 타사와 협력하여 위험 완화 런북을 수립합니다. 보안 인시던트의 잠재적 영향에 대한 대응 및 완화에 대한 권장 가이드는 인시던트 대응을 참조하세요.

5. 설정에 권장 가이드가 있거나 자동화되어 있는지 확인합니다.

   계정의 크로스 계정 액세스를 위해 생성된 정책은 최소 권한 원칙을 따라야 합니다. 타사는 역할 정책 문서를 제공하거나 AWS CloudFormation 템플릿 또는 이에 상응하는 템플릿을 사용하는 자동화된 설정 메커니즘을 제공해야 합니다. 이는 수동 정책 생성과 관련된 오류가 발생할 가능성을 줄이고 감사 가능한 트레일을 제공합니다. AWS CloudFormation 템플릿을 사용하여 크로스 계정 역할을 생성하는 방법에 대한 자세한 내용은 크로스 계정 역할을 참조하세요.

   타사는 자동화되고 감사 가능한 설정 메커니즘을 제공해야 합니다. 그러나 필요한 액세스를 설명하는 역할 정책 문서를 사용하여 역할 설정을 자동화해야 합니다. AWS CloudFormation 템플릿 또는 이에 상응하는 템플릿을 사용하여 감사 관행의 일환으로 드리프트 감지를 사용하여 변경 사항을 모니터링해야 합니다.

6. 변경 사항을 설명합니다.

   계정 구조, 타사에 대한 요구 사항 또는 제공되는 서비스 오퍼링이 변경될 수 있습니다. 변경 사항과 장애를 예상하고 적절한 인력, 프로세스 및 기술을 사용하여 그에 따라 계획을 수립해야 합니다. 사용자가 제공하는 액세스 수준을 정기적으로 감사하고 감지 방법을 구현하여 예상치 못한 변경 사항

을 알립니다. 외부 ID의 역할 및 데이터 스토어의 사용을 모니터링하고 감사합니다. 예상치 못한 변경 사항 또는 액세스 패턴의 결과로 일시적으로 또는 영구적으로 타사 액세스를 취소할 준비가 되어 있어야 합니다. 또한 수행하는 데 걸리는 시간, 관련된 담당자, 비용, 다른 리소스에 미치는 영향을 포함하여 취소 작업에 미치는 영향을 측정합니다.

감지 방법에 대한 권장 가이드는 감지 모범 사례를 참조하세요.

리소스

관련 모범 사례:

- SEC02-BP02 임시 보안 인증 정보 사용
- SEC03-BP05 조직에 대한 권한 가드레일 정의
- SEC03-BP06 수명 주기에 따라 액세스 관리
- SEC03-BP07 퍼블릭 및 크로스 계정 액세스 분석
- SEC04 감지

관련 문서:

- 버킷 소유자가 소유하지 않은 객체에 크로스 계정 권한 부여
- IAM 역할을 통한 신뢰 정책 사용 방법
- IAM 역할을 사용하여 AWS 계정 간에 액세스 위임
- IAM을 사용하여 다른 AWS 계정의 리소스에 액세스하려면 어떻게 해야 하나요?
- IAM의 보안 모범 사례
- 크로스 계정 정책 평가 로직
- AWS 리소스에 액세스 권한을 타사에 부여할 때 외부 ID를 사용하는 방법
- 사용자 지정 리소스를 사용하여 외부 계정에서 생성된 AWS CloudFormation 리소스에서 정보 수집
- 다른 사용자가 소유한 AWS 계정에 액세스하기 위한 외부 ID를 안전하게 사용
- IAM Roles Anywhere를 사용하여 IAM 외부의 워크로드로 IAM 역할 확장

관련 동영상:

- How do I allow users or roles in a separate AWS 계정 access to my AWS 계정?(내 AWS 계정에 대한 별도의 AWS 계정 액세스에서 사용자 또는 역할을 허용하려면 어떻게 해야 하나요?)

- AWS re:Invent 2018: Become an IAM Policy Master in 60 Minutes or Less(AWS re:Invent 2018: 60분 이내에 IAM 정책 마스터하기)
- AWS Knowledge Center Live: IAM Best Practices and Design Decisions(AWS 지식 센터 라이브: IAM 모범 사례 및 설계 결정)

관련 예시:

- Well-Architected Lab - Lambda cross account IAM role assumption (Level 300)(Lambda 크로스 계정 IAM 역할 수임(레벨 300))
- Configure cross-account access to Amazon DynamoDB(Amazon DynamoDB에 대한 크로스 계정 액세스 구성)
- AWS STS Network Query Tool(AWS STS 네트워크 쿼리 도구)

탐지

두 부분으로 구성된 탐지: 예기치 않거나 원치 않는 구성 변경 탐지 및 예기치 않은 동작 탐지. 전자는 애플리케이션 전송 수명 주기의 여러 지점에서 발생합니다. 코드형 인프라(예: CloudFormation 템플릿)를 사용하면 CI/CD 파이프라인에 검사를 구현하거나 또는 소스 제어를 구현하여 워크로드가 배포되기 전에 원치 않는 구성을 검사할 수 있습니다. 그런 다음 비프로덕션 및 프로덕션 환경에 워크로드를 배포할 때 기본 AWS 도구, 오픈 소스 도구 또는 AWS 파트너 도구를 사용하여 구성을 검사할 수 있습니다. 이러한 검사는 보안 원칙 또는 모범 사례를 준수하지 않는 구성을 찾거나 구성 테스트와 배포 사이에 있었던 변경 사항을 찾기 위해 수행될 수 있습니다. 실행 중인 애플리케이션에 대해서는 알려진 배포 또는 자동화된 크기 조정 이벤트 외에 구성이 원치 않는 방식으로 변경되었는지 확인할 수 있습니다.

두 번째 탐지인 예기치 않은 동작의 경우 도구를 사용하거나 특정 API 호출 유형이 증가하면 알림을 받을 수 있습니다. Amazon GuardDuty를 사용하면 예기치 않았으며 무단의 가능성이 있는 악성 활동이 AWS 계정에서 발생하면 알림을 받을 수 있습니다. 또한 워크로드에서 사용될 것으로 예상되지 않는 API 호출 변경과 보안 태세를 변경하는 API 호출을 명시적으로 모니터링해야 합니다.

탐지를 통해 잠재적인 보안 구성 오류, 위협 또는 예기치 않은 동작을 식별할 수 있습니다. 이것은 보안 수명 주기의 핵심 부분으로서 품질 프로세스, 법률 또는 규정 준수 의무, 위협 식별 및 대응 과정을 지원하는 데 사용됩니다. 탐지 메커니즘에는 여러 가지 유형이 있습니다. 예를 들어 워크로드 로그를 분석하여 사용 중인 익스플로잇을 알아낼 수 있습니다. 워크로드와 관련된 탐지 메커니즘을 정기적으로 검토하여 사내외 정책과 요구 사항에 부합하는지 확인해야 합니다. 자동 알림은 팀이나 도구가 조사에 착수할 수 있도록 정의된 조건을 기반으로 설정해야 합니다. 이러한 메커니즘은 조직 내에서 변칙적 활동 범위를 식별하고 파악하는 데 도움이 되는 중요한 대응 요소입니다.

AWS에는 탐지 메커니즘을 다룰 때 사용할 수 있는 방식이 아주 많습니다. 다음 섹션에서는 아래와 같은 방식을 사용하는 방법을 설명합니다.

모범 사례

- SEC04-BP01 서비스 및 애플리케이션 로깅 구성
- SEC04-BP02 로그, 결과 및 지표를 중앙에서 분석
- SEC04-BP03 이벤트 대응 자동화
- SEC04-BP04 조치 가능한 보안 이벤트 구현

SEC04-BP01 서비스 및 애플리케이션 로깅 구성

서비스 및 애플리케이션의 보안 이벤트 로그를 유지합니다. 이는 감사, 조사 및 운영 사용 사례에 대한 보안의 기본 원칙이며 거버넌스, 위험 및 규정 준수(GRC) 표준, 정책 및 절차를 기반으로 하는 공통 보안 요구 사항입니다.

원하는 결과: 조직은 AWS 서비스 및 애플리케이션에서 보안 인시던트 대응과 같은 내부 프로세스 또는 의무를 이행해야 할 때 적시에 안정적이고 일관되게 보안 이벤트 로그를 검색할 수 있어야 합니다. 더 나은 운영 결과를 위해 로그를 중앙 집중화하는 것이 좋습니다.

일반적인 안티 패턴:

- 로그는 영구적으로 저장되거나 너무 빨리 삭제됩니다.
- 누구나 로그에 액세스할 수 있습니다.
- 로그 거버넌스 및 사용을 위해 수동 프로세스에 전적으로 의존합니다.
- 필요한 경우를 대비하여 모든 유형의 로그를 저장합니다.
- 필요한 경우에만 로그 무결성을 확인합니다.

이 모범 사례 확립의 이점: 보안 인시던트에 대한 근본 원인 분석(RCA) 메커니즘과 거버넌스, 위험 및 규정 준수 의무에 대한 증거 소스를 구현합니다.

이 모범 사례를 따르지 않을 경우 노출 위험도: 높음

구현 가이드

요구 사항에 따른 보안 조사 또는 기타 사용 사례 중에 관련 로그를 검토하여 인시던트의 전체 범위와 타임라인을 기록하고 이해할 수 있어야 합니다. 관심 있는 특정 작업이 발생했음을 나타내는 알림 생성에도 로그가 필요합니다. 쿼리 및 검색 메커니즘과 알림을 선택, 활성화, 저장 및 설정하는 것이 중요합니다.

구현 단계

- 로그 소스를 선택하고 활성화합니다. 보안 조사에 앞서 관련 로그를 캡처하여 AWS 계정의 활동을 소급하여 재구성해야 합니다. 워크로드와 관련된 로그 소스를 선택하고 활성화합니다.

  로그 소스 선택 기준은 비즈니스에 필요한 사용 사례를 기반으로 해야 합니다. AWS CloudTrail 또는 AWS Organizations 트레일을 사용하여 각 AWS 계정에 대한 트레일을 설정하고 이에 대한 Amazon S3 버킷을 구성합니다.

AWS CloudTrail은 AWS 서비스 활동을 캡처하는 AWS 계정에 대해 수행된 API 호출을 추적하는 로깅 서비스입니다. AWS Management Console, AWS CLI 또는 AWS SDK를 사용하여 CloudTrail 이벤트 기록을 통해 검색할 수 있도록 기본적으로 관리 이벤트의 90일 보존으로 활성화됩니다. 데이터 이벤트를 더 오래 보존하고 가시성을 확보하려면 CloudTrail 트레일을 생성하고 이를 Amazon S3 버킷과 연결하고 선택적으로 Amazon CloudWatch 로그 그룹과 연결합니다. 또는 최대 7년 동안 CloudTrail 로그를 유지하고 SQL 기반 쿼리 기능을 제공하는 CloudTrail Lake를 생성할 수 있습니다.

AWS는 VPC를 사용하는 고객이 각각 VPC 흐름 로그 및 Amazon Route 53 확인자 쿼리 로그를 사용하여 네트워크 트래픽 및 DNS 로그를 활성화하고 Amazon S3 버킷 또는 CloudWatch 로그 그룹으로 스트리밍할 것을 권장합니다. VPC, 서브넷 또는 네트워크 인터페이스에 대한 VPC 흐름 로그를 생성할 수 있습니다. VPC 흐름 로그의 경우 흐름 로그를 사용하는 방법과 위치를 선택하여 비용을 절감할 수 있습니다.

AWS CloudTrail 로그, VPC 흐름 로그 및 Route 53 확인자 쿼리 로그는 AWS에서 보안 조사를 지원하는 기본 로깅 소스입니다. 또한 Amazon Security Lake를 사용하여 쿼리할 준비가 된 Apache Parquet 형식 및 OCSF(Open Cybersecurity Schema Framework)로 이 로그 데이터를 수집, 정규화 및 저장할 수 있습니다. Security Lake는 다른 AWS 로그와 타사 소스의 로그도 지원합니다.

AWS 서비스는 Elastic Load Balancing 로그, AWS WAF 로그, AWS Config 레코더 로그, Amazon GuardDuty 조사 결과, Amazon Elastic Kubernetes Service(Amazon EKS) 감사 로그, Amazon EC2 인스턴스 운영 체제 및 애플리케이션 로그와 같은 기본 로그 소스에서 캡처하지 않은 로그를 생성할 수 있습니다. 로깅 및 모니터링 옵션의 전체 목록은 AWS 보안 인시던트 대응 가이드의 부록 A: 클라우드 기능 정의 - 로깅 및 이벤트를 참조하세요.

- 각 AWS 서비스 및 애플리케이션에 대한 로깅 기능 연구: 각 AWS 서비스 및 애플리케이션은 각각 고유한 보존 및 수명 주기 기능이 있는 로그 스토리지 옵션을 제공합니다. 가장 일반적인 두 가지 로그 스토리지 서비스는 Amazon Simple Storage Service(Amazon S3) 및 Amazon CloudWatch입니다. 보존 기간이 긴 경우 비용 효율성과 유연한 수명 주기 기능을 위해 Amazon S3를 사용하는 것이 좋습니다. 기본 로깅 옵션이 Amazon CloudWatch 로그인 경우 액세스 빈도가 낮은 로그를 Amazon S3에 아카이브하는 것이 좋습니다.
- 로그 스토리지 선택: 로그 스토리지 선택은 일반적으로 사용하는 쿼리 도구, 보존 기능, 친숙도, 비용과 관련이 있습니다. 로그 스토리지의 기본 옵션은 Amazon S3 버킷 또는 CloudWatch 로그 그룹입니다.

  Amazon S3 버킷은 선택적 수명 주기 정책을 통해 비용 효율적이고 내구성이 뛰어난 스토리지를 제공합니다. Amazon S3 버킷에 저장된 로그는 Amazon Athena와 같은 서비스를 사용하여 쿼리할 수 있습니다.

CloudWatch 로그 그룹은 CloudWatch Logs Insights를 통해 내구성이 뛰어난 스토리지와 기본 제공 쿼리 기능을 제공합니다.

- 적절한 로그 보존 파악: Amazon S3 버킷 또는 CloudWatch 로그 그룹을 사용하여 로그를 저장하는 경우 각 로그 소스에 적절한 수명 주기를 설정하여 저장 및 검색 비용을 최적화해야 합니다. 고객은 일반적으로 3개월에서 1년 사이의 로그를 쉽게 쿼리할 수 있으며 최대 7년 동안 보존할 수 있습니다. 가용성 및 보존에 대한 선택은 보안 요구 사항과 법적, 규제 및 비즈니스 의무의 조합과 일치해야 합니다.
- 적절한 보존 및 수명 주기 정책으로 각 AWS 서비스 및 애플리케이션에 대한 로깅 활성화: 조직의 각 AWS 서비스 또는 애플리케이션에 대해 특정 로깅 구성 지침을 찾습니다.
  - AWS CloudTrail 트레일 구성
  - VPC Flow Logs 구성
  - Amazon GuardDuty 조사 결과 내보내기 구성
  - AWS Config 기록 구성
  - AWS WAF 웹 ACL 트래픽 구성
  - AWS Network Firewall 네트워크 트래픽 로그 구성
  - Elastic Load Balancing 액세스 로그 구성
  - Amazon Route 53 확인자 쿼리 로그 구성
  - Amazon RDS 로그 구성
  - Amazon EKS 컨트롤 플레인 로그 구성
  - Amazon EC2 인스턴스 및 온프레미스 서버에 대한 Amazon CloudWatch 에이전트 구성
- 로그에 대한 쿼리 메커니즘 선택 및 구현: 로그 쿼리의 경우 CloudWatch 로그 그룹에 저장된 데이터에는 CloudWatch Logs Insight를 사용할 수 있고 Amazon S3에 저장된 데이터에는 Amazon Athena 및 Amazon OpenSearch Service를 사용할 수 있습니다. 보안 정보 및 이벤트 관리(SIEM) 서비스와 같은 타사 쿼리 도구를 사용할 수도 있습니다.

  로그 쿼리 도구를 선택하는 프로세스는 보안 작업의 인력, 프로세스 및 기술 측면을 고려해야 합니다. 운영, 비즈니스 및 보안 요구 사항을 충족하고 장기적으로 액세스 및 유지 관리 가능한 도구를 선택합니다. 로그 쿼리 도구는 스캔할 로그 수가 도구의 한도 내에서 유지될 때 최적으로 작동합니다. 비용이나 기술적 제약으로 인해 여러 쿼리 도구를 사용하는 것이 일반적입니다.

  예를 들어 타사 보안 정보 및 이벤트 관리(SIEM) 도구를 사용하여 지난 90일 데이터에 대한 쿼리를 수행할 수 있지만, SIEM의 로그 수집 비용으로 인해 90일 이후 데이터에 대한 쿼리를 수행할 때는 Athena를 사용합니다. 구현에 관계없이, 특히 보안 이벤트 조사 중에 운영 효율성을 극대화하는 데 필요한 도구의 수를 최소화하는 접근 방식인지 확인합니다.

- 알림에 로그 사용: AWS는 여러 보안 서비스를 통해 알림을 제공합니다.
  - AWS Config는 AWS 리소스 구성을 모니터링 및 기록하며, 원하는 구성을 기준으로 평가 및 개선 조치를 자동화할 수 있습니다.
  - Amazon GuardDuty는 AWS 계정 및 워크로드를 보호하기 위해 악의적인 활동 및 무단 동작을 지속적으로 모니터링하는 위협 탐지 서비스입니다. GuardDuty는 AWS CloudTrail 관리 및 데이터 이벤트, DNS 로그, VPC 흐름 로그 및 Amazon EKS 감사 로그와 같은 소스에서 정보를 수집, 집계 및 분석합니다. GuardDuty는 CloudTrail, VPC 흐름 로그, DNS 쿼리 로그 및 Amazon EKS에서 직접 독립적인 데이터 스트림을 가져옵니다. Amazon S3 버킷 정책을 관리하거나 로그를 수집하고 저장하는 방식을 수정할 필요가 없습니다. 자체 조사 및 규정 준수 목적으로 이러한 로그를 보관하는 것이 좋습니다.
  - AWS Security Hub는 여러 AWS 서비스 및 타사 제품(선택 사항)의 보안 알림 또는 조사 결과를 집계하고 정리하고 우선순위를 지정함으로써 보안 알림 및 규정 준수 상태를 종합적으로 파악할 수 있는 단일 장소를 제공합니다.

  이러한 서비스에서 다루지 않는 보안 알림 또는 환경과 관련된 특정 알림에 대해 사용자 지정 알림 생성 엔진을 사용할 수도 있습니다. 이러한 알림 및 감지 구축에 대한 자세한 내용은 AWS 보안 인시던트 대응 탐지 가이드를 참조하세요.

리소스

관련 모범 사례:

- SEC04-BP02 로그, 결과 및 지표를 중앙에서 분석
- SEC07-BP04 데이터 수명 주기 관리 정의
- SEC10-BP06 도구 사전 배포

관련 문서:

- AWS 보안 인시던트 대응 가이드
- Amazon Security Lake 시작하기
- 시작하기: Amazon CloudWatch Logs
- 보안 파트너 솔루션: 로깅 및 모니터링

관련 동영상:

- AWS re:Invent 2022 - Introducing Amazon Security Lake(AWS re:Invent 2022 - Amazon Security Lake 소개)

관련 예시:

- Assisted Log Enabler for AWS(AWS의 지원 로그 인에이블러)
- AWS Security Hub Findings Historical Export(AWS Security Hub 조사 결과 기록 내보내기)

관련 도구:

- Snowflake 사이버 보안

SEC04-BP02 로그, 결과 및 지표를 중앙에서 분석

보안 운영팀은 로그를 수집하고 검색 도구를 사용하여 발생 가능한 관심 이벤트(무단 활동 또는 의도하지 않은 변경을 나타낼 수 있음)를 검색할 수 있습니다. 하지만 수집된 데이터를 분석하고 정보를 수동으로 처리하는 것만으로는 복잡한 아키텍처에서 유입되는 대량의 정보를 파악하기가 어렵습니다. 분석 및 보고만 수행하면 이벤트를 처리하는 데 적합한 리소스를 제때 원활하게 할당할 수 없습니다.

완성된 보안 운영팀을 구축하기 위한 모범 사례는 보안 이벤트 흐름 및 이벤트에서 확인된 정보를 알림 및 워크플로 시스템(예: 티켓팅 시스템, 버그 또는 문제 시스템 또는 기타 보안 정보 및 이벤트 관리(SIEM) 시스템)에 심층적으로 통합하는 것입니다. 이렇게 하면 이메일 및 정적 보고서가 아닌 효율적 방식으로 워크플로를 파악할 수 있으며 이벤트나 이벤트를 통해 확인된 정보를 라우팅, 에스컬레이션 및 관리할 수 있습니다. 대부분의 조직은 보안 알림도 채팅 또는 협업 및 개발자 생산성 플랫폼에 통합하고 있습니다. 자동화에 착수하는 조직의 경우, API 기반의 지연 시간이 짧은 티켓팅 시스템은 먼저 자동화할 대상을 계획할 때 상당한 유연성을 제공합니다.

이러한 모범 사례는 사용자 활동이나 네트워크 이벤트를 보여 주는 로그 메시지에서 생성된 보안 이벤트뿐 아니라 인프라 자체에서 감지된 변경 사항에도 적용됩니다. 어느 정도의 변화를 받아들일 것인지에 대한 기준이 명확하지 않기 때문에 AWS Identity and Access Management(IAM) 및 AWS Organizations 구성의 조합을 통해 변경 사항이 실행되는 것을 방지할 수 없는 경우에는 변경을 감지하고 변경 사항이 적절한지 판단한 다음 해당 정보를 올바른 수정 워크플로로 라우팅하는 능력이 보안 아키텍처를 유지 관리하고 검증하는 데 필수적입니다.

Amazon GuardDuty 및 AWS Security Hub는 다른 AWS 서비스를 통해서도 사용할 수 있는 로그 레코드에 대한 집계, 중복 제거, 분석 메커니즘을 제공합니다. GuardDuty는 AWS CloudTrail 관리 및 데이터 이벤트, VPC DNS 로그, VPC 흐름 로그와 같은 소스에서의 정보를 수집, 집계, 분석합니다.

Security Hub는 GuardDuty, AWS Config, Amazon Inspector, Amazon Macie, AWS Firewall Manager 및 AWS Marketplace에서 사용할 수 있는 수많은 서드 파티 보안 제품에서의 출력은 물론 적절히 구축하는 경우 사용자 자체 코드에서의 출력을 수집, 집계, 분석할 수 있습니다. GuardDuty 및 Security Hub 모두 여러 계정의 결과와 분석 정보를 집계할 수 있는 관리자-멤버 모델을 보유하고 있으며, Security Hub는 온프레미스 SIEM을 AWS 측 로그 및 알림 프리프로세서와 어그리게이터로 사용하는 고객들이 주로 사용합니다. 이러한 고객들은 AWS Lambda 기반 프로세서와 전달자를 통해 Amazon EventBridge에 수집할 수 있습니다.

이 모범 사례가 수립되지 않을 경우 노출되는 위험의 수준: 높음

구현 가이드

- 로그 처리 기능 평가: 로그 처리에 사용할 수 있는 옵션을 평가합니다.
  - Amazon OpenSearch Service를 사용하여 거의 모든 것을 로깅 및 모니터링하기
  - 로깅 및 모니터링 솔루션 전문 파트너 찾기
- CloudTrail 로그 분석을 시작할 때 Amazon Athena를 테스트합니다.
  - Athena를 구성하여 CloudTrail 로그 분석
- AWS에서 중앙 집중식 로깅 구현: 다음 AWS 예시 솔루션을 통해 여러 소스에서 로깅을 중앙 집중화하는 방법을 알아보십시오.
  - 로깅 솔루션 중앙 집중화
- 파트너와 중앙 집중식 로깅 구현: APN 파트너는 중앙에서 로그를 분석하는 데 도움이 되는 솔루션을 보유하고 있습니다.
  - 로깅 및 모니터링

리소스

관련 문서:

- AWS Answers: 중앙 집중식 로깅
- AWS Security Hub
- Amazon CloudWatch
- Amazon EventBridge
- 시작하기: Amazon CloudWatch Logs
- 보안 파트너 솔루션: 로깅 및 모니터링

관련 동영상:

- 리소스 구성 및 규정 준수를 중앙에서 모니터링
- Amazon GuardDuty 및 AWS Security Hub 결과 수정
- 클라우드에서 위협 관리: Amazon GuardDuty 및 AWS Security Hub

SEC04-BP03 이벤트 대응 자동화

자동화 기능을 사용하여 이벤트를 조사하고 해결하면 수작업 부담과 인적 오류가 줄어들고 조사 역량을 확대할 수 있습니다. 정기적인 검토는 자동화 도구를 튜닝하고 지속적으로 반복하는 데 도움이 됩니다.

AWS에서는 Amazon EventBridge를 사용하여 관심 있는 이벤트와 자동화된 워크플로에 대해 예기치 않은 잠재적 변경 사항에 대한 정보를 조사할 수 있습니다. 이 서비스는 AWS CloudTrail 이벤트 등의 기본 AWS 이벤트 형식과 애플리케이션에서 생성 가능한 사용자 지정 이벤트를 중개하도록 설계된 확장 가능 규칙 엔진을 제공합니다. 또한 Amazon GuardDuty를 사용하면 인시던트 대응 시스템(AWS Step Functions)을 구축하는 워크플로 시스템 또는 중앙 보안 계정에 이벤트를 라우팅하거나, 추가 분석을 위해 버킷에 이벤트를 라우팅할 수 있습니다.

AWS Config 규칙 및 규정 준수 팩을사용하여 변경 사항을 감지하고 이 정보를 올바른 워크플로로 라우팅할 수도 있습니다. AWS Config는 범위 내 서비스의 변경 사항을 감지(EventBridge보다 지연 시간이 길)한 다음 롤백/규정 준수 정책 적용/변경 관리 플랫폼 및 운영 티켓팅 시스템 등으로 정보 전달을 위해, AWS Config 규칙 규칙을 사용하여 구문 분석할 수 있는 이벤트를 생성합니다. 자체 Lambda 함수를 작성하여 AWS Config 이벤트에 응답하는 것은 물론 AWS Config 규칙 개발 키트 및 오픈 소스 AWS Config 규칙도 활용할 수 있습니다. 규정 준수 팩은 YAML 템플릿으로 작성된 단일 엔터티로 배포하는 AWS Config 규칙 및 해결 조치의 모음입니다. 샘플 규정 준수 팩 템플릿은 AWS Well-Architected 보안 원칙에서 사용 가능합니다.

이 모범 사례가 수립되지 않을 경우 노출되는 위험의 수준: 보통

구현 가이드

- GuardDuty로 자동화된 알림 구현: GuardDuty는 악성 활동 및 무단 행위를 지속적으로 모니터링하여 AWS 계정 및 워크로드를 보호하는 위협 탐지 서비스입니다. GuardDuty를 활성화하고 자동 알림을 구성합니다.
- 조사 프로세스 자동화: 이벤트를 조사하여 관리자가 시간을 절약할 수 있도록 정보를 보고하는 자동화된 프로세스를 개발합니다.
  - 실습: Amazon GuardDuty 체험

리소스

관련 문서:

- AWS Answers: 중앙 집중식 로깅
- AWS Security Hub
- Amazon CloudWatch
- Amazon EventBridge
- 시작하기: Amazon CloudWatch Logs
- 보안 파트너 솔루션: 로깅 및 모니터링
- Amazon GuardDuty 설정

관련 동영상:

- 리소스 구성 및 규정 준수를 중앙에서 모니터링
- Amazon GuardDuty 및 AWS Security Hub 결과 수정
- 클라우드에서 위협 관리: Amazon GuardDuty 및 AWS Security Hub

관련 예시:

- 실습: 탐지 제어 자동 배포

SEC04-BP04 조치 가능한 보안 이벤트 구현

팀에게 전송되고 팀에서 조치를 취할 수 있는 알림을 생성합니다. 팀에서 조치를 취하는 데 필요한 관련 정보가 알림에 포함되도록 합니다. 보유한 각 탐지 메커니즘에 대해 런북 또는 플레이북형태의 조사 프로세스도 있어야 합니다. 예를 들어 Amazon GuardDuty를 활성화하면 서로 다른 결과가 생성됩니다.. 각 결과 유형에 대한 런북 항목이 있어야 합니다. 예를 들어 트로이 목마 를 발견한 경우에는 누군가에게 조사 및 수정을 지시하는 간단한 지침이 런북 안에 있습니다.

이 모범 사례를 정립하지 않을 경우 노출되는 위험의 수준: 낮음

구현 가이드

- AWS 서비스에서 사용할 수 있는 지표 파악: 사용 중인 서비스에 대해 Amazon CloudWatch를 통해 사용할 수 있는 지표를 검색합니다.

  - AWS 서비스 설명서
  - Amazon CloudWatch 지표 사용
- Amazon CloudWatch 경보를 구성합니다.
  - Amazon CloudWatch 경보 사용

리소스

관련 문서:

- Amazon CloudWatch
- Amazon EventBridge
- 보안 파트너 솔루션: 로깅 및 모니터링

관련 동영상:

- 리소스 구성 및 규정 준수를 중앙에서 모니터링
- Amazon GuardDuty 및 AWS Security Hub 결과 수정
- 클라우드에서 위협 관리: Amazon GuardDuty 및 AWS Security Hub

인프라 보호

모범 사례와 업계 규정 또는 규제 의무를 준수하기 위해서는 인프라 보호가 필요하며, 여기에는 심층 방어 등의 제어 방법이 포함됩니다. 클라우드에서 작업을 계속 성공적으로 수행하려면 이러한 방법을 사용해야 합니다.

인프라 보호는 정보 보안 프로그램의 핵심 요소입니다. 인프라 보호 기능을 사용하면 의도하지 않은 무단 침입 및 잠재적 취약성으로부터 워크로드 내의 시스템과 서비스를 보호할 수 있습니다. 예를 들어 신뢰 경계(예: 네트워크 및 계정 경계), 시스템 보안 구성 및 유지 관리(예: 강화, 최소화 및 패치 적용), 운영 체제 인증 및 권한 부여(예: 사용자, 키 및 액세스 수준) 및 기타 적절한 정책 적용 지점(예: 웹 애플리케이션 방화벽 및/또는 API 게이트웨이)을 정의할 수 있습니다.

리전, 가용 영역, AWS 로컬 영역, AWS Outposts

리전, 가용 영역, AWS 로컬 영역및 AWS Outposts등 AWS 보안 글로벌 인프라의 구성 요소에 친숙해지시기 바랍니다.

AWS는 데이터 센터를 클러스터링하는 세계 곳곳의 물리적 로케이션을 의미하는 리전이라는 개념을 사용합니다. 각각의 논리적 데이터 센터 그룹을 가용 영역(AZ)이라고 합니다. 각 AWS 리전은 지리적 영역 내에서 물리적으로 분리된 여러 개의 격리된 AZ로 구성됩니다. 데이터 상주 요구 사항이 있는 경우 원하는 위치와 가까운 AWS 리전을 선택하면 됩니다. 데이터가 물리적으로 위치한 리전에 대한 제어와 소유권은 완전히 사용자에게 있으므로 지역의 규정을 준수하고 데이터 상주 요구 사항을 충족하는 데 유용합니다. 각 AZ에는 독립적인 전원과 냉각 시스템, 물리적 보안이 구축되어 있습니다. 하나의 애플리케이션이 여러 AZ에 파티셔닝되어 있는 경우 정전, 번개, 토네이도, 지진 등의 문제로부터 애플리케이션을 더 효과적으로 격리하고 보호할 수 있습니다. AZ는 다른 AZ로부터 물리적으로 수 킬로미터에 달하는 상당한 거리로 분리되어 있으나 서로 100km(60마일) 반경 내에 있습니다. 한 AWS 리전 내의 모든 AZ는 완전히 중복된 전용 메트로 섬유를 사용하여 고대역폭과 짧은 지연 시간의 네트워킹으로 상호 연결되어 있기 때문에 AZ 간에 높은 처리량과 짧은 지연 시간을 제공합니다. AZ 간의 모든 트래픽은 암호화됩니다. 고가용성을 중시하는 AWS 고객은 더 높은 내결함성을 달성하기 위해 애플리케이션이 여러 개의 AZ에서 실행되도록 설계할 수 있습니다. AWS 리전은 가장 높은 수준의 보안, 규정 준수, 데이터 보호 요건을 충족합니다.

AWS 로컬 영역은 컴퓨팅, 스토리지, 데이터베이스 및 선택한 기타 AWS 서비스가 최종 사용자에게 더 가까이에서 실행되도록 합니다. AWS 로컬 영역을 사용하면 미디어 및 엔터테인먼트 콘텐츠 제작, 실시간 게이밍, 저수지 시뮬레이션, 전자 설계 자동화, 기계 학습 등 최종 사용자에게 10밀리 초 미만의 지연 시간을 요구하는 매우 까다로운 애플리케이션을 쉽게 실행할 수 있습니다. 각 AWS 로컬 영역 위치는 AWS 리전이 확장된 것으로, Amazon EC2, Amazon VPC, Amazon EBS, Amazon File Storage,

Elastic Load Balancing 등의 AWS 서비스를 사용하여 최종 사용자에게 지리적으로 가까운 곳에서 지연 시간에 민감한 애플리케이션을 실행할 수 있습니다. AWS 로컬 영역은 로컬 워크로드와 AWS 리전에서 실행되는 워크로드 간에 고대역폭의 보안 연결을 제공하므로 동일한 API와 도구 집합을 사용하여 리전 내의 모든 서비스에 원활하게 연결할 수 있습니다.

AWS Outposts는 기본 AWS 서비스, 인프라 및 운영 모델을 거의 모든 데이터 센터, 코로케이션 공간 또는 온프레미스 설비에 제공합니다. 온프레미스 시설과 AWS 클라우드 전체에서 동일한 AWS API, 도구, 인프라를 사용하여 진정으로 일관성 있는 하이브리드 경험을 제공할 수 있습니다. AWS Outposts는 연결된 환경을 위해 설계되었으며 짧은 지연 시간으로 인해 또는 로컬 데이터 처리 요구 사항에 따라 온프레미스에 남아 있어야 하는 워크로드를 지원하는 데 사용 가능합니다.

AWS에서는 다양한 방식으로 인프라를 보호할 수 있습니다. 다음 섹션에서는 아래와 같은 방식을 사용하는 방법을 설명합니다.

주제

- 네트워크 보호
- 컴퓨팅 보호

네트워크 보호

기업 내의 사용자와 고객 측 사용자는 어디에나 있을 수 있습니다. 네트워크에 액세스하는 모든 사람과 모든 주체를 신뢰하는 전통적인 모델로부터 전환해야 합니다. 모든 계층에 보안을 적용하는 원칙을 따르면 제로 트러스트 접근 방식을 사용하게 됩니다. 제로 트러스트 보안은 애플리케이션 구성 요소 또는 마이크로서비스를 서로 별개의 것으로 간주하고 어떤 구성 요소나 마이크로서비스도 서로를 신뢰하지 않는 모델입니다.

워크로드 내의 리소스에 격리와 경계를 적용하려면 기본적으로 네트워크 설계를 철저하게 계획하고 관리해야 합니다. 워크로드의 많은 리소스가 VPC에서 작동하고 보안 속성을 상속하기 때문에 자동화된 검사 및 보호 메커니즘을 기반으로 설계하는 것이 중요합니다. 마찬가지로, 순전히 엣지 서비스 및/또는 서버리스를 사용하여 VPC 외부에서 작동하는 워크로드의 경우에는 모범 사례가 좀 더 간단하게 적용됩니다. 애플리케이션이 이 시나리오에서 다루지 않는 검색 기능을 사용하는 경우 AWS Well-Architected 서버리스 애플리케이션 렌즈 를 참조하십시오.

모범 사례

- SEC05-BP01 네트워크 계층 생성
- SEC05-BP02 모든 계층에서 트래픽 제어

- SEC05-BP03 네트워크 보호 자동화
- SEC05-BP04 검사 및 보호 구현

SEC05-BP01 네트워크 계층 생성

민감도 요구 사항을 공유하는 구성 요소를 계층으로 그룹화하여 무단 액세스의 잠재적 영향 범위를 최소화합니다. 예를 들어 인터넷에 액세스할 필요가 없는 Virtual Private Cloud(VPC)의 데이터베이스 클러스터는 인터넷에 연결되는 경로가 없는 서브넷에 배치해야 합니다. 트래픽은 인접해 있는 다음으로 덜 민감한 리소스에서만 전달되어야 합니다. 로드 밸런서 뒤에 있는 웹 애플리케이션을 고려합니다. 데이터베이스는 로드 밸런서에서 직접 액세스할 수 없어야 합니다. 비즈니스 로직 또는 웹 서버만 데이터베이스에 직접 액세스할 수 있어야 합니다.

원하는 결과: 계층화된 네트워크를 생성합니다. 계층화된 네트워크는 유사한 네트워킹 구성 요소를 논리적으로 그룹화하는 데 도움이 됩니다. 또한 무단 네트워크 액세스의 잠재적 영향 범위를 축소합니다. 적절하게 계층화된 네트워크는 권한이 없는 사용자가 AWS 환경 내에서 추가 리소스로 전환하는 것을 더 어렵게 만듭니다. 내부 네트워크 경로를 보호하는 것 외에도 웹 애플리케이션 및 API 엔드포인트와 같은 네트워크 엣지도 보호해야 합니다.

일반적인 안티 패턴:

- 단일 VPC 또는 서브넷에서 모든 리소스를 생성합니다.
- 지나치게 관대한 보안 그룹을 사용합니다.
- 서브넷을 사용하지 못합니다.
- 데이터베이스와 같은 데이터 스토어에 직접 액세스할 수 있습니다.

이 모범 사례를 따르지 않을 경우 노출 위험도: 높음

구현 가이드

연결성 요구 사항을 공유하는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스, Amazon Relational Database Service(Amazon RDS) 데이터베이스 클러스터, AWS Lambda 함수와 같은 구성 요소를 서브넷으로 구성된 계층으로 분할할 수 있습니다. VPC 내 또는 Amazon API Gateway 뒤에 Lambda 함수와 같은 서버리스 워크로드를 배포하는 것이 좋습니다. 인터넷에 액세스할 필요가 없는 AWS Fargate (Fargate) 작업은 인터넷에 연결되는 경로가 없는 서브넷에 배치해야 합니다. 이러한 계층화된 접근 방식은 의도치 않은 액세스를 허용할 수 있는 단일 계층 구성 오류로 인한 영향을 완화합니다. AWS Lambda의 경우, VPC에서 함수를 실행하여 VPC 기반 제어를 활용할 수 있습니다.

수천 개의 VPC, AWS 계정 및 온프레미스 네트워크를 포함할 수 있는 네트워크 연결의 경우 AWS Transit Gateway를 사용해야 합니다. Transit Gateway는 스포크처럼 작동하는 연결된 모든 네트워크 간에 트래픽이 라우팅되는 방식을 제어하는 허브 역할을 합니다. Amazon Virtual Private Cloud(Amazon VPC)와 Transit Gateway 간의 트래픽은 AWS 프라이빗 네트워크에 남아 있어 권한이 없는 사용자 및 잠재적인 보안 문제에 대한 외부 노출을 줄입니다. 또한 Transit Gateway 리전 간 피어링은 단일 장애 지점이나 대역폭 병목 없이 리전 간 트래픽을 암호화합니다.

구현 단계

- Reachability Analyzer를 사용하여 구성에 따라 소스와 대상 사이의 경로를 분석: Reachability Analyzer를 사용하면 VPC 연결 리소스와의 연결 확인을 자동화할 수 있습니다. 이 분석은 구성을 검토하여 수행됩니다(분석을 수행할 때 네트워크 패킷이 전송되지 않음).
- Amazon VPC Network Access Analyzer를 사용하여 리소스에 대한 의도치 않은 네트워크 액세스 식별: Amazon VPC Network Access Analyzer를 사용하면 네트워크 액세스 요구 사항을 지정하고 잠재적인 네트워크 경로를 식별할 수 있습니다.
- 리소스가 퍼블릭 서브넷에 있어야 하는지 고려: 리소스가 퍼블릭 소스에서 인바운드 네트워크 트래픽을 반드시 수신해야 하는 경우가 아니라면 VPC의 퍼블릭 서브넷에 리소스를 배치하지 않아야 합니다.
- VPC에 서브넷 생성: 각 네트워크 계층(여러 가용 영역을 포함하는 그룹)에 대한 서브넷을 생성하여 마이크로 세분화를 개선합니다. 또한 라우팅 및 인터넷 연결을 제어하기 위해 올바른 라우팅 테이블을 서브넷과 연결했는지 확인합니다.
- AWS Firewall Manager를 사용하여 VPC 보안 그룹 관리: AWS Firewall Manager는 여러 보안 그룹을 사용하는 관리 부담을 줄이는 데 도움이 됩니다.
- AWS WAF를 사용하여 일반적인 웹 취약성으로부터 보호: AWS WAF는 트래픽에서 SQL 명령어 삽입과 같은 일반적인 웹 취약성을 검사하여 엣지 보안을 강화하는 데 도움이 됩니다. 또한 특정 국가 또는 지리적 위치에서 발생하는 IP 주소의 트래픽을 제한할 수 있습니다.
- Amazon CloudFront를 콘텐츠 배포 네트워크(CDN)로 사용: Amazon CloudFront는 데이터를 사용자에게 더 가깝게 저장하여 웹 애플리케이션 속도를 높이는 데 도움이 됩니다. 또한 HTTPS를 시행하고, 지리적 영역에 대한 액세스 권한을 제한하고, 네트워크 트래픽이 CloudFront를 통해 라우팅될 때만 리소스에 액세스할 수 있도록 하여 엣지 보안을 개선할 수 있습니다.
- 애플리케이션 프로그래밍 인터페이스(API) 생성 시 Amazon API Gateway 사용: Amazon API Gateway는 REST, HTTPS 및 WebSocket API를 게시, 모니터링 및 보호하는 데 도움이 됩니다.

리소스

관련 문서:

- AWS Firewall Manager
- Amazon Inspector
- Amazon VPC 보안
- Reachability Analyzer
- Amazon VPC Network Access Analyzer

관련 동영상:

- AWS Transit Gateway reference architectures for many VPCs(여러 VPC를 위한 AWS Tranit Gateway 참조 아키텍처)
- Application Acceleration and Protection with Amazon CloudFront, AWS WAF, and AWS Shield(Amazon CloudFront, AWS WAF 및 AWS Shield를 사용한 애플리케이션 가속화 및 보호)
- AWS re:Inforce 2022 - Validate effective network access controls on AWS(AWS re:Inforce 2022 - AWS에서 효과적인 네트워크 액세스 검증)
- AWS re:Inforce 2022 - Advanced protections against bots using AWS WAF(AWS re:Inforce 2022 - AWS WAF를 사용하여 봇에 대한 고급 보호)

관련 예시:

- Well-Architected Lab - Automated Deployment of VPC(VPC 자동 배포)
- 워크숍: Amazon VPC Network Access Analyzer

SEC05-BP02 모든 계층에서 트래픽 제어

네트워크 토폴로지를 설계할 때 각 구성 요소의 연결 요구 사항을 조사해야 합니다. 예를 들어 구성 요소에 인터넷 액세스(인바운드 및 아웃바운드), VPC 연결, 엣지 서비스, 외부 데이터 센터가 필요한지 조사해야 합니다.

VPC를 사용하면 설정한 프라이빗 IPv4 주소 범위 또는 AWS에서 선택한 IPv6 주소 범위를 사용하여 AWS 리전 전반의 네트워크 토폴로지를 정의할 수 있습니다. 보안 그룹(상태 저장 검사 방화벽), 네트워크 ACL, 서브넷, 라우팅 테이블을 사용하는 등 인바운드 및 아웃바운드 트래픽 모두에 대해 심층적인 방어 접근 방식을 갖춘 여러 제어를 적용해야 합니다. VPC 내의 가용 영역에서 서브넷을 생성할 수 있습니다. 각 서브넷에는 서브넷 내의 트래픽이 전송되는 경로 관리를 위한 라우팅 규칙을 정의하는 연결된 경로 테이블이 있을 수 있습니다. VPC에 연결된 인터넷 또는 NAT 게이트웨이로 이동하거나 다른 VPC를 통해 이동하는 경로를 설정하면 인터넷 라우팅 가능한 서브넷을 정의할 수 있습니다.

VPC 내에서 시작되는 인스턴스, Amazon Relational Database Service(Amazon RDS) 데이터베이스 또는 기타 서비스에는 네트워크 인터페이스별로 자체 보안 그룹이 있습니다. 이 방화벽은 운영 체제 계층 외부에 있으며, 허용되는 인바운드 및 아웃바운드 트래픽용 규칙을 정의하는 데 사용할 수 있습니다. 보안 그룹 간의 관계를 정의할 수도 있습니다. 예를 들어 데이터베이스 계층 보안 그룹 내의 인스턴스는 관련 인스턴스에 적용된 보안 그룹을 참조하여 애플리케이션 계층 내의 인스턴스에서 전송하는 트래픽만 수락합니다. 비TCP 프로토콜을 사용하지 않는 한, 로드 밸런서 또는 다음 서비스 없이 인터넷에서 직접 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에 액세스할 필요가 없습니다(보안 그룹에 의해 제한된 포트를 사용하는 경우에도). CloudFront. 이것은 운영 체제 또는 애플리케이션 문제를 통해 이루어지는 무단 침입으로부터 보호하는 데 도움이 됩니다. 서브넷은 상태 비저장 방화벽 역할을 하는 네트워크 ACL을 연결할 수도 있습니다. 계층 간에 허용되는 트래픽 범위를 좁히도록 네트워크 ACL을 구성해야 합니다. 이때 인바운드 규칙과 아웃바운드 규칙을 모두 정의해야 합니다.

일부 AWS 서비스에서는 API 호출을 위해 AWS API 엔드포인트가 위치한 인터넷에 구성 요소가 액세스해야 합니다. 다른 AWS 서비스에서는 VPC 엔드포인트 를 Amazon VPC 내에서 사용합니다. Amazon S3 및 Amazon DynamoDB를 비롯한 여러 AWS 서비스가 VPC 엔드포인트를 지원하며, 이 기술은 다음에서 일반화되었습니다. AWS PrivateLink. 이 접근법을 사용하여 AWS 서비스, 서드 파티 서비스, 다른 VPC에 호스팅되는 사용자의 자체 서비스에 안전하게 액세스하는 것을 권장합니다. AWS PrivateLink의 모든 트래픽은 글로벌 AWS 백본에 유지되며 인터넷을 통해 이동하지 않습니다. 연결은 서비스의 제공업체가 아닌 서비스의 소비자만 시작할 수 있습니다. 외부 서비스 액세스에 AWS PrivateLink를 사용하면 인터넷 액세스 없이 에어 갭 VPC를 생성할 수 있으며 VPC를 외부 위협 벡터로부터 보호하는 데 도움이 됩니다. 서드 파티 서비스는 AWS PrivateLink를 사용하여 고객이 VPC에서 프라이빗 IP 주소를 통해 서비스에 연결하도록 할 수 있습니다. 인터넷에 아웃바운드 연결해야 하는 VPC 자산의 경우 AWS 관리형 NAT 게이트웨이, 아웃바운드 전용 인터넷 게이트웨이 또는 사용자가 생성하고 관리하는 웹 프록시를 통해 아웃바운드 전용(단방향)으로 연결할 수 있습니다.

이 모범 사례를 정립하지 않을 경우 노출되는 위험의 수준: 높음

구현 가이드

- VPC에서 네트워크 트래픽 제어: VPC 모범 사례를 구현하여 트래픽을 제어합니다.
  - Amazon VPC 보안
  - VPC 엔드포인트
  - Amazon VPC 보안 그룹
  - 네트워크 ACL
- 엣지에서 트래픽 제어: Amazon CloudFront와 같은 엣지 서비스를 구현하여 추가 보호 계층과 기타 기능을 제공합니다.
  - Amazon CloudFront 사용 사례

  - AWS Global Accelerator
  - AWS Web Application Firewall(AWS WAF)
  - Amazon Route 53
  - Amazon VPC 인그레스 라우팅
- 프라이빗 네트워크 트래픽 제어: 워크로드에 대한 프라이빗 트래픽을 보호하는 서비스를 구현합니다.
  - Amazon VPC 피어링
  - Amazon VPC 엔드포인트 서비스(AWS PrivateLink)
  - Amazon VPC Transit Gateway
  - AWS Direct Connect
  - AWS Site-to-Site VPN
  - AWS 클라이언트 VPN
  - Amazon S3 액세스 포인트

리소스

관련 문서:

- AWS Firewall Manager
- Amazon Inspector
- AWS WAF 시작하기

관련 동영상:

- AWS Transit Gateway reference architectures for many VPCs(여러 VPC를 위한 AWS Transit Gateway 참조 아키텍처)
- Amazon CloudFront, AWS WAF, AWS Shield를 사용한 애플리케이션 가속화 및 보호

관련 예시:

- 실습: VPC 자동 배포

SEC05-BP03 네트워크 보호 자동화

위협 정보 및 이상 상태 감지 결과에 따라 자체 방어 네트워크를 제공하는 보호 메커니즘을 자동화합니다. 예를 들어 최신 위협에 적응하고 위협의 영향을 줄일 수 있는 침입 탐지 및 방지 도구가 있습니다. 웹 애플리케이션 방화벽은 네트워크 보호를 자동화할 수 있는 곳의 일례입니다. 예를 들어 AWS WAF Security Automations 솔루션(https://github.com/awslabs/aws-waf-security-automations)을 사용하여 알려진 위협 요소와 연결된 IP 주소에서 시작되는 요청을 자동으로 차단할 수 있습니다.

이 모범 사례를 정립하지 않을 경우 노출되는 위험의 수준: 보통

구현 가이드

- 웹 기반 트래픽에 대한 보호 자동화: AWS는 일반적인 웹 기반 공격을 필터링하도록 설계된 AWS WAF 규칙 세트를 AWS CloudFormation을 사용하여 자동으로 배포하는 솔루션을 제공합니다. 사용자는 AWS WAF 웹 액세스 제어 목록(웹 ACL)에 포함된 규칙을 정의하도록 사전 구성된 다양한 보호 기능 중에서 선택할 수 있습니다.
  - AWS WAF 보안 자동화
- AWS Partner 솔루션 고려: AWS 파트너는 고객 온프레미스 환경의 기존 제어 솔루션과 동등한 수준이거나, 동일하거나 통합된 업계 최고 수준의 수백 가지 제품을 제공합니다. 이러한 제품은 기존 AWS 서비스를 보완하여 클라우드 및 온프레미스 환경에 포괄적인 보안 아키텍처와 보다 원활한 환경을 배포할 수 있도록 해줍니다.
  - 인프라 보안

리소스

관련 문서:

- AWS Firewall Manager
- Amazon Inspector
- Amazon VPC 보안
- AWS WAF 시작하기

관련 동영상:

- AWS Transit Gateway reference architectures for many VPCs(여러 VPC를 위한 AWS Transit Gateway 참조 아키텍처)

- Amazon CloudFront, AWS WAF, AWS Shield를 사용한 애플리케이션 가속화 및 보호

관련 예시:

- 실습: VPC 자동 배포

SEC05-BP04 검사 및 보호 구현

각 계층에서 트래픽을 검사하고 필터링합니다. 사용자는 VPC Network Access Analyzer를 사용하여 잠재적인 의도치 않은 액세스에 대해 VPC 구성을 검사할 수 있습니다. 네트워크 액세스 요구 사항을 지정하고 이 요구 사항을 충족하지 않는 잠재적인 네트워크 경로를 찾을 수 있습니다. HTTP 기반 프로토콜을 통해 트랜잭션되는 구성 요소의 경우, 웹 애플리케이션 방화벽이 일반 공격으로부터 보호하는 데 도움을 줄 수 있습니다. AWS WAF 는 Amazon API Gateway API, Amazon CloudFront 또는 Application Load Balancer로 전달되는 구성 가능한 규칙과 일치하는 HTTP(s) 요청을 모니터링하고 차단할 수 있는 웹 애플리케이션 방화벽입니다. AWS WAF를 시작하려면 AWS Managed Rules 를 자체 규칙과 함께 사용하거나 기존 파트너 통합을 사용할 수 있습니다..

AWS Organizations 전반에서 AWS WAF, AWS Shield Advanced 보호, Amazon VPC 보안 그룹을 관리하기 위해 AWS Firewall Manager를 사용할 수 있습니다. 그러면 여러 계정과 애플리케이션의 방화벽 규칙을 중앙에서 구성하고 관리할 수 있으므로 일반 규칙 적용을 좀 더 쉽게 확장할 수 있습니다. 공격에 신속하게 대응하기 위해 AWS Shield Advanced나 웹 애플리케이션에 대한 원치 않는 요청을 자동으로 차단할 수 있는 솔루션 을 사용할 수도 있습니다. Firewall Manager는 AWS Network Firewall과도 연동됩니다. AWS Network Firewall은 규칙 엔진을 사용하여 스테이트풀 및 스테이트리스 네트워크를 모두 세세하게 제어할 수 있는 관리형 서비스입니다. 이 서비스는 워크로드 보호에 도움이 되도록 규칙에 Suricata 호환 오픈 소스 침입 예방 시스템(IPS) 사양을 지원합니다.

이 모범 사례가 수립되지 않을 경우 노출되는 위험의 수준: 낮음

구현 가이드

- Amazon GuardDuty 구성: GuardDuty는 악성 활동 및 무단 행위를 지속적으로 모니터링하여 AWS 계정 및 워크로드를 보호하는 위협 탐지 서비스입니다. GuardDuty를 활성화하고 자동 알림을 구성합니다.
  - Amazon GuardDuty
  - 실습: 탐지 제어 자동 배포
- Virtual Private Cloud(VPC) 흐름 로그 구성: VPC 흐름 로그는 VPC의 네트워크 인터페이스에서 전송되고 수신되는 IP 트래픽에 대한 정보를 캡처하는 데 사용할 수 있는 기능입니다. 흐름 로그 데이터

를 Amazon CloudWatch Logs 및 Amazon Simple Storage Service(Amazon S3)에 게시할 수 있습니다. 흐름 로그를 생성한 후에는 선택한 대상에서 해당 데이터를 검색하여 확인할 수 있습니다.

- VPC 트래픽 미러링 고려: 트래픽 미러링은 콘텐츠 검사, 위협 모니터링 및 문제 해결을 위해 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스의 탄력적 네트워크 인터페이스에서 네트워크 트래픽을 복사한 다음 대역 외 보안 및 모니터링 어플라이언스로 전송하는 데 사용할 수 있는 Amazon VPC 기능입니다.
  - VPC 트래픽 미러링

리소스

관련 문서:

- AWS Firewall Manager
- Amazon Inspector
- Amazon VPC 보안
- AWS WAF 시작하기

관련 동영상:

- 여러 VPC를 위한 AWS Transit Gateway 참조 아키텍처
- Amazon CloudFront, AWS WAF, AWS Shield를 사용한 애플리케이션 가속화 및 보호

관련 예시:

- 실습: VPC 자동 배포

컴퓨팅 보호

컴퓨팅 리소스에는 EC2 인스턴스, 컨테이너, AWS Lambda 함수, 데이터베이스 서비스, IoT 디바이스 등이 포함됩니다. 이 컴퓨팅 리소스 유형은 보안을 위해 저마다 다른 접근 방식을 요구합니다. 그러나 사용자가 고려해야 하는 공통된 전략이 있습니다. 바로 종심 방어, 취약점 관리, 공격 표면 축소, 구성 및 운영 자동화, 원격 작업 수행입니다. 이 섹션에서는 주요 서비스의 컴퓨팅 리소스를 보호하는 일반적인 지침을 배웁니다. 사용하는 AWS 서비스마다 서비스 설명서에서 구체적인 보안 권장 사항을 확인하는 것이 중요합니다.

모범 사례

SEC06-BP01 취약성 관리 수행

코드, 종속성 및 인프라에 취약성이 있는지 자주 스캔하고 패치를 적용하여 새로운 위협으로부터 보호합니다.

원하는 결과: 취약성 관리 프로그램을 생성하고 유지합니다. Amazon EC2 인스턴스, Amazon Elastic Container Service(Amazon ECS) 컨테이너 및 Amazon Elastic Kubernetes Service(Amazon EKS) 워크로드와 같은 리소스를 정기적으로 스캔하고 패치를 적용합니다. Amazon Relational Database Service(Amazon RDS) 데이터베이스와 같은 AWS 관리형 리소스에 대한 유지 관리 기간을 구성합니다. 정적 코드 스캔을 사용하여 일반적인 문제에 대한 애플리케이션 소스 코드를 검사합니다. 조직에 필요한 기술이 있거나 외부 지원을 고용할 수 있는 경우 웹 애플리케이션 침투 테스트를 고려합니다.

일반적인 안티 패턴:

- 취약성 관리 프로그램이 없습니다.
- 심각도나 위험 회피를 고려하지 않고 시스템 패치 적용을 수행합니다.
- 공급업체에서 제공한 수명 종료(EOL) 날짜가 지난 소프트웨어를 사용합니다.
- 보안 문제를 분석하기 전에 코드를 프로덕션 환경에 배포합니다.

이 모범 사례 확립의 이점:

이 모범 사례를 따르지 않을 경우 노출 위험도: 높음

구현 가이드

취약성 관리 프로그램에는 문제 해결의 일환으로 보안 평가, 문제 식별, 우선순위 지정 및 패치 작업 수행이 포함됩니다. 자동화는 워크로드를 지속적으로 스캔하여 문제 및 의도치 않은 네트워크 노출이 있는지 확인하고 문제 해결 수행이 핵심입니다. 리소스 생성 및 업데이트를 자동화하면 시간이 절약되고 추가 문제를 야기하는 구성 오류의 위험이 줄어듭니다. 잘 설계된 취약성 관리 프로그램은 소프트웨어

수명 주기의 개발 및 배포 단계에서 취약성 테스트도 고려해야 합니다. 개발 및 배포 중에 취약성 관리를 구현하면 취약성이 프로덕션 환경에 침투할 가능성을 줄이는 데 도움이 됩니다.

취약성 관리 프로그램을 구현하려면 AWS 공동 책임 모델과 이 모델이 특정 워크로드와 어떻게 관련되어 있는지 파악하고 있어야 합니다. 공동 책임 모델에 따라 AWS는 AWS 클라우드의 인프라를 보호하는 업무를 담당합니다. 이 인프라는 AWS 클라우드 서비스를 실행하는 하드웨어, 소프트웨어, 네트워킹 및 시설로 구성됩니다. 사용자는 Amazon EC2 인스턴스의 실제 데이터, 보안 구성, 관리 작업과 같은 클라우드의 보안을 담당하고 Amazon S3 객체가 적절하게 분류 및 구성되었는지 확인해야 합니다. 취약성 관리에 대한 접근 방식은 사용하는 서비스에 따라 달라질 수 있습니다. 예를 들어 AWS은 관리되는 관계형 데이터베이스 서비스인 Amazon RDS에 대한 패치 적용을 관리하지만 셀프 호스팅 데이터베이스에 패치 적용은 사용자가 담당합니다.

AWS에는 취약성 관리 프로그램을 지원하는 다양한 서비스가 있습니다. Amazon Inspector는 AWS 워크로드를 지속적으로 스캔하여 소프트웨어 문제 및 의도치 않은 네트워크 액세스가 있는지 확인합니다. AWS Systems Manager Patch Manager는 Amazon EC2 인스턴스 전반에 대한 패치 적용을 관리하는 데 도움이 됩니다. Amazon Inspector 및 Systems Manager는 AWS 보안 점검을 자동화하고 보안 알림을 중앙 집중화하는 데 도움이 되는 클라우드 보안 상태 관리 서비스인 AWS Security Hub에서 볼 수 있습니다.

Amazon CodeGuru는 정적 코드 분석을 사용하여 Java 및 Python 애플리케이션에서 잠재적인 문제를 식별하는 데 도움이 될 수 있습니다.

구현 단계

- Amazon Inspector 구성: Amazon Inspector는 새로 실행된 Amazon EC2 인스턴스, Lambda 함수 및 Amazon ECR에 푸시된 적격한 컨테이너 이미지를 자동으로 감지하고 즉시 스캔하여 소프트웨어 문제, 잠재적 결함 및 의도치 않은 네트워크 노출이 있는지 확인합니다.
- 소스 코드 스캔: 라이브러리 및 종속성을 스캔하여 문제 및 결함이 있는지 확인합니다. Amazon CodeGuru는 Java 및 Python 애플리케이션을 모두 스캔하여 일반적인 보안 문제가 있는지 확인하고 이를 해결하기 위한 권장 사항을 제공할 수 있습니다. OWASP Foundation은 소스 코드 분석 도구(SAST 도구라고도 함) 목록을 게시합니다.
- CI/CD 파이프라인 빌드 프로세스의 일환으로 스캔할 뿐만 아니라 기존 환경을 스캔하고 패치를 적용하는 메커니즘 구현: 보호를 위해 종속성 및 운영 체제의 문제를 스캔하고 패치를 적용하는 메커니즘을 구현하여 새로운 위협으로부터 보호합니다. 해당 메커니즘을 정기적으로 실행합니다. 소프트웨어 취약성 관리는 패치를 적용하거나 소프트웨어 문제를 해결해야 하는 위치를 파악하는 데 필수적입니다. 취약성 평가를 지속적 통합/지속적 전달(CI/CD) 파이프라인에 조기에 포함하여 잠재적 보안 문제의 해결 우선순위를 지정합니다. 사용 중인 AWS 서비스에 따라 접근 방식이 달라질 수 있습니다. Amazon EC2 인스턴스에서 실행 중인 소프트웨어의 잠재적 문제를 확인하려면

Amazon Inspector를 파이프라인에 추가하여 문제 또는 잠재적 결함이 감지되는 경우 이를 알리고 빌드 프로세스를 중지합니다. Amazon Inspector는 지속적으로 리소스를 모니터링합니다. OWASP Dependency-Check, Snyk, OpenVAS와 같은 오픈 소스 제품, 패키지 관리자, 취약성 관리를 위한 AWS Partner 도구를 사용할 수도 있습니다.

- AWS Systems Manager 사용: AWS(Amazon Elastic Compute Cloud) 인스턴스, Amazon Machine Image(AMI), 기타 컴퓨팅 리소스를 비롯한 Amazon EC2 리소스에 대한 패치 관리에 대한 책임은 사용자에게 있습니다. AWS Systems Manager Patch Manager는 보안 관련 업데이트와 기타 유형의 업데이트를 모두 사용하여 관리형 인스턴스에 패치를 적용하는 프로세스를 자동화합니다. Patch Manager는 Microsoft 애플리케이션, Windows 서비스 팩, Linux 기반 인스턴스용 마이너 버전 업그레이드를 포함하여 운영 체제 및 애플리케이션 모두에 대해 Amazon EC2 인스턴스에 패치를 적용하는 데 사용할 수 있습니다. Amazon EC2 외에도 Patch Manager를 사용하면 온프레미스 서버에 패치를 적용할 수도 있습니다.

  지원되는 운영 체제 목록은 Systems Manager 사용자 가이드에서 지원되는 운영 체제를 참조하세요. 인스턴스를 스캔하여 누락된 패치 보고서만 확인하거나, 스캔하고 누락된 모든 패치를 자동으로 설치할 수 있습니다.
- AWS Security Hub 사용: Security Hub는 AWS의 보안 상태에 대한 포괄적인 보기를 제공합니다. 여러 AWS 서비스에서 보안 데이터를 수집하고 이러한 조사 결과를 표준화된 형식으로 제공하므로 AWS 서비스에서 보안 조사 결과의 우선순위를 지정할 수 있습니다.
- AWS CloudFormation 사용: AWS CloudFormation은 여러 계정 및 환경에서 리소스 배포를 자동화하고 리소스 아키텍처를 표준화하여 취약성 관리에 도움을 줄 수 있는 코드형 인프라(IaC) 서비스입니다.

리소스

관련 문서:

- AWS Systems Manager
- AWS Lambda 보안 개요
- Amazon CodeGuru
- 새로운 Amazon Inspector를 사용하여 클라우드 워크로드를 위한 개선되고 자동화된 취약성 관리
- Amazon Inspector 및 AWS Systems Manager를 사용하여 AWS에서 취약성 관리 및 개선조치 자동화 - 1부

관련 동영상:

- Securing Serverless and Container Services(서버리스 및 컨테이너 서비스 보호)
- Security best practices for the Amazon EC2 instance metadata service(Amazon EC2 인스턴스 메타데이터 서비스에 대한 보안 모범 사례)

SEC06-BP02 공격 표면 축소

운영 체제를 강화하고 사용 중인 구성 요소, 라이브러리 및 외부 사용 서비스를 최소화하여 의도치 않은 액세스에 대한 노출을 줄입니다. 운영 체제 패키지 또는 애플리케이션이나(Amazon Elastic Compute Cloud(Amazon EC2) 기반 워크로드의 경우) 코드의 외부 소프트웨어 모듈(모든 워크로드의 경우)에서 사용하지 않는 구성 요소를 줄이는 것으로 시작합니다. 일반적인 운영 체제 및 서버 소프트웨어에 대한 여러 가지 강화 및 보안 구성 가이드를 찾아볼 수 있습니다. 예를 들면 Center for Internet Security 로 시작하고 반복하면 됩니다.

Amazon EC2에서는 자체적으로 패치하고 강화한 Amazon Machine Image(AMI)를 생성하여 조직의 특정 보안 요구 사항을 충족할 수 있습니다. AMI에 적용하는 패치 및 기타 보안 제어 조치는 생성 시점에 발효되며 시작한 후 AWS Systems Manager 등을 사용하여 수정하지 않는 이상 동적으로 변경되지 않습니다.

EC2 Image Builder를 사용하여 보안 AMI 구축 프로세스를 간소화할 수 있습니다. EC2 Image Builder는 자동화를 작성하고 유지 관리할 필요 없이 골든 이미지를 생성하여 유지 관리하는 데 필요한 노력을 크게 절감해 줍니다. 소프트웨어 업데이트가 가능하면 사용자가 이미지 빌드를 시작할 필요 없이 Image Builder가 자동으로 새로운 이미지를 생성합니다. EC2 Image Builder를 사용하면 이미지를 프로덕션에 사용하기 전에 AWS에서 제공하는 테스트와 사용자의 자체 테스트를 사용하여 이미지의 기능과 보안을 쉽게 검증할 수 있습니다. 또한 AWS에서 제공하는 보안 설정을 적용하여 이미지 보안을 강화함으로써 내부 보안 기준을 충족할 수 있습니다. 예를 들면 AWS에서 제공하는 템플릿을 사용하여 Security Technical Implementation Guide(STIG) 표준을 준수하는 이미지를 생성할 수 있습니다.

서드 파티 정적 코드 분석 도구를 사용하여 확인되지 않은 함수 입력 범위와 해당하는 일반 취약성 및 노출(CVE)와 같은 일반적인 보안 문제를 식별합니다. 전용 인프라에서 Amazon CodeGuru 를 지원되는 언어에 대해 사용할 수 있습니다. 또한 종속성 확인 도구를 사용하여 코드가 링크된 라이브러리가 최신 버전인지, 해당 라이브러리에 CVE가 없는지, 소프트웨어 정책 요구 사항에 부합하는 라이선스 조건이 있는지를 확인할 수 있습니다.

Amazon Inspector를 사용하면 인스턴스에 대해 알려진 CVE를 확인하는 구성 평가를 수행하고, 보안 벤치마크를 기준으로 평가하고, 결함 알림을 자동화할 수 있습니다. 프로덕션 인스턴스 또는 빌드 파이프라인에서 실행되는 Amazon Inspector는 확인된 정보가 있으면 개발자와 엔지니어에게 알림을 보냅니다. 프로그래밍 방식으로 확인된 정보에 접근할 수 있으며, 팀에게 백로그 및 버그 추적 시스템에 접근 권한을 제공할 수 있습니다. EC2 Image Builder 는 자동화된 패치 적용, AWS에서 제공하는

보안 정책 적용 및 기타 사용자 지정을 통해 서버 이미지(AMI)를 유지 관리하는 데 사용될 수 있습니다. 컨테이너를 사용할 때는 빌드 파이프라인에서 이미지 리포지토리에 대해 정기적으로 ECR Image Scanning 을 구현하여 컨테이너에서 CVE를 찾습니다.

Amazon Inspector 및 기타 도구는 존재하는 구성 및 CVE를 식별하는 데 효과적이지만, 애플리케이션 수준에서 워크로드를 테스트하려면 다른 방법이 필요합니다. Fuzzing 은 자동화를 사용하여 잘못된 형식의 데이터를 입력 필드 및 애플리케이션의 기타 영역에 주입함으로써 버그를 찾는 유명한 방법입니다.

이 모범 사례가 수립되지 않을 경우 노출되는 위험의 수준: 높음

구현 가이드

- 운영 체제 강화: 모범 사례에 맞춰 운영 체제를 구성합니다.
  - Amazon Linux 보호
  - Microsoft Windows Server 보호
- 컨테이너식 리소스 강화: 보안 모범 사례에 맞춰 컨테이너식 리소스를 구성합니다.
- AWS Lambda 모범 사례를 구현합니다.
  - AWS Lambda 모범 사례

리소스

관련 문서:

- AWS Systems Manager
- Amazon EC2 Systems Manager로 Bastion 호스트 대체
- AWS Lambda 보안 개요

관련 동영상:

- Amazon EKS에서 고도의 보안 워크로드 실행
- Securing Serverless and Container Services
- Amazon EC2 인스턴스 메타데이터 서비스에 대한 보안 모범 사례

관련 예시:

- 실습: 웹 애플리케이션 방화벽 자동 배포

SEC06-BP03 관리형 서비스 구현

공유 책임 모델의 일환으로 보안 유지 관리 태스크를 줄일 수 있도록 Amazon Relational Database Service(Amazon RDS), AWS Lambda, Amazon Elastic Container Service(Amazon ECS) 등 리소스를 관리하는 서비스를 구현합니다. 예를 들어 Amazon RDS는 관계형 데이터베이스를 설정, 운영, 확장하는 데 도움을 주고 하드웨어 프로비저닝, 데이터베이스 설정, 패치 적용, 백업 등의 관리 작업을 자동화합니다. 즉, AWS Well-Architected Framework에 설명된 다른 방법으로 애플리케이션을 보호하는 데 더 많은 시간을 할애할 수 있습니다. Lambda를 사용하면 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있으므로 인프라나 운영 체제가 아니라 코드 수준의 연결, 호출, 보안에만 집중하면 됩니다.

이 모범 사례가 수립되지 않을 경우 노출되는 위험의 수준: 보통

구현 가이드

- 사용 가능한 서비스 탐색: Amazon RDS, AWS Lambda, Amazon ECS 등 리소스를 관리하는 서비스를 탐색, 테스트 및 구현합니다.

리소스

관련 문서:

- AWS 웹 사이트
- AWS Systems Manager
- Replacing a Bastion Host with Amazon EC2 Systems Manager(Amazon EC2 Systems Manager로 Bastion 호스트 대체)
- AWS Lambda 보안 개요

관련 동영상:

- Running high-security workloads on Amazon EKS(Amazon EKS에서 고보안 워크로드 실행)
- Securing Serverless and Container Services
- Security best practices for the Amazon EC2 instance metadata service(Amazon EC2 인스턴스 메타데이터 서비스에 대한 보안 모범 사례)

관련 예시:

- 실습: AWS Certificate Manager 퍼블릭 인증서 요청

SEC06-BP04 컴퓨팅 보호 자동화

취약성 관리, 공격 대상 영역 축소, 리소스 관리 등 컴퓨팅 보호 메커니즘을 자동화합니다. 자동화를 사용하면 워크로드의 다른 측면을 보호하는 데 시간을 투자하고 인적 오류의 위험을 줄일 수 있습니다.

이 모범 사례가 수립되지 않을 경우 노출되는 위험의 수준: 보통

구현 가이드

- 구성 관리 자동화: 구성 관리 서비스 또는 도구를 사용하여 보안 구성을 자동으로 적용하고 확보합니다.
  - AWS Systems Manager
  - AWS CloudFormation
  - 실습: VPC 자동 배포
  - 실습: EC2 웹 애플리케이션 자동 배포

- Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스 자동 패칭: AWS Systems Manager Patch Manager는 보안 관련 업데이트와 기타 유형의 업데이트를 모두 사용하여 관리형 인스턴스를 패치하는 프로세스를 자동화합니다. Patch Manager를 사용하여 운영 체제와 애플리케이션 모두에 패치를 적용할 수 있습니다.
  - AWS Systems Manager Patch Manager
  - AWS Systems Manager Automation으로 다중 계정 및 다중 리전 패치 작업을 중앙 집중화

- 침입 탐지 및 차단 구현: 침입 탐지 및 차단 도구를 구현하여 인스턴스에서 악의적인 활동을 모니터링하고 중지합니다.
- AWS Partner 솔루션 고려: AWS 파트너는 고객 온프레미스 환경의 기존 제어 솔루션과 동등한 수준이거나, 동일하거나 통합된 업계 최고 수준의 수백 가지 제품을 제공합니다. 이러한 제품은 기존 AWS 서비스를 보완하여 클라우드 및 온프레미스 환경에 포괄적인 보안 아키텍처와 보다 원활한 환경을 배포할 수 있도록 해줍니다.
  - 인프라 보안

리소스

관련 문서:

- AWS CloudFormation
- AWS Systems Manager
- AWS Systems Manager Patch Manager
- AWS Systems Manager Automation으로 다중 계정 및 다중 리전 패치 작업을 중앙 집중화
- 인프라 보안
- Amazon EC2 Systems Manager로 Bastion 호스트 대체
- AWS Lambda 보안 개요

관련 동영상:

- Amazon EKS에서 고도의 보안 워크로드 실행
- Securing Serverless and Container Services
- Amazon EC2 인스턴스 메타데이터 서비스에 대한 보안 모범 사례

관련 예시:

- 실습: 웹 애플리케이션 방화벽 자동 배포
- 실습: EC2 웹 애플리케이션 자동 배포

SEC06-BP05 사용자가 원격으로 작업을 수행할 수 있도록 지원

대화형 액세스 기능을 제거하면 인적 오류의 위험과 수동 구성 또는 관리의 필요성을 줄일 수 있습니다. 예를 들어, 변경 관리 워크플로를 사용하여 코드형 인프라를 사용하는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스를 배포한 후 직접 액세스를 허용하는 대신 AWS Systems Manager와 같은 도구를 사용하거나 Bastion 호스트를 통해 Amazon EC2 인스턴스를 관리합니다. AWS Systems Manager는 자동화 워크플로, 문서 (플레이북) 및 Run Command등의 기능을 사용하여 다양한 유지 관리 및 배포 작업을 자동화할 수 있습니다. AWS CloudFormation 스택은 파이프라인에서 구축되며 AWS Management Console 또는 API를 직접 사용하지 않고도 인프라 배포 및 관리 작업을 자동화할 수 있습니다.

이 모범 사례가 수립되지 않을 경우 노출되는 위험의 수준: 낮음

구현 가이드

- 콘솔 액세스 대체: AWS Systems Manager Run Command로 인스턴스에 대한 콘솔 액세스(SSH 또는 RDP)를 교체하여 관리 태스크를 자동화합니다.

- AWS Systems Manager Run Command

리소스

관련 문서:

- AWS Systems Manager
- AWS Systems Manager Run Command
- Replacing a Bastion Host with Amazon EC2 Systems Manager(Amazon EC2 Systems Manager로 Bastion 호스트 대체)
- AWS Lambda 보안 개요

관련 동영상:

- Running high-security workloads on Amazon EKS(Amazon EKS에서 고보안 워크로드 실행)
- Securing Serverless and Container Services
- Security best practices for the Amazon EC2 instance metadata service(Amazon EC2 인스턴스 메타데이터 서비스에 대한 보안 모범 사례)

관련 예시:

- 실습: 웹 애플리케이션 방화벽 자동 배포

SEC06-BP06 소프트웨어 무결성 검증

워크로드에 사용되는 소프트웨어, 코드, 라이브러리가 신뢰할 수 있는 소스에서 온 것이며 변조되지 않았는지 검증하는 메커니즘(예: 코드 서명)을 구현합니다. 예를 들어 바이너리 및 스크립트의 코드 서명 인증서를 검토하여 작성자를 확인하고 작성자가 생성한 후에 변조되지 않았는지 확인해야 합니다. AWS Signer 는 인증서 서명과 퍼블릭 및 프라이빗 키 등의 코드 서명 수명 주기를 중앙에서 관리하여 코드의 신뢰성과 무결성을 보장하는 데 도움이 됩니다. 코드 서명에 대한 고급 패턴과 모범 사례를 사

용하는 방법은 다음에서 확인할 수 있습니다. AWS Lambda. 또한 다운로드한 소프트웨어의 체크섬을 공급자의 체크섬과 비교하면 변조되지 않았는지를 확인하는 데 도움이 될 수 있습니다.

이 모범 사례를 정립하지 않을 경우 노출되는 위험의 수준: 낮음

구현 가이드

- 메커니즘 조사: 코드 서명은 소프트웨어 무결성을 검증하는 데 사용할 수 있는 메커니즘입니다.
  - NIST: 코드 서명을 위한 보안 고려 사항

리소스

관련 문서:

- AWS Signer
- New - Code Signing, a Trust and Integrity Control for AWS Lambda(신규 - 코드 서명, AWS Lambda의 신뢰 및 무결성 제어)

데이터 보호

워크로드를 설계하려면 먼저 보안과 관련된 기본적인 관례부터 마련해야 합니다. 예를 들어 데이터 분류는 민감도에 따라 데이터를 구분하는 하나의 방법이고, 암호화는 무단 침입 사용자가 데이터를 해석하지 못하게 만들어 데이터를 보호하는 방법입니다. 이는 잘못된 취급 방지 또는 규제 의무 준수 등 목표 달성을 뒷받침하는 중요한 방법입니다.

AWS에서는 데이터 보호를 수행할 때 사용할 수 있는 여러 가지 방식이 있습니다. 다음 섹션에서는 아래와 같은 방식을 사용하는 방법을 설명합니다.

주제

- 데이터 분류
- 저장된 데이터 보호
- 전송 중 데이터 보호

데이터 분류

데이터 분류는 적절한 보호 및 보존 제어를 결정하는 데 도움이 되도록 중요도를 기준으로 조직 데이터를 분류하는 방법을 제공합니다.

모범 사례

- SEC07-BP01 워크로드 안에서 데이터 식별
- SEC07-BP02 데이터 보호 제어 정의
- SEC07-BP03 식별 및 분류 자동화
- SEC07-BP04 데이터 수명 주기 관리 정의

SEC07-BP01 워크로드 안에서 데이터 식별

워크로드가 처리하는 데이터의 유형과 분류, 관련 비즈니스 프로세스, 데이터가 저장되는 위치, 데이터 소유자를 이해하는 것이 중요합니다. 또한 워크로드의 해당 법률 및 규정 준수 요구 사항과 적용해야 하는 데이터 제어를 이해해야 합니다. 데이터 식별은 데이터 분류 여정의 첫 번째 단계입니다.

이 모범 사례 확립의 이점:

데이터 분류를 통해 워크로드 소유자는 민감한 데이터를 저장하는 위치를 식별하고 해당 데이터에 액세스하고 공유하는 방법을 결정할 수 있습니다.

데이터 분류는 다음 질문에 답변하는 것을 목표로 합니다.

- 어떤 유형의 데이터가 있나요?

  다음과 같은 데이터가 있을 수 있습니다.
  - 영업 비밀, 특허, 계약과 같은 지적 재산(IP)
  - 개인과 연결된 병력 정보가 포함된 의료 기록과 같은 보호 대상 건강 정보(PHI)
  - 이름, 주소, 생년월일, 국가 ID, 등록 번호와 같은 개인 식별 정보(PII)
  - 기본 계정 번호(PAN), 카드 소유자 이름, 만료 날짜, 서비스 코드 번호와 같은 신용 카드 데이터
  - 민감한 데이터는 어디에 저장되나요?
  - 누가 데이터에 액세스하여 수정 및 삭제할 수 있나요?
  - 데이터를 잘못 취급하지 못하도록 방지하려면 사용자 권한을 이해하는 것이 필수적입니다.
- 생성, 읽기, 업데이트, 삭제(CRUD) 작업은 누가 수행할 수 있나요?
  - 누가 데이터에 대한 권한을 관리할 수 있는지 파악하여 권한 상승 가능성을 고려합니다.
- 데이터가 의도치 않게 공개, 변경, 삭제되면 비즈니스에 어떤 영향을 미칠 수 있나요?
  - 데이터가 의도치 않게 수정, 삭제, 공개되는 경우의 위험 결과를 이해합니다.

이러한 질문에 대한 답변을 알면 다음과 같은 조치를 취할 수 있습니다.

- 민감한 데이터 범위(예: 민감한 데이터 위치 수)를 줄이고 민감한 데이터에 대한 액세스 권한을 승인된 사용자로만 제한합니다.
- 암호화, 데이터 손실 방지, 자격 증명 및 액세스 관리와 같은 적절한 데이터 보호 메커니즘 및 기술을 구현할 수 있도록 다양한 데이터 유형을 파악합니다.
- 데이터에 대한 올바른 제어 목표를 제공하여 비용을 최적화합니다.
- 데이터 유형과 양, 민감도가 다른 데이터를 서로 분리하는 방법에 대한 규제 기관 및 감사 담당자의 질문에 자신 있게 답변합니다.

이 모범 사례를 따르지 않을 경우 노출 위험도: 높음

구현 가이드

데이터 분류는 데이터의 민감도를 식별하는 행위입니다. 데이터를 쉽게 검색하고 추적할 수 있도록 태그 지정이 포함될 수 있습니다. 또한 데이터 분류는 데이터 중복을 줄여 검색 프로세스 속도를 높이면서 스토리지 및 백업 비용을 줄이는 데 도움이 될 수 있습니다.

Amazon Macie과 같은 서비스를 사용하여 민감한 데이터의 검색 및 분류를 대규모로 자동화합니다. Amazon EventBridge 및 AWS Config와 같은 다른 서비스를 사용하면 암호화되지 않은 Amazon Simple Storage Service(Amazon S3) 버킷 및 Amazon EC2 EBS 볼륨 또는 태그가 지정되지 않은 데이터 리소스와 같은 데이터 보안 문제에 대한 개선조치를 자동화할 수 있습니다. AWS 서비스 통합의 전체 목록은 EventBridge 설명서를 참조하세요.

고객 이메일, 지원 티켓, 제품 리뷰, 소셜 미디어와 같은 비정형 데이터에서 PII 감지는 자연어 처리(NLP) 서비스인 Amazon Comprehend를 사용하면 됩니다. 이 서비스는 기계 학습(ML)을 사용하여 비정형 텍스트에서 사람, 장소, 감정, 주제와 같은 인사이트와 관계를 찾습니다. 데이터 식별을 지원할 수 있는 AWS 서비스 목록은 AWS 서비스를 사용하여 PHI 및 PII 데이터를 감지하는 일반 기술을 참조하세요.

데이터 분류 및 보호를 지원하는 또 다른 방법은 AWS 리소스 태그 지정입니다. 태그 지정을 사용하면 리소스를 관리, 식별, 구성, 검색 및 필터링하는 데 사용할 수 있는 메타데이터를 AWS 리소스에 할당할 수 있습니다.

경우에 따라 전체 리소스(예: S3 버킷)에 태그를 지정하도록 선택할 수 있습니다. 특히 특정 워크로드 또는 서비스가 이미 알려진 데이터 분류의 프로세스 또는 전송을 저장해야 하는 경우에 더욱 그렇습니다.

적절한 경우 관리 및 보안 유지 관리를 쉽게 하기 위해 개별 객체 대신 S3 버킷에 태그를 지정할 수 있습니다.

구현 단계

Amazon S3 내의 민감한 데이터 감지:

1. 시작하기 전에 Amazon Macie 콘솔 및 API 작업에 액세스할 수 있는 적절한 권한이 있는지 확인합니다. 자세한 내용은 Amazon Macie 시작하기를 참조하세요.
2. 민감한 데이터가 Amazon S3에 있는 경우 Amazon Macie를 사용하여 자동화된 데이터 검색을 수행합니다.
   - Amazon Macie 시작하기 가이드를 사용하여 민감한 데이터 검색 결과에 대한 리포지토리를 구성하고 민감한 데이터에 대한 검색 작업을 생성합니다.
   - Amazon Macie를 사용하여 S3 버킷에서 민감한 정보를 미리 보는 방법

     기본적으로 Macie는 자동화된 민감한 데이터 검색을 위해 권장되는 관리형 데이터 식별자를 사용하여 객체를 분석합니다. 계정 또는 조직에 대해 자동화된 민감한 데이터 검색을 수행할 때 특정 관리형 데이터 식별자, 사용자 지정 데이터 식별자 및 허용 목록을 사용하도록 Macie를 구성하여

분석을 맞춤 조정할 수 있습니다. 특정 버킷(예: 일반적으로 AWS 로깅 데이터를 저장하는 S3 버킷)을 제외하여 분석 범위를 조정할 수 있습니다.

3. 자동화된 민감한 데이터 검색을 구성하고 사용하려면 Performing automated sensitive data discovery with Amazon Macie(Amazon Macie를 사용하여 자동화된 민감한 데이터 검색 수행)를 참조하세요.
4. Amazon Macie에 대한 자동 데이터 검색을 고려할 수도 있습니다.

Amazon RDS 내의 민감한 데이터 감지:

Amazon Relational Database Service(Amazon RDS) 데이터베이스의 데이터 검색에 대한 자세한 내용은 Macie를 사용하여 Amazon RDS 데이터베이스에 대한 데이터 분류 활성화를 참조하세요.

DynamoDB 내의 민감한 데이터 감지:

- Macie를 사용하여 DynamoDB에서 민감한 데이터 감지는 Amazon Macie를 사용하여 스캔을 위해 데이터를 Amazon S3로 내보내서 Amazon DynamoDB 테이블에서 민감한 데이터를 감지하는 방법을 설명합니다.

AWS 파트너 솔루션:

- 광범위한 AWS Partner Network 사용을 고려합니다. AWS 파트너는 AWS 서비스와 직접 통합되는 광범위한 도구 및 규정 준수 프레임워크를 보유하고 있습니다. 파트너는 조직의 요구 사항을 충족하는 데 도움이 되는 맞춤형 거버넌스 및 규정 준수 솔루션을 제공할 수 있습니다.
- 데이터 분류의 맞춤형 솔루션은 Data governance in the age of regulation and compliance requirements(규제 및 규정 준수 요구 사항 시대의 데이터 거버넌스)를 참조하세요.

AWS Organizations를 사용하여 정책을 생성하고 배포하여 조직에서 채택하는 태그 지정 표준을 자동으로 적용할 수 있습니다. 태그 정책을 사용하면 유효한 키 이름과 각 키에 유효한 값을 정의하는 규칙을 지정할 수 있습니다. 모니터링만 선택하면 기존 태그를 평가하고 정리할 수 있습니다. 태그가 선택한 표준을 준수하면 태그 정책에서 적용을 사용 설정하여 규정 미준수 태그가 생성되는 것을 방지할 수 있습니다. 자세한 내용은 AWS Organizations에서 서비스 제어 정책을 사용하여 권한 부여에 사용되는 리소스 태그 보호 및 권한이 부여된 보안 주체에 의한 것을 제외한 태그 수정 방지에 대한 예시 정책을 참조하세요.

- AWS Organizations에서 태그 정책 사용을 시작하려면 고급 태그 정책으로 이동하기 전에 태그 정책 시작하기의 워크플로를 따르는 것이 좋습니다. 전체 조직 단위(OU) 또는 조직으로 확장하기 전에 단일 계정에 간단한 태그 정책을 연결하는 효과를 이해하면 태그 정책을 준수하기 전에 태그 정책의 효

과를 볼 수 있습니다. 태그 정책 시작하기에서는 고급 정책 관련 작업에 대한 지침 링크를 제공합니다.

- 데이터 분류 백서에 나열된 데이터 분류를 지원하는 다른 AWS 서비스 및 기능을 평가하는 것이 좋습니다.

리소스

관련 문서:

- Amazon Macie 시작하기
- Amazon Macie을 사용한 자동화된 데이터 검색
- 태그 정책 시작하기
- PII 엔터티 감지

관련 블로그:

- Amazon Macie를 사용하여 S3 버킷에서 민감한 정보를 미리 보는 방법
- Performing automated sensitive data discovery with Amazon Macie(Amazon Macie를 사용하여 자동화된 민감한 데이터 검색 수행)
- Common techniques to detect PHI and PII data using AWS Services(AWS 서비스를 사용하여 PHI 및 PII 데이터를 감지하는 일반 기술)
- Detecting and redacting PII using Amazon Comprehend(Amazon Comprehend를 사용하여 PII 감지 및 교정)
- Securing resource tags used for authorization using a service control policy in AWS Organizations(AWS Organizations에서 서비스 제어 정책을 사용하여 권한 부여에 사용되는 리소스 태그 보호)
- Enabling data classification for Amazon RDS database with Macie(Macie를 사용하여 Amazon RDS 데이터베이스에 대한 데이터 분류 활성화)
- Detecting sensitive data in DynamoDB with Macie(Macie를 사용하여 DynamoDB에서 민감한 데이터 감지)
-

관련 동영상:

- Event-driven data security using Amazon Macie(Amazon Macie를 사용한 이벤트 기반 데이터 보안)

- Amazon Macie for data protection and governance(데이터 보호 및 거버넌스를 위한 Amazon Macie)
- Fine-tune sensitive data findings with allow lists(허용 목록을 사용하여 민감한 데이터 조사 결과 미세 조정)

SEC07-BP02 데이터 보호 제어 정의

분류 수준에 따라 데이터를 보호합니다. 예를 들어 관련 권장 사항을 사용하여 공개용으로 분류된 데이터를 보호하면서, 추가 제어 기능을 통해 민감한 데이터를 보호합니다.

리소스 태그, 중요도별(각 주의, 영역, 커뮤니티별로도 가능) 개별 AWS 계정, IAM 정책, AWS Organizations SCP, AWS Key Management Service(AWS KMS), AWS CloudHSM을 사용함으로써 데이터 분류 및 암호화를 통한 보호를 위한 정책을 정의하고 구현할 수 있습니다. 예를 들어 기밀 데이터를 처리하는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스 또는 매우 중요한 데이터가 포함된 S3 버킷을 사용하는 프로젝트가 있는 경우 `Project=ABC` 태그를 지정할 수 있습니다. 직속 팀만이 프로젝트 코드의 의미를 알고 있으므로 속성 기반 액세스 제어를 사용하는 것이 가능합니다. 적절한 서비스만 보안 메커니즘을 통해 중요한 콘텐츠에 액세스할 수 있도록 키 정책 및 부여를 통해 AWS KMS 암호화 키 액세스 수준을 정의할 수 있습니다. 태그를 기반으로 권한 부여 결정을 내리는 경우, 태그에 대한 권한이 AWS Organizations의 태그 정책을 사용하여 적절하게 정의되었는지 확인해야 합니다.

이 모범 사례를 정립하지 않을 경우 노출되는 위험의 수준: 높음

구현 가이드

- 데이터 식별 및 분류 스키마 정의: 데이터 식별 및 분류를 수행하여 저장한 데이터 유형과 잠재적 영향, 그리고 이 데이터에 액세스할 수 있는 사용자를 평가합니다.
  - AWS 설명서
- 사용 가능한 AWS 제어 기능 파악: 기존에 사용 중이거나 앞으로 사용하려는 AWS 서비스에 대한 보안 제어 옵션을 알아봅니다. 서비스의 설명서에 보안 섹션이 있는 경우가 많습니다.
  - AWS 설명서
- AWS 규정 준수 리소스 파악: AWS에서 제공하는 리소스를 파악합니다.
  - https://aws.amazon.com/compliance/

리소스

관련 문서:

- AWS 설명서
- 데이터 분류 백서
- Amazon Macie 시작하기
- 누락된 텍스트

관련 동영상:

- 새로운 Amazon Macie 소개

SEC07-BP03 식별 및 분류 자동화

데이터 식별 및 분류를 자동화하면 올바른 제어를 구현하는 데 도움이 될 수 있습니다. 사람이 직접 액세스하도록 하는 대신 자동화를 사용하면 인적 오류와 노출의 위험이 줄어듭니다. 기계 학습을 사용하여 AWS에서 민감한 데이터를 자동으로 검색, 분류 및 보호하는 Amazon Macie와 같은 도구를 고려해 보아야 합니다. Amazon Macie는 개인 식별 정보(PII) 또는 지적 재산과 같은 민감한 데이터를 인식하고, 이러한 데이터가 어떻게 액세스되고 이동되는지 파악할 수 있는 대시보드 및 알림을 제공합니다.

이 모범 사례가 수립되지 않을 경우 노출되는 위험의 수준: 보통

구현 가이드

- Amazon Simple Storage Service(Amazon S3) 인벤토리 사용: Amazon S3 인벤토리는 객체의 복제 및 암호화 상태를 감사하고 보고하는 데 사용할 수 있는 도구 중 하나입니다.
  - Amazon S3 인벤토리
- Amazon Macie 고려: Amazon Macie는 기계 학습을 사용하여 Amazon S3에 저장된 데이터를 자동으로 검색하고 분류합니다.
  - Amazon Macie

리소스

관련 문서:

- Amazon Macie
- Amazon S3 인벤토리
- 데이터 분류 백서

- Amazon Macie 시작하기

관련 동영상:

- 새로운 Amazon Macie 소개

SEC07-BP04 데이터 수명 주기 관리 정의

정의된 수명 주기 전략은 중요도는 물론 법률 및 조직 요구 사항을 기반으로 해야 합니다. 데이터 보존 기간, 데이터 폐기 프로세스, 데이터 액세스 관리, 데이터 변환, 데이터 공유 등의 측면을 고려해야 합니다. 데이터 분류 방법론을 선택할 때는 사용 가능성과 액세스 권한을 적절하게 절충해야 합니다. 또한 여러 액세스 수준, 그리고 각 수준에 대해 안전하면서도 쉽게 사용할 수 있는 방식을 구현하기 위한 여러 가지 방법도 고려해야 합니다. 항상 심층 방어 방식을 사용하고 데이터 그리고 데이터 변환, 삭제 또는 복사 메커니즘에 사람이 접근하는 것을 줄입니다. 예를 들어 사용자에게 애플리케이션에 대한 강력한 인증을 요구하고, 필요한 액세스 권한을 사용자보다는 애플리케이션에 부여함으로써 '한 발 떨어져서 작업'을 수행하도록 합니다. 또한 사용자가 신뢰할 수 있는 네트워크 경로에서 애플리케이션에 액세스하며, 암호 해독 키 액세스 권한이 있어야 하도록 설정합니다. 사용자에게 데이터 직접 액세스 권한을 제공하기보다는 대시보드 및 자동화된 보고와 같은 도구를 사용하여 데이터의 정보를 제공하는 것이 좋습니다.

이 모범 사례가 수립되지 않을 경우 노출되는 위험의 수준: 낮음

구현 가이드

- 데이터 유형 파악: 워크로드에서 저장하거나 처리하는 데이터 유형을 식별합니다. 이러한 데이터로는 텍스트, 이미지, 이진 데이터베이스 등이 있습니다.

리소스

관련 문서:

- 데이터 분류 백서
- Amazon Macie 시작하기

관련 동영상:

- 새로운 Amazon Macie 소개

저장된 데이터 보호

저장된 데이터 란 워크로드의 어느 기간에서든지 비휘발성 스토리지에 지속되는 모든 데이터를 의미합니다. 여기에는 블록 스토리지, 객체 스토리지, 데이터베이스, 아카이브, IoT 디바이스 그리고 데이터가 지속되는 모든 기타 스토리지 미디어가 포함됩니다. 저장된 데이터를 보호하여 암호화 및 적절한 액세스 제어가 구현될 경우 무단 액세스 위험이 감소합니다.

암호화와 토큰화는 두가지 다 중요한 데이터 보호 체계이나 구분됩니다.

토큰화 는 사용자 신용 카드 정보와 같이 중요한 정보를 나타내는 토큰을 정의할 수 있는 프로세스입니다. 토큰 자체는 아무 의미가 없어야 하며, 토큰화하는 데이터에서 파생되어서는 안 됩니다. 따라서 암호화 다이제스트를 토큰으로 사용할 수 없습니다. 토큰화 방식을 신중하게 계획하면 콘텐츠를 추가로 보호할 수 있으며 규정 준수 요구 사항을 충족할 수 있습니다. 예를 들어 신용 카드 번호 대신 토큰을 활용하는 경우 신용 카드 처리 시스템의 규정 준수 범위를 줄일 수 있습니다.

암호화 는 콘텐츠를 다시 일반 텍스트로 해독하는 데 필요한 비밀 키가 없으면 읽을 수 없는 방식으로 콘텐츠를 변환하는 방식입니다. 토큰화 및 암호화를 사용하면 정보를 효과적으로 보호할 수 있습니다. 또한 마스킹은 중요하지 않은 것으로 간주되는 데이터만 남을 때까지 데이터의 일부를 수정할 수 있는 기술입니다. 예를 들어 PCI-DSS를 사용하면 카드 번호의 마지막 네 자리가 인덱싱을 위한 규정 준수 범위 경계를 벗어나도록 할 수 있습니다.

암호화 키 사용 감사: 암호화 키 사용을 이해하고 감사하여 키에 대한 액세스 제어 메커니즘이 적절하게 구현되었는지 검증합니다. 예를 들어 AWS KMS 키를 사용하는 AWS 서비스는 AWS CloudTrail에서 모든 사용 사례를 로깅합니다. 나중에 Amazon CloudWatch Insights와 같은 도구를 사용하여 AWS CloudTrail을 쿼리함으로써 모든 키 사용이 유효한지 확인할 수 있습니다.

모범 사례

- SEC08-BP01 보안 키 관리 구현
- SEC08-BP02 저장 시 암호화 적용
- SEC08-BP03 저장 데이터 보호 자동화
- SEC08-BP04 액세스 제어 적용
- SEC08-BP05 사람들이 데이터에 쉽게 액세스할 수 없도록 하는 메커니즘 사용

SEC08-BP01 보안 키 관리 구현

보안 키 관리에는 워크로드의 저장된 데이터를 보호하는 데 필요한 주요 자료의 저장, 순환, 액세스 제어 및 모니터링이 포함됩니다.

원하는 결과: 확장 가능하고 반복 가능하며 자동화된 키 관리 메커니즘을 구축합니다. 메커니즘은 키 자료에 대한 액세스 권한을 최소한으로 제한하고 키 가용성, 기밀성 및 무결성 간에 적절한 균형을 유지하는 기능을 제공해야 합니다. 키에 대한 접근을 모니터링하고 자동화된 프로세스를 통해 키 자료를 교체해야 합니다. 키 구성 요소는 인적 자격 증명이 절대 접근할 수 없어야 합니다.

일반적인 안티 패턴:

- 암호화되지 않은 키 자료에 대한 인적 접근이 가능합니다.
- 사용자 지정 암호화 알고리즘을 생성합니다.
- 주요 자료에 액세스할 수 있는 권한이 지나치게 광범위합니다.

이 모범 사례 확립의 이점: 워크로드에 대한 보안 키 관리 메커니즘을 구축하면 무단 액세스로부터 콘텐츠를 보호하는 데 도움이 될 수 있습니다. 또한 데이터를 암호화하기 위한 규제 요구 사항이 적용될 수 있습니다. 효과적인 키 관리 솔루션은 주요 자료를 보호하기 위해 해당 규정에 맞는 기술적 메커니즘을 제공할 수 있습니다.

이 모범 사례가 확립되지 않을 경우 노출되는 위험 수준: 높음

구현 가이드

많은 규제 요구 사항 및 모범 사례에는 기본 보안 제어 수단으로 저장된 데이터의 암호화가 포함됩니다. 이러한 제어를 준수하려면 워크로드에 저장된 데이터를 암호화하는 데 사용되는 주요 자료를 안전하게 저장하고 관리하는 메커니즘이 필요합니다.

AWS은 AWS Key Management Service(AWS KMS)를 제공하여 내구성이 뛰어나고 안전하며 중복된 AWS KMS 키 스토리지를 제공합니다. 많은 AWS 서비스가 AWS KMS와 통합하여 데이터 암호화를 지원합니다. AWS KMS은 FIPS 140-2 레벨 3 검증 하드웨어 보안 모듈을 사용하여 키를 보호합니다. AWS KMS키를 일반 텍스트로 내보내는 메커니즘은 없습니다.

다중 계정 전략을 사용하여 워크로드를 배포할 때 AWS KMS 키를 사용하는 워크로드와 동일한 계정에 키를 유지하는 것이 모범 사례로 간주됩니다. 이 분산 모델에서는 AWS KMS 키 관리에 대한 책임이 애플리케이션 팀에 있습니다. 다른 사용 사례에서는 조직에서 AWS KMS 키를 중앙 집중식 계정에 저장하도록 선택할 수 있습니다. 이 중앙 집중식 구조에는 워크로드 계정이 중앙 집중식 계정에 저장된 키에 액세스하는 데 필요한 교차 계정 액세스를 가능하게 하는 추가 정책이 필요하지만 단일 키를 여러 AWS 계정에서 공유하는 사용 사례에서는 더 적합할 수 있습니다.

키 자료의 보관 위치에 관계없이 다음을 사용하여 키에 대한 액세스를 엄격하게 제어해야 합니다. 주요 정책 및 IAM 정책 키 정책은 AWS KMS 키에 대한 액세스를 제어하는 기본 방법입니다. 또한 AWS

KMS 키 부여는 사용자를 대신하여 데이터를 암호화 및 복호화하는 AWS 서비스에 대한 액세스를 제공할 수 있습니다. AWS KMS 키 액세스 제어 모범 사례를 시간을 내어 검토하세요.

암호화 키 사용을 모니터링하여 비정상적인 액세스 패턴을 탐지하는 것이 가장 좋습니다. 저장된 관리 키와 고객 AWS 관리 키를 사용하여 수행한 작업은 AWS KMS 로그인할 수 AWS CloudTrail 있으며 정기적으로 검토해야 합니다. 키 폐기 이벤트를 모니터링하는 데 특별한 주의를 기울여야 합니다. 키 자료의 우발적 또는 악의적 폐기를 방지하기 위해 키 폐기 이벤트는 키 자료를 즉시 삭제하지 않습니다. AWS KMS의 키 삭제 시도는 대기 시간이 적용 됩니다.기본값은 30일로 설정되므로 관리자는 이러한 작업을 검토하고 필요한 경우 요청을 롤백할 시간을 확보할 수 있습니다.

대부분의 AWS 서비스는 사용자에게 투명한 방식으로 사용됩니다. 관리형 키를 AWS KMS 사용할지 아니면 고객 AWS 관리형 키를 사용할지 결정하는 것만이 유일한 요구 사항입니다. 워크로드에서 데이터를 암호화하거나 복호화하는 데 AWS KMS를 직접 사용해야 하는 경우 가장 좋은 방법은 봉투 암호화를 사용하여 데이터를 보호하는 것입니다. 해당 AWS Encryption SDK는 애플리케이션에 클라이언트 측 암호화 프리미티브를 제공하여 봉투 암호화를 구현하고 AWS KMS와 통합할 수 있습니다.

구현 단계

1. 키에 적합한 키 관리 옵션 (AWS 관리형 또는 고객 관리형)을 결정합니다.
    - 사용 편의성을 위해 대부분의 서비스에 대해 AWS 소유 및 AWS 관리 키를 AWS 제공하며, 키 자료나 키 정책을 관리할 필요 없이 저장 시 암호화 기능을 제공합니다.
    - 고객 관리형 키를 사용할 때는 민첩성, 보안, 데이터 주권, 가용성 사이에서 최상의 균형을 유지할 수 있도록 기본 키 스토어를 고려하세요. 다른 사용 사례에서는 AWS CloudHSM 또는 외부 키 저장소와 함께 사용자 지정 키 저장소를 사용해야 할 수 있습니다.
2. 워크로드에 사용 중인 서비스 목록을 검토하여 서비스와의 AWS KMS 통합 방식을 파악하세요. 예를 들어, EC2 인스턴스는 암호화된 EBS 볼륨을 사용하여 해당 볼륨에서 생성된 Amazon EBS 스냅샷도 고객 관리 키를 사용하여 암호화되는지 확인하고 암호화되지 않은 스냅샷 데이터가 우발적으로 공개되는 것을 방지할 수 있습니다.
    - AWS가 AWS KMS서비스를 사용하는 방법
    - AWS 서비스가 제공하는 암호화 옵션에 대한 자세한 내용은 사용 설명서의 유휴 시 암호화 항목 또는 서비스 개발자 안내서를 참조하세요.
3. 구현AWS KMS: 다양한 AWS 서비스와 애플리케이션에서 AWS KMS 간단하게 키를 생성 및 관리하고 암호화 사용을 제어할 수 있습니다.
    - AWS Key Management Service(AWS KMS) 시작하기
    - 다음 AWS KMS 키에 대한 액세스 제어 모범 사례를 검토하세요.

4. 고려 사항AWS Encryption SDK: 애플리케이션이 클라이언트 측 데이터를 암호화해야 하는 경우 AWS Encryption SDK와 AWS KMS를 통합하세요.
   - AWS Encryption SDK
5. 지원 IAM Access Analyzer AWS KMS 키 정책이 지나치게 광범위한지 자동으로 검토하고 알립니다.
6. Security Hub을 활성화하여 잘못 구성된 키 정책, 삭제 예정 키 또는 자동 순환이 활성화되지 않은 키가 있는 경우 알림을 받습니다.
7. AWS KMS 키에 적합한 로깅 수준을 결정하세요. 읽기 전용 이벤트를 AWS KMS 포함하여 에 대한 호출이 로깅되므로 관련된 CloudTrail 로그의 양이 많아질 AWS KMS 수 있습니다.
   - 일부 조직에서는 AWS KMS 로깅 활동을 별도의 트레일로 분리하는 것을 선호합니다. 자세한 내용은 AWS KMS 개발자 가이드의 CloudTrail로 AWS KMS API 호출 로깅하기 섹션을 참조하세요.

리소스

관련 문서:

- AWS Key Management Service
- AWS 암호화 서비스 및 도구
- 암호화를 사용하여 Amazon S3 데이터 보호
- 봉투 암호화
- 디지털 주권 서약
- AWS KMS 키 조작에 대한 설명, 자체 키 가져오기, 사용자 지정 키 저장소, 암호문 이동성
- AWS Key Management Service 암호화 세부 정보

관련 동영상:

- AWS에서 암호화가 작동하는 방식
- AWS에서 블록 스토리지 보호
- AWS 데이터 보호: 잠금, 키, 서명 및 인증서 사용

관련 예시:

- 를 사용하여 고급 액세스 제어 메커니즘 구현 AWS KMS

SEC08-BP02 저장 시 암호화 적용

저장 데이터에 대한 암호화 사용을 적용해야 합니다. 암호화는 무단 액세스 또는 우발적 공개의 경우 민감한 데이터의 기밀성을 유지합니다.

원하는 결과: 프라이빗 데이터는 저장 시 기본적으로 암호화되어야 합니다. 암호화는 데이터의 기밀성을 유지하는 데 도움이 되며 의도적이거나 의도치 않은 데이터 공개 또는 유출에 대한 추가 보호 계층을 제공합니다. 암호화된 데이터는 먼저 데이터의 암호화를 해제하지 않고는 읽거나 액세스할 수 없습니다. 암호화되지 않은 상태로 저장된 모든 데이터는 인벤토리를 만들고 제어해야 합니다.

일반적인 안티 패턴:

- 기본적으로 암호화 구성을 사용하지 않습니다.
- 복호화 키에 지나치게 관대한 액세스를 제공합니다.
- 암호화 및 복호화 키의 사용을 모니터링하지 않습니다.
- 데이터를 암호화되지 않은 상태로 저장합니다.
- 데이터 용도, 유형 및 분류에 관계없이 모든 데이터에 동일한 암호화 키를 사용합니다.

이 모범 사례를 따르지 않을 경우 노출 위험도: 높음

구현 가이드

워크로드 내의 데이터 분류에 암호화 키를 매핑합니다. 이 접근 방식은 데이터에 단일 암호화 키 또는 매우 적은 수의 암호화 키를 사용할 때 지나치게 관대한 액세스로부터 데이터를 보호하는 데 도움이 됩니다(SEC07-BP01 워크로드 안에서 데이터 식별 참조).

AWS Key Management Service(AWS KMS)는 많은 AWS 서비스와 통합되어 저장 데이터를 보다 쉽게 암호화할 수 있습니다. 예를 들어 Amazon Simple Storage Service(Amazon S3)의 경우 버킷에 기본 암호화를 설정하여 새 객체가 자동으로 암호화되도록 하면 됩니다. AWS KMS를 사용할 때 데이터를 얼마나 엄격하게 제한해야 하는지 고려합니다. 기본 및 서비스 제어 AWS KMS 키는 사용자를 대신하여 AWS에서 관리하고 사용합니다. 기본 암호화 키에 대한 세분화된 액세스 권한이 필요한 민감한 데이터의 경우 고객 관리형 키(CMK)를 고려합니다. 키 정책을 사용하여 교체 및 액세스 관리를 포함하여 CMK를 완전히 제어할 수 있습니다.

또한 Amazon Elastic Compute Cloud(Amazon EC2) 및 Amazon S3는 기본 암호화 설정을 통해 암호화를 기본적으로 적용하도록 지원합니다. AWS Config 규칙를 사용하여 Amazon Elastic Block Store(Amazon EBS) 볼륨, Amazon Relational Database Service(Amazon RDS) 인스턴스 및 Amazon S3 버킷에 암호화를 사용하고 있는지 자동으로 확인할 수 있습니다.

AWS는 또한 클라이언트측 암호화 옵션을 제공하므로 데이터를 클라우드에 업로드하기 전에 암호화할 수 있습니다. AWS Encryption SDK는 봉투 암호화를 사용하여 데이터를 암호화하는 방법을 제공합니다. 래핑 키를 제공하면 AWS Encryption SDK가 암호화하는 각 데이터 객체에 대해 고유한 데이터 키를 생성합니다. 관리형 단일 테넌트 하드웨어 보안 모듈(HSM)이 필요한 경우 AWS CloudHSM을 고려합니다. AWS CloudHSM을 사용하면 FIPS 140-2 레벨 3 검증 HSM에서 암호화 키를 생성, 가져오기 및 관리할 수 있습니다. AWS CloudHSM의 일부 사용 사례에는 인증 기관(CA) 발급을 위한 프라이빗 키 보호와 Oracle 데이터베이스용 투명한 데이터 암호화(TDE) 활성화가 포함됩니다. AWS CloudHSM 클라이언트 SDK는 데이터를 AWS에 업로드하기 전에 AWS CloudHSM에 저장된 키를 사용하여 데이터 클라이언트측을 암호화할 수 있는 소프트웨어를 제공합니다. Amazon DynamoDB Encryption Client를 사용하면 DynamoDB 테이블에 업로드하기 전에 항목을 암호화하고 서명할 수도 있습니다.

구현 단계

- Amazon S3에 저장 시 암호화 적용: Amazon S3 버킷 기본 암호화를 구현합니다.

  새 Amazon EBS 볼륨에 기본 암호화 구성: Amazon EBS에서 제공하는 기본 키 또는 사용자가 생성한 키를 사용하는 옵션을 통해 새로 생성되는 모든 AWS 볼륨이 암호화된 형식으로 생성되도록 지정합니다.

  암호화된 Amazon Machine Image(AMI) 구성: 암호화가 활성화된 기존 AMI를 복사하면 루트 볼륨과 스냅샷이 자동으로 암호화됩니다.

  Amazon RDS 암호화 구성: 암호화 옵션을 사용하여 저장 시 Amazon RDS 데이터베이스 클러스터 및 스냅샷에 대한 암호화를 구성합니다.

  각 데이터 분류를 위해 적절한 보안 주체에 대한 액세스를 제한하는 정책으로 AWS KMS 키 생성 및 구성: 예를 들어 프로덕션 데이터 암호화를 위한 하나의 AWS KMS 키와 개발 또는 테스트 데이터 암호화를 위한 다른 키를 생성합니다. 다른 AWS 계정에 키 액세스를 제공할 수도 있습니다. 개발 및 프로덕션 환경에 대해 서로 다른 계정을 사용하는 것이 좋습니다. 프로덕션 환경에서 개발 계정의 아티팩트를 복호화해야 하는 경우 개발 아티팩트를 암호화하는 데 사용되는 CMK 정책을 편집하여 프로덕션 계정에 해당 아티팩트를 복호화할 수 있는 기능을 제공할 수 있습니다. 그러면 프로덕션 환경에서 프로덕션에 사용하기 위해 복호화된 데이터를 수집할 수 있습니다.

  추가 AWS 서비스에서 암호화 구성: 사용하는 다른 AWS 서비스의 경우 해당 서비스의 보안 문서를 검토하여 서비스의 암호화 옵션을 결정합니다.

리소스

관련 문서:

- AWS 암호화 도구
- AWS 설명서
- AWS Encryption SDK
- AWS KMS 암호화 세부 정보 백서
- AWS Key Management Service
- AWS 암호화 서비스 및 도구
- Amazon EBS 암호화
- Amazon EBS 볼륨의 기본 암호화
- Amazon RDS 리소스 암호화
- Amazon S3 버킷에 기본 암호화를 사용하려면 어떻게 해야 하나요?
- 암호화를 사용하여 Amazon S3 데이터 보호

관련 동영상:

- How Encryption Works in AWS(AWS에서 암호화가 작동하는 방식)
- Securing Your Block Storage on AWS(AWS에서 블록 스토리지 보호)

SEC08-BP03 저장 데이터 보호 자동화

자동화된 도구를 사용하여 저장된 데이터 제어를 지속적으로 검증하고 적용합니다. 예를 들어 암호화된 스토리지 리소스만 있는지 확인합니다. 다음을 사용하여 모든 EBS 볼륨이 암호화되었는지 검증을 자동화 할 수 있습니다. AWS Config 규칙참조. AWS Security Hub 를 사용하여 보안 표준을 기준으로 자동화된 검사를 통해 몇 가지 제어의 유효성을 확인할 수도 있습니다. 또한 AWS Config 규칙는 자동으로 규정 미준수 리소스를 수정할 수 있습니다..

이 모범 사례가 수립되지 않을 경우 노출되는 위험의 수준: 보통

구현 가이드

저장된 데이터 란 워크로드의 어느 기간에서든지 비휘발성 스토리지에 지속되는 모든 데이터를 의미합니다. 여기에는 블록 스토리지, 객체 스토리지, 데이터베이스, 아카이브, IoT 디바이스 그리고 데이터가 지속되는 모든 기타 스토리지 미디어가 포함됩니다. 저장된 데이터를 보호하여 암호화 및 적절한 액세스 제어가 구현될 경우 무단 액세스 위험이 감소합니다.

저장 시 암호화 적용: 데이터를 저장할 때 반드시 암호화를 사용하도록 해야 합니다. AWS KMS는 여러 AWS 서비스와 원활하게 통합되므로 모든 저장 데이터를 쉽게 암호화할 수 있습니다. 예를 들어

Amazon Simple Storage Service(Amazon S3)의 경우 버킷에 기본 암호화 를 설정하여 새 객체가 모두 자동으로 암호화되도록 하면 됩니다. 또한 Amazon EC2 및 Amazon S3 는 기본 암호화 설정을 통해 암호화를 기본적으로 적용하도록 지원합니다. 전용 인프라에서 AWS Managed Config Rules 를 사용하여 예를 들면 다음에 암호화를 사용하고 있는지 자동으로 검사할 수 있습니다. EBS 볼륨, Amazon Relational Database Service(Amazon RDS) 인스턴스 및 Amazon S3 버킷.

리소스

관련 문서:

- AWS 암호화 도구
- AWS Encryption SDK

관련 동영상:

- AWS에서 암호화가 작동하는 방식
- AWS에서 블록 스토리지 보호

SEC08-BP04 액세스 제어 적용

저장 데이터를 보호하려면 격리 및 버전 관리와 같은 메커니즘을 사용하여 액세스 제어를 적용하고 최소 권한 원칙을 적용합니다. 데이터에 대한 퍼블릭 액세스 권한 부여를 방지합니다.

원하는 결과: 권한이 부여된 사용자만 알아야 할 필요가 있을 때 데이터에 액세스할 수 있는지 확인합니다. 정기적인 백업 및 버전 관리를 통해 데이터를 보호하여 의도적이거나 우발적인 데이터 수정 또는 삭제를 방지합니다. 중요한 데이터를 다른 데이터와 분리하여 기밀성과 데이터 무결성을 보호합니다.

일반적인 안티 패턴:

- 민감도 요구 사항이 다르거나 분류가 다른 데이터를 함께 저장합니다.
- 복호화 키에 지나치게 관대한 권한을 사용합니다.
- 데이터를 잘못 분류합니다.
- 중요한 데이터의 자세한 백업을 유지하지 않습니다.
- 프로덕션 데이터에 대한 지속적인 액세스를 제공합니다.
- 데이터 액세스를 감사하거나 정기적으로 권한을 검토하지 않습니다.

이 모범 사례를 따르지 않을 경우 노출 위험도: 낮음

구현 가이드

액세스(최소 권한 사용), 격리, 버전 관리를 포함한 여러 제어를 사용하여 저장 데이터를 보호할 수 있습니다. 데이터에 대한 액세스는 AWS CloudTrail과 같은 탐지 메커니즘과 Amazon Simple Storage Service(Amazon S3) 액세스 로그와 같은 서비스 수준 로그를 사용하여 감사해야 합니다. 공개적으로 액세스할 수 있는 데이터의 인벤토리를 만들고 시간이 지남에 따라 공개적으로 사용 가능한 데이터의 양을 줄이기 위한 계획을 수립해야 합니다.

Amazon S3 Glacier 저장소 잠금 및 Amazon S3 객체 잠금은 Amazon S3의 객체에 대한 필수 액세스 제어를 제공합니다. 규정 준수 옵션으로 저장소 정책을 잠그면 루트 사용자도 잠금이 만료되기 전까지는 변경할 수 없습니다.

구현 단계

- 액세스 제어 적용: 암호화 키 액세스를 포함하여 최소 권한을 사용하는 액세스 제어를 적용합니다.
- 다양한 분류 수준에 따라 데이터 분리: 데이터 분류 수준에 서로 다른 AWS 계정를 사용하고, AWS Organizations를 사용하여 해당 계정을 관리합니다.
- AWS Key Management Service(AWS KMS) 정책 검토: AWS KMS 정책에서 부여된 액세스 수준을 검토합니다.
- Amazon S3 버킷 및 객체 권한 검토: S3 버킷 정책에서 부여된 액세스 수준을 주기적으로 검토합니다. 공개적으로 읽을 수 있는 버킷이나 쓸 수 있는 버킷을 사용하지 않는 것이 모범 사례입니다. AWS Config를 사용하여 공개적으로 사용 가능한 버킷을 감지하고 Amazon CloudFront를 사용하여 Amazon S3에서 콘텐츠를 제공하는 것이 좋습니다. 퍼블릭 액세스를 허용하면 안 되는 버킷은 퍼블릭 액세스가 되지 않도록 적절히 구성되어 있는지 확인합니다. 기본적으로 모든 S3 버킷은 프라이빗 버킷이며 명시적으로 액세스 권한이 부여된 사용자만 액세스할 수 있습니다.
- AWS IAM Access Analyzer 사용: IAM Access Analyzer는 Amazon S3 버킷을 분석하고 S3 정책이 외부 엔터티에 대한 액세스 권한을 부여할 때 조사 결과를 생성합니다.
- 적절한 경우 Amazon S3 버전 관리 및 객체 잠금을 활성화합니다.
- Amazon S3 인벤토리 사용: Amazon S3 인벤토리를 사용하면 S3 객체의 복제 및 암호화 상태를 감사하고 보고할 수 있습니다.
- Amazon EBS 및 AMI 공유 권한 검토: 권한을 공유하면 워크로드 외부의 AWS 계정과 이미지 및 볼륨을 공유할 수 있습니다.
- AWS Resource Access Manager 공유를 주기적으로 검토하여 리소스를 계속 공유해야 하는지 여부를 결정합니다. Resource Access Manager를 사용하면 Amazon VPC 내에서 AWS 네트워크 방화벽 정책, Amazon Route 53 확인자 규칙, 서브넷과 같은 리소스를 공유할 수 있습니다. 공유 리소스를 정기적으로 감사하고 더 이상 공유할 필요가 없는 리소스 공유를 중지합니다.

리소스

관련 모범 사례:

- SEC03-BP01 액세스 요구 사항 정의
- SEC03-BP02 최소 권한 액세스 부여

관련 문서:

- AWS KMS 암호화 세부 정보 백서
- Amazon S3 리소스에 대한 액세스 권한 관리 소개
- AWS KMS 리소스에 대한 액세스 권한 관리 개요
- AWS Config 규칙
- Amazon S3 + Amazon CloudFront: 클라우드 최적의 조합
- 버전 관리 사용
- Amazon S3 객체 잠금을 사용한 객체 잠금
- Amazon EBS 스냅샷 공유
- 공유 AMI
- Amazon S3에서 단일 페이지 애플리케이션 호스팅

관련 동영상:

- Securing Your Block Storage on AWS(AWS에서 블록 스토리지 보호)

SEC08-BP05 사람들이 데이터에 쉽게 액세스할 수 없도록 하는 메커니즘 사용

정상적인 운영 상황에서 모든 사용자가 민감한 데이터와 시스템에 직접 액세스하지 못하도록 합니다. 예를 들어 변경 관리 워크플로를 사용하여 직접 액세스를 허용하는 대신 도구나 Bastion 호스트를 사용하여 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스를 관리할 수 있습니다. 이렇게 하려면 AWS Systems Manager Automation을 사용할 수 있으며, 이 서비스는 작업을 수행할 때 사용하는 단계가 포함된 자동화 문서 를 사용합니다. 이러한 문서는 소스 제어에 저장되고, 실행하기 전에 피어 검토를 받고, 철저한 테스트를 받아 셸 액세스와 비교해 위험을 최소화할 수 있습니다. 비즈니스 사용자에게는 데이터 스토어에 대한 직접적인 액세스 대신 대시보드를 제공하여 쿼리를 실행하게 할 수 있

습니다. CI/CD 파이프라인이 사용되지 않는 경우, 정상적으로 비활성화된 브레이크-글라스 액세스 메커니즘을 적절하게 제공하기 위해 필요한 제어 및 프로세스를 결정합니다.

이 모범 사례가 수립되지 않을 경우 노출되는 위험의 수준: 낮음

구현 가이드

- 사람들이 데이터에 쉽게 액세스할 수 없도록 하는 메커니즘 구현: 이러한 메커니즘에는 직접 쿼리하는 대신 Amazon QuickSight와 같은 대시보드를 사용하여 사용자에게 데이터를 표시하는 방식이 포함됩니다.
  - Amazon QuickSight
- 구성 관리 자동화: 구성 관리 서비스 또는 도구를 사용하여 원격으로 작업을 수행하고 보안 구성을 자동으로 적용하고 확인합니다. 배스천 호스트를 사용하거나 EC2 인스턴스에 직접 액세스하지 않습니다.
  - AWS Systems Manager
  - AWS CloudFormation
  - AWS 기반 AWS CloudFormation 템플릿용 CI/CD 파이프라인

리소스

관련 문서:

- AWS KMS 암호화 세부 정보 백서

관련 동영상:

- How Encryption Works in AWS(AWS에서 암호화가 작동하는 방식)
- Securing Your Block Storage on AWS(AWS에서 블록 스토리지 보호)

전송 중 데이터 보호

전송 중인 데이터 는 시스템 간에 전송되는 모든 데이터를 의미합니다. 여기에는 워크로드 내 리소스 간의 통신과 다른 서비스 및 최종 사용자 간의 통신이 포함됩니다. 전송 중 데이터를 적절한 수준으로 보호하면 워크로드 데이터의 기밀성과 무결성을 보호할 수 있습니다.

VPC 또는 온프레미스 위치 간 데이터 보안: 전용 인프라에서 AWS PrivateLink 를 사용하여 AWS에서 호스팅되는 서비스에 대해 Amazon Virtual Private Cloud(Amazon VPC) 또는 온프레미스 연결 간에

안전한 프라이빗 네트워크 연결을 생성할 수 있습니다. AWS 서비스, 서드 파티 서비스, 다른 AWS 계정의 서비스가 사용자의 프라이빗 네트워크에 있는 것처럼 액세스할 수 있습니다. AWS PrivateLink를 사용하면 인터넷 게이트웨이 또는 NAT가 필요 없이 중복 IP CIDR을 통해 여러 계정의 서비스에 액세스할 수 있습니다. 또한 방화벽 규칙, 경로 정의, 라우팅 테이블을 구성할 필요도 없습니다. 트래픽은 Amazon 백본에 유지되며 인터넷을 통해 이동하지 않으므로 데이터가 보호됩니다. HIPAA 및 EU/US Privacy Shield와 같은 업계 특정 규정을 준수하는 상태를 유지할 수 있습니다. AWS PrivateLink는 서드 파티 솔루션과 원활하게 호환되어 간소화된 글로벌 네트워크를 생성하므로 사용자는 클라우드로의 마이그레이션 속도를 높이고 지원되는 AWS 서비스를 활용할 수 있습니다.

모범 사례

- SEC09-BP01 보안 키 및 인증서 관리 구현
- SEC09-BP02 전송 중 데이터 암호화 적용
- SEC09-BP03 무단 데이터 침입 탐지 자동화
- SEC09-BP04 네트워크 통신 인증

SEC09-BP01 보안 키 및 인증서 관리 구현

전송 계층 보안(TLS) 인증서는 인터넷과 프라이빗 네트워크에서 네트워크 통신을 보호하고 웹사이트, 리소스 및 워크로드의 ID를 설정하는 데 사용됩니다.

원하는 결과: 퍼블릭 키 인프라(PKI)에서 인증서를 프로비저닝, 배포, 저장 및 갱신할 수 있는 보안 인증서 관리 시스템 보안 키 및 인증서 관리 메커니즘은 인증서 개인 키 자료가 공개되는 것을 방지하고 정기적으로 인증서를 자동 갱신합니다. 또한 다른 서비스와 통합하여 워크로드 내부의 머신 리소스에 대한 보안 네트워크 통신 및 ID를 제공합니다. 키 구성 요소는 인적 자격 증명이 절대 접근할 수 없어야 합니다.

일반적인 안티 패턴:

- 인증서 배포 또는 갱신 프로세스 중에 수동 단계 수행
- 프라이빗 CA를 설계할 때 인증 기관(CA) 계층 구조에 충분히 주의를 기울이지 않음
- 퍼블릭 리소스에 자체 서명된 인증서 사용

이 모범 사례 확립의 이점:

- 자동 배포 및 갱신을 통해 인증서 관리 간소화
- TLS 인증서를 사용하여 전송 중 데이터의 암호화 장려

- 인증 기관이 취한 인증서 작업의 보안 및 감사 가능성 향상
- CA 계층 구조의 여러 계층에서 관리 업무 구성

이 모범 사례가 확립되지 않을 경우 노출되는 위험 수준: 높음

구현 가이드

최신 워크로드는 TLS와 같은 PKI 프로토콜을 사용하는 암호화된 네트워크 통신을 광범위하게 사용합니다. PKI 인증서 관리는 복잡할 수 있지만 자동화된 인증서 프로비저닝, 배포 및 갱신을 통해 인증서 관리와 관련된 마찰을 줄일 수 있습니다.

AWS는 범용 PKI 인증서를 관리하기 위해 AWS Certificate Manager 및 AWS Private Certificate Authority(AWS Private CA)의 두 가지 서비스를 제공합니다. ACM은 고객이 퍼블릭 워크로드와 프라이빗 AWS 워크로드 모두에서 사용할 인증서를 프로비전, 관리 및 배포하는 데 사용하는 기본 서비스입니다. ACM은 AWS Private CA를 사용하여 인증서를 발급하고 다른 많은 AWS 관리형 서비스와 통합하여 워크로드에 보안 TLS 인증서를 제공합니다.

AWS Private CA를 사용하면 자체 루트의 CA 또는 하위 CA를 설정하고 API를 통해 TLS 인증서를 발급할 수 있습니다. 이러한 종류의 인증서는 TLS 연결의 클라이언트 측에서 신뢰 체인을 제어하고 관리하는 시나리오에서 사용할 수 있습니다. TLS 사용 사례 외에도 AWS Private CA를 사용하면 사용자 지정 템플릿으로 Kubernetes 포드, Matter 디바이스 제품 증명, 코드 서명 및 기타 사용 사례에 인증서를 발급할 수 있습니다. 또한 IAM Roles Anywhere 를 사용하여 Private CA에서 서명한 X.509 인증서를 발급한 온프레미스 워크로드에 임시 IAM 보안 인증을 제공할 수 있습니다.

ACM 및 AWS Private CA 외에도 AWS IoT Core 는 PKI 인증서를 IoT 디바이스에 프로비전, 관리 및 배포하기 위한 전문 지원을 제공합니다. AWS IoT Core는 대규모 공개 키 인프라에 적용할 수 있는 IoT 디바이스 온보딩용 전문 메커니즘을 제공합니다.

프라이빗 CA 계층 구조 설정 시 고려 사항

프라이빗 CA를 설정해야 하는 경우 CA 계층 구조를 미리 적절하게 설계할 수 있도록 특별히 주의를 기울이는 것이 중요합니다. 프라이빗 CA 계층 구조를 만들 때는 CA 계층 구조의 각 수준을 별도의 AWS 계정에 배포하는 것이 좋습니다. 이 의도적인 단계는 CA 계층 구조의 각 수준에 대한 노출 영역을 줄여 CloudTrail 로그 데이터에서 이상 징후를 더 쉽게 발견하고 계정 중 하나에 대한 무단 액세스가 발생할 경우 액세스 또는 영향 범위를 줄일 수 있습니다. 루트 CA는 별도의 계정에 있어야 하며 하나 이상의 중간 CA 인증서를 발급하는 데만 사용해야 합니다.

그런 다음 루트 CA 계정과 분리된 계정에 하나 이상의 중간 CA를 생성하여 최종 사용자, 디바이스 또는 기타 워크로드에 대한 인증서를 발급합니다. 마지막으로 루트 CA에서 중간 CA로 인증서를 발급합

니다. 그러면 중간 CA가 최종 사용자나 디바이스에 인증서를 발급합니다. 복원력 계획, 교차 리전 복제, 조직 전반의 CA 공유 등을 포함하여 CA 배포 계획 및 CA 계층 설계에 대한 자세한 내용은 AWS Private CA 배포 계획하기를 참조하십시오.

구현 단계

1. 사용 사례에 필요한 관련 AWS 서비스 확인:
   - 많은 사용 사례에서 다음을 사용하여 기존 AWS 공개 키 인프라를 활용할 수 있습니다. AWS Certificate Manager. ACM는 웹 서버나 로드 밸런서용 또는 공개적으로 신뢰할 수 있는 인증서를 위한 기타 용도로 TLS 인증서를 배포하는 데 사용할 수 있습니다.
   - 자체 사설 인증 기관 계층 구조를 설정해야 하거나 내보낼 수 있는 인증서에 액세스해야 하는 경우 AWS Private CA 를 고려하세요. 그런 다음 ACM을 사용하여 다양한 유형의 최종 엔티티 인증서 (AWS Private CA 사용)를 발급할 수 있습니다.
   - 내장된 사물 인터넷(IoT) 디바이스에 대규모로 인증서를 프로비저닝해야 하는 사용 사례의 경우에는 AWS IoT Core를 고려해 보십시오.
2. 가능한 경우 자동 인증서 갱신 구현:
   - 다음을 사용하십시오. ACM 관리형 갱신 (통합 AWS 관리형 서비스와 함께 ACM 발급 인증서에 대해 사용)
3. 로깅 및 감사 트레일 설정:
   - 또한 CloudTrail 로그 를 활성화하여 인증 기관이 있는 계정에 대한 액세스를 추적합니다. CloudTrail에서 로그 파일 무결성 검증을 구성하여 로그 데이터의 신뢰성을 확인하는 것이 좋습니다.
   - 프라이빗 CA가 발급하거나 해지한 인증서가 기재된 감사 보고서 를 주기적으로 생성 및 검토합니다. 이러한 보고서는 S3 버킷으로 내보낼 수 있습니다.
   - 프라이빗 CA를 배포할 때는 인증서 폐기 목록(CRL)을 저장할 S3 버킷도 설정해야 합니다. 워크로드 요구 사항에 따라 이 S3 버킷을 구성하는 방법에 대한 지침은 CRL(인증서 폐기 목록) 계획을 참조하십시오.

리소스

관련 모범 사례:

- SEC02-BP02 임시 보안 인증 정보 사용
- SEC08-BP01 보안 키 관리 구현
- SEC09-BP04 네트워크 통신 인증

관련 문서:

- AWS에서 전체 프라이빗 인증서 인프라를 호스팅하고 관리하는 방법
- 자동차 및 제조업에서 엔터프라이즈 규모의 ACM Private CA 계층을 보호하는 방법법
- Private CA 모범 사례
- AWS RAM을 사용하여 ACM Private CA 교차 계정을 공유하는 방법

관련 동영상:

- AWS Certificate Manager Private CA 활성화(워크숍)

관련 예시:

- Private CA workshop
- IOT 디바이스 관리 워크숍 (디바이스 프로비저닝 포함)

관련 도구:

- AWS Private CA를 사용하기 위한 Kubernetes 인증서 관리자 플러그인

SEC09-BP02 전송 중 데이터 암호화 적용

조직, 법률 및 규정 준수 요구 사항을 충족할 수 있도록 조직의 정책, 규제 의무 및 표준에 따라 정의된 암호화 요구 사항을 적용합니다. 민감한 데이터를 Virtual Private Cloud(VPC) 외부로 전송할 때 암호화된 프로토콜만 사용합니다. 암호화는 데이터가 신뢰할 수 없는 네트워크로 전송되는 경우에도 데이터 기밀성을 유지하는 데 도움이 됩니다.

원하는 결과: 모든 데이터는 보안 TLS 프로토콜 및 암호 그룹을 사용하여 전송 중 암호화되어야 합니다. 데이터에 대한 무단 액세스를 완화하려면 리소스와 인터넷 간의 네트워크 트래픽을 암호화해야 합니다. 가능한 경우 내부 AWS 환경 내의 네트워크 트래픽은 TLS를 사용하여 암호화해야 합니다. AWS 내부 네트워크는 기본적으로 암호화되며 권한이 없는 당사자가 트래픽을 생성하는 모든 리소스(예: Amazon EC2 인스턴스 및 Amazon ECS 컨테이너)에 대한 액세스 권한을 획득하지 않는 한 VPC 내의 네트워크 트래픽을 스푸핑하거나 스니핑할 수 없습니다. IPsec 가상 프라이빗 네트워크(VPN)를 사용하여 네트워크 간 트래픽을 보호하는 것이 좋습니다.

일반적인 안티 패턴:

- 사용 중단된 버전의 SSL, TLS 및 암호 그룹 구성 요소(예: SSL v3.0, 1024비트 RSA 키 및 RC4 암호)를 사용합니다.
- 퍼블릭 리소스에서 암호화되지 않은(HTTP) 트래픽을 허용합니다.
- 만료되기 전에 X.509 인증서를 모니터링하고 교체하지 않습니다.
- TLS에 자체 서명된 X.509 인증서를 사용합니다.

이 모범 사례를 따르지 않을 경우 노출 위험도: 높음

구현 가이드

AWS 서비스는 통신에 TLS를 사용하는 HTTPS 엔드포인트를 제공하여 AWS API와 통신할 때 전송 중 암호화 기능을 제공합니다. HTTP와 같은 안전하지 않은 프로토콜은 보안 그룹을 사용하여 VPC에서 감사 및 차단할 수 있습니다. 또한 HTTP 요청은 Amazon CloudFront 또는 Application Load Balancer에서 HTTPS로 자동 리디렉션될 수 있습니다. 컴퓨팅 리소스를 완전하게 제어하여 서비스 간에 전송 중 암호화를 구현할 수 있습니다. 또한 외부 네트워크 또는 AWS Direct Connect로부터 특정 VPC로의 VPN 연결을 사용하여 트래픽을 쉽게 암호화할 수도 있습니다. 2023년 6월에 AWS에서 TLS 1.0 및 1.1 사용을 중단하므로 클라이언트가 TLS 1.2 이상을 사용하여 AWS API를 호출하는지 확인합니다. 특별한 요구 사항이 있는 경우 AWS Marketplace에서 타사 솔루션을 사용할 수 있습니다.

구현 단계

- 전송 중 암호화 적용: 정의된 암호화 요구 사항은 최신 표준 및 모범 사례를 토대로 하고 보안 프로토콜만 허용해야 합니다. 예를 들어 Application Load Balancer 또는 Amazon EC2 인스턴스로의 HTTPS 프로토콜을 허용하는 보안 그룹만 구성합니다.
- 엣지 서비스에서 보안 프로토콜 구성: Amazon CloudFront로 HTTPS를 구성하고 보안 태세 및 사용 사례에 적합한 보안 프로필을 사용합니다.
- 외부 연결에 VPN 사용: 데이터 프라이버시와 무결성을 모두 제공할 수 있도록 지점 간 또는 네트워크 간 연결에 IPsec VPN을 사용하는 것이 좋습니다.
- 로드 밸런서에서 보안 프로토콜 구성: 리스너에 연결할 클라이언트가 지원하는 가장 강력한 암호 그룹을 제공하는 보안 정책을 선택합니다. Application Load Balancer에 대한 HTTPS 리스너를 생성합니다..
- Amazon Redshift에서 보안 프로토콜 구성: 클러스터가 보안 소켓 계층(SSL) 또는 전송 계층 보안(TLS) 연결을 요구하도록 구성합니다.
- 보안 프로토콜 구성: AWS 서비스 설명서를 검토하여 전송 중 암호화 기능을 확인합니다.
- Amazon S3 버킷에 업로드할 때 보안 액세스 구성: Amazon S3 버킷 정책 제어를 사용하여 데이터에 대한 보안 액세스를 적용합니다.

- AWS Certificate Manager 사용 고려: ACM을 사용하면 AWS 서비스와 함께 사용할 퍼블릭 TLS 인증서를 프로비저닝, 관리 및 배포할 수 있습니다.
- 프라이빗 PKI 요구 사항에 AWS Private Certificate Authority 사용 고려: AWS Private CA를 사용하면 프라이빗 인증 기관(CA) 계층 구조를 생성하여 암호화된 TLS 채널을 생성하는 데 사용할 수 있는 최종 엔터티 X.509 인증서를 발급할 수 있습니다.

리소스

관련 문서:

- AWS 설명서
- CloudFront와 함께 HTTPS 사용
- AWS Virtual Private Network를 사용하여 원격 네트워크에 VPC 연결
- Application Load Balancer에 대한 HTTPS 리스너 생성
- 자습서: Amazon Linux 2에서 SSL/TLS 구성
- SSL/TLS를 사용하여 DB 인스턴스에 대한 연결 암호화
- 연결을 위한 보안 옵션 구성

SEC09-BP03 무단 데이터 침입 탐지 자동화

Amazon GuardDuty 등의 도구를 사용하여 의심스러운 활동 또는 정의된 경계 외부로 데이터를 이전하려는 시도를 자동으로 감지합니다. 예를 들어, GuardDuty는 다음을 사용하여 일반적이지 않은 Amazon Simple Storage Service(Amazon S3) 읽기 활동을 감지할 수 있습니다. Exfiltration:S3/AnomalousBehavior 결과. GuardDuty 외에도 네트워크 트래픽 정보를 캡처하는 Amazon VPC 흐름 로그를 Amazon EventBridge와 함께 사용하여 비정상적 연결(성공한 연결과 거부된 연결 모두)을 탐지할 수 있습니다. Amazon S3 Access Analyzer 는 Amazon S3 버킷에서 누가 어떤 데이터에 액세스할 수 있는지를 평가하는 데 도움이 될 수 있습니다.

이 모범 사례를 정립하지 않을 경우 노출되는 위험의 수준: 보통

구현 가이드

- 무단 데이터 탐지 자동화: 도구 또는 감지 메커니즘을 사용하여 정의된 경계 외부로 데이터를 이동하려는 시도를 직접 감지합니다. 예를 들어 알 수 없는 호스트로 데이터를 복사하는 데이터베이스 시스템을 감지할 수 있습니다.

  - VPC 흐름 로그
- Amazon Macie 고려: Amazon Macie는 기계 학습 및 패턴 일치를 사용하여 AWS에서 민감한 데이터를 검색하고 보호하는 완전관리형 데이터 보안 및 데이터 개인 정보 보호 서비스입니다.
  - Amazon Macie

리소스

관련 문서:

- VPC 흐름 로그
- Amazon Macie

SEC09-BP04 네트워크 통신 인증

TLS(전송 계층 보안) 또는 IPsec과 같은 인증을 지원하는 프로토콜을 사용하여 통신의 자격 증명을 확인합니다.

서비스, 애플리케이션 또는 사용자 간에 통신할 때마다 안전하고 인증된 네트워크 프로토콜을 사용하도록 워크로드를 설계합니다. 인증 및 권한 부여를 지원하는 네트워크 프로토콜을 사용하면 네트워크 흐름을 더 강력하게 제어할 수 있고 무단 액세스의 영향을 줄일 수 있습니다.

원하는 결과: 워크로드에서 잘 정의된 데이터 영역과 컨트롤 플레인의 트래픽이 서비스 간에 흐릅니다. 트래픽 흐름은 기술적으로 가능한 경우 인증되고 암호화된 네트워크 프로토콜을 사용합니다.

일반적인 안티 패턴:

- 암호화되지 않았거나 인증되지 않은 트래픽이 워크로드 내에 흐릅니다.
- 여러 사용자 또는 엔터티가 보안 인증 정보를 재사용합니다.
- 액세스 제어 메커니즘으로 네트워크 제어에만 의존합니다.
- 업계 표준 인증 메커니즘에 의존하지 않고 사용자 지정 인증 메커니즘을 구축합니다.
- 서비스 구성 요소 또는 VPC의 다른 리소스 간에 지나치게 허용적인 트래픽이 흐릅니다.

이 모범 사례 확립의 이점:

- 무단 액세스의 영향 범위를 워크로드의 한 부분으로 제한합니다.
- 작업이 인증된 엔터티에 의해서만 수행되도록 더 높은 수준의 보장을 제공합니다.

- 의도된 데이터 전송 인터페이스를 명확하게 정의하고 적용함으로써 서비스의 분리를 개선합니다.
- 요청 어트리뷰션 및 잘 정의된 통신 인터페이스를 통해 모니터링, 로깅 및 인시던트 대응을 개선합니다.
- 네트워크 제어와 인증 및 권한 제어를 결합하여 워크로드에 대한 심층 방어를 제공합니다.

이 모범 사례를 따르지 않을 경우 노출되는 위험 수준: 낮음

구현 가이드

워크로드의 네트워크 트래픽 패턴은 두 가지 범주로 분류할 수 있습니다.

- 동서 트래픽은 워크로드를 구성하는 서비스 간의 트래픽 흐름을 나타냅니다.
- 남북 트래픽은 워크로드와 소비자 간의 트래픽 흐름을 나타냅니다.

남북 트래픽을 암호화하는 것이 일반적이지만 인증된 프로토콜을 사용하여 동서 트래픽을 보호하는 것은 흔하지 않습니다. 최신 보안 관행에서는 네트워크 설계만으로는 두 엔터티 간에 신뢰할 수 있는 관계를 부여하지 않을 것을 권장합니다. 두 서비스가 공통 네트워크 경계 내에 있을 수 있는 경우에도 이러한 서비스 간의 통신을 암호화 및 인증하고 권한을 부여하는 것이 가장 좋습니다.

예를 들어, AWS 서비스 API는 요청이 시작된 네트워크와 상관없이 AWS Signature Version 4(SigV4) 서명 프로토콜을 사용하여 호출자를 인증합니다. 이 인증을 통해 AWS API는 작업을 요청한 자격 증명을 확인할 수 있으며, 그런 다음 해당 자격 증명을 정책과 결합하여 권한 부여 결정을 내려 작업의 허용 여부를 결정할 수 있습니다.

Amazon VPC Lattice 및 Amazon API Gateway 등의 서비스를 통해 동일한 SigV4 서명 프로토콜을 사용하여 자체 워크로드의 동서 트래픽에 인증 및 권한 부여를 추가할 수 있습니다. AWS 환경 외부의 리소스가 SigV4 기반 인증 및 권한 부여를 요구하는 서비스와 통신해야 하는 경우 AWS 리소스가 아닌 다른 리소스에 AWS Identity and Access Management(IAM) Roles Anywhere를 사용하여 임시 AWS 보안 인증 정보를 얻을 수 있습니다. 이러한 보안 인증 정보를 사용하여 액세스 권한 부여에 SigV4를 사용하는 서비스에 대한 요청에 서명할 수 있습니다.

동서 트래픽을 인증하는 또 다른 일반적인 메커니즘은 TLS 상호 인증(mTLS)입니다. 많은 사물 인터넷(IoT), B2B 애플리케이션 및 마이크로서비스는 mTLS를 사용하여 클라이언트 및 서버 측 X.509 인증서를 모두 사용하여 TLS 통신 양측의 자격 증명을 확인합니다. 이러한 인증서는 AWS Private Certificate Authority(AWS Private CA)에서 발급할 수 있습니다. Amazon API Gateway 및 AWS App Mesh 등의 서비스를 사용하여 워크로드 간 또는 워크로드 내 통신을 위한 mTLS 인증을 제공할 수 있습니다. mTLS는 TLS 통신 양쪽에 대한 인증 정보를 제공하지만 권한 부여 메커니즘을 제공하지는 않습니다.

마지막으로 OAuth 2.0과 OpenID Connect(OIDC)는 일반적으로 사용자의 서비스 액세스를 제어하는 데 사용되는 프로토콜이지만 이제는 서비스 간 트래픽에서도 널리 사용되고 있습니다. API Gateway는 JSON 웹 토큰(JWT) 권한 부여자를 제공하여 워크로드가 OIDC 또는 OAuth 2.0 자격 증명 제공업체에서 발급한 JWT를 사용하여 API 경로에 대한 액세스를 제한할 수 있도록 합니다. OAuth2 범위는 기본 권한 부여 결정을 위한 소스로 사용될 수 있지만, 애플리케이션 계층에서 여전히 권한 부여 검사를 구현해야 하며, OAuth2 범위만으로는 더 복잡한 권한 부여 요구 사항을 지원할 수 없습니다.

구현 단계

- 워크로드 네트워크 흐름 정의 및 문서화: 심층 방어 전략을 구현하기 위한 첫 번째 단계는 워크로드의 트래픽 흐름을 정의하는 것입니다.
  - 워크로드를 구성하는 여러 서비스 간에 데이터가 전송되는 방식을 명확하게 정의하는 데이터 흐름도를 만듭니다. 이 다이어그램은 인증된 네트워크 채널을 통해 이러한 흐름을 적용하는 첫 번째 단계입니다.
  - 개발 및 테스트 단계에서 워크로드를 계측하여 데이터 흐름도가 런타임 시 워크로드의 동작을 정확하게 반영하는지 확인합니다.
  - 데이터 흐름도는 SEC01-BP07 위협 모델을 사용하여 위협 식별 및 완화 조치의 우선순위 지정에 설명된 대로 위협 모델링 연습을 수행할 때도 유용할 수 있습니다.
- 네트워크 제어 설정: 데이터 흐름에 맞게 네트워크 제어를 설정할 수 있는 AWS 기능을 고려합니다. 네트워크 경계가 유일한 보안 제어가 되어서는 안 되지만, 네트워크 경계는 워크로드를 보호하기 위한 심층 방어 전략의 한 계층을 제공합니다.
  - 보안 그룹을 사용하여 리소스 간 데이터 흐름을 설정, 정의 및 제한합니다.
  - AWS 서비스 및 AWS PrivateLink를 지원하는 타사 서비스와 통신하는 데 AWS PrivateLink를 사용하는 것을 고려해 보세요. AWS PrivateLink 인터페이스 엔드포인트를 통해 전송된 데이터는 AWS 네트워크 백본 내에 머물며 퍼블릭 인터넷을 통과하지 않습니다.
- 워크로드의 서비스 전반에 인증 및 권한 부여 구현: 워크로드에 인증되고 암호화된 트래픽 흐름을 제공하는 데 가장 적합한 AWS 서비스 세트를 선택합니다.
  - 서비스 간 통신 보안을 위해 Amazon VPC Lattice를 고려하세요. VPC Lattice는 SigV4 인증을 인증 정책과 결합하여 서비스 간 액세스를 제어할 수 있습니다.
  - mTLS를 사용한 서비스 간 통신의 경우 API Gateway 또는 App Mesh를 고려해 보세요. AWS Private CA는 mTLS와 함께 사용할 인증서를 발급할 수 있는 사설 CA 계층 구조를 설정하는 데 사용할 수 있습니다.
  - OAuth 2.0 또는 OIDC를 사용하여 서비스와 통합할 때는 JWT 권한 부여자를 사용하여 API Gateway를 사용하는 것을 고려해 보세요.

  - 워크로드와 IoT 디바이스 간 통신의 경우 네트워크 트래픽 암호화 및 인증을 위한 여러 옵션을 제공하는 AWS IoT Core를 고려해 보세요.
- 무단 액세스 모니터링: 의도하지 않은 통신 채널, 보호된 리소스에 대한 무단 액세스 시도 및 기타 부적절한 액세스 패턴을 지속적으로 모니터링합니다.
  - 서비스에 대한 액세스를 관리하는 데 VPC Lattice를 사용하는 경우 VPC Lattice 액세스 로그를 활성화하고 모니터링하는 것을 고려해 보세요. 이러한 액세스 로그에는 요청 엔티티에 대한 정보, 소스 및 대상 VPC를 비롯한 네트워크 정보, 요청 메타데이터가 포함됩니다.
  - VPC 흐름 로그를 활성화하여 네트워크 흐름의 메타데이터를 캡처하고 정기적으로 이상 징후를 검토하는 것을 고려해 보세요.
  - AWS 보안 인시던트 대응 안내서 및 AWS Well-Architected Framework의 인시던트 대응 섹션에서 보안 인시던트 계획, 시뮬레이션, 대응에 대해 자세히 알아보세요.

리소스

관련 모범 사례:

- SEC03-BP07 퍼블릭 및 크로스 계정 액세스 분석
- SEC02-BP02 임시 보안 인증 정보 사용
- SEC01-BP07 위협 모델을 사용하여 위협 식별 및 완화 조치의 우선순위 지정

관련 문서:

- Evaluating access control methods to secure Amazon API Gateway APIs
- REST API에 대한 상호 TLS 인증 구성
- How to secure API Gateway HTTP endpoints with JWT authorizer
- AWS IoT Core 자격 증명 공급자를 사용하여 AWS 서비스에 직접 호출 권한 부여하기
- AWS 보안 인시던트 대응 가이드

관련 동영상:

- AWS re:invent 2022: Introducing VPC Lattice
- AWS re:invent 2020: Serverless API authentication for HTTP APIs on AWS

관련 예시:

- Amazon VPC Lattice Workshop
- 제로 트러스트 에피소드 1 - The Phantom Service Perimeter 워크숍

사고 대응

예방 및 탐지 제어를 사용하더라도 조직은 잠재적 보안 인시던트에 대응하고 그 영향을 완화하기 위한 메커니즘을 구현해야 합니다. 이러한 준비는 인시던트 발생 시 보안팀이 효과적으로 문제를 격리 및 억제하고 문제에 대한 포렌식을 수행하고, 운영을 알려진 정상 상태로 복구하는 능력에 지대한 영향을 미칩니다. 보안 인시던트보다 앞서 도구 및 액세스를 마련하고 실전 연습을 통해 인시던트 대응을 정기적으로 연습한다면 비즈니스 중단을 최소화하면서 복구할 수 있습니다.

주제

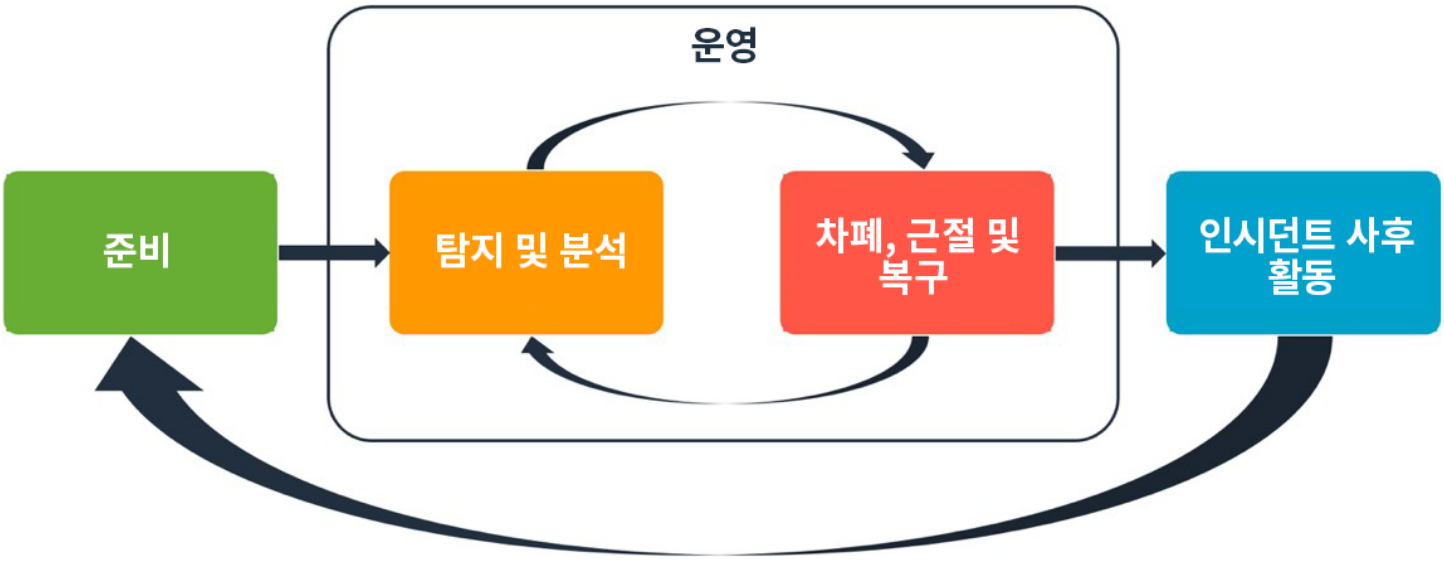
AWS 사고 대응의 측면

조직 내 모든 AWS 사용자는 보안 사고 대응 프로세스에 대한 기본적인 이해가 있어야 하며 보안 직원은 보안 문제에 대응하는 방법을 이해해야 합니다. 교육, 훈련 및 경험은 성공적인 클라우드 사고 대응 프로그램에 필수적이며 발생 가능한 보안 사고를 처리하기 전에 미리 구현하는 것이 이상적입니다. 클라우드에서의 성공적인 사고 대응 프로그램의 기반은 준비, 운영및 인시던트 사후 활동입니다..

이러한 각 측면을 이해하려면 다음 설명을 고려하세요.

- 준비: 탐지 제어를 활성화하고 필요한 도구 및 클라우드 서비스에 대한 적절한 액세스를 확인하여 사고 대응 팀이 내부 AWS 사고를 탐지하고 이에 대응할 수 있도록 준비하세요. 또한 안정적이고 일관된 응답을 보장하는 데 필요한 수동 및 자동 런북을 준비합니다.
- 운영: 탐지, 분석, 억제, 근절 및 복구와 같은 NIST의 사고 대응 단계에 따라 보안 이벤트 및 잠재적 사고를 해결합니다.
- 인시던트 사후 활동입니다.: 보안 이벤트 및 시뮬레이션의 결과를 반복하여 대응의 효율성을 높이고, 대응 및 조사를 통해 도출된 가치를 높이고, 위험을 더욱 줄이세요. 사고를 통해 배우고 개선 활동에 대한 강한 주인의식을 가져야 합니다.

다음 다이어그램은 앞서 언급한 NIST 사고 대응 수명주기와 일치하지만 탐지 및 분석, 억제, 근절 및 복구를 포함하는 작업을 포함하여 이러한 측면의 흐름을 보여줍니다.



AWS 사고 대응의 측면

클라우드 응답의 설계 목표

일반적인 사고 대응 프로세스 및 메커니즘(예: NIST SP 800-61 Computer Security Incident Handling Guide에 정의된 프로세스 및 메커니즘)도 여전히 유효하지만, 클라우드 환경에서 보안 인시던트에 대응하는 것과 관련된 구체적인 설계 목표를 평가해 보는 것이 좋습니다.

- 대응 목표 수립: 이해관계자, 법률 자문, 조직 리더십과 협력하여 인시던트 대응 목표를 결정합니다. 몇 가지 일반적인 목표에는 문제 억제 및 완화, 영향을 받는 리소스 복구, 포렌식을 위한 데이터 보존, 알려진 안전한 운영 환경으로 복귀, 궁극적으로 사고를 통한 학습 등이 포함됩니다.
- 클라우드를 사용하여 대응: 이벤트와 데이터가 발생하는 클라우드 내에서 응답 패턴을 구현합니다.
- 무엇을 가지고 있고 무엇이 필요한지 파악: 로그, 리소스, 스냅샷 및 기타 증거를 응답 전용 중앙 집중식 클라우드 계정에 복사 및 저장하여 보존합니다. 보존 정책을 적용하는 태그, 메타데이터, 메커니즘을 사용합니다. 어떤 서비스를 사용하고 있는지 파악한 다음 해당 서비스를 조사하기 위한 요구 사항을 파악해야 합니다. 환경을 이해하는 데 도움이 되도록 태깅을 사용할 수도 있습니다.
- 재배포 메커니즘 사용: 잘못된 구성으로 인해 보안 이상이 발생한 경우 올바른 구성으로 리소스를 재배포하여 변형을 제거하는 것만큼 간단하게 문제를 해결할 수 있습니다. 손상 가능성이 확인되면 재배포에 성공적이고 검증된 근본 원인 완화 조치가 포함되어 있는지 확인하세요.
- 가능한 경우 자동화: 문제가 발생하거나 사고가 반복되면 프로그래밍 방식으로 분류하고 일반적인 이벤트에 대응하는 메커니즘을 구축하세요. 자동화가 불가능한 고유하거나 복잡하거나 민감한 사고에는 사람의 대응을 활용하세요.

- 확장 가능한 솔루션 선택: 클라우드 컴퓨팅에 대한 조직의 접근 방식의 확장성과 일치하도록 노력하세요. 환경 전반으로 확장되는 탐지 및 대응 메커니즘을 구현하여 탐지와 대응 사이의 시간을 효과적으로 줄이세요.
- 프로세스 교육 및 개선: 프로세스, 도구 또는 인력의 격차를 사전에 파악하고 이를 해결하기 위한 계획을 실행하세요. 시뮬레이션은 격차를 찾고 프로세스를 개선할 수 있는 안전한 방법입니다.

이러한 설계 목표는 사고 대응과 위협 탐지를 모두 수행할 수 있는지 아키텍처 구현을 검토하도록 상기시켜줍니다. 클라우드 구현을 계획할 때는 포렌식적으로 타당한 대응 방법론을 사용하여 사고에 대응하는 방안을 생각해 보세요. 경우에 따라 이러한 대응 작업을 위해 특별히 설정된 조직, 계정 및 도구가 여러 개 있을 수 있습니다. 이러한 도구와 기능은 배포 파이프라인을 통해 사고 대응 담당자가 사용할 수 있도록 해야 합니다. 더 큰 위험을 초래할 수 있으므로 정적이어서는 안 됩니다.

준비

사고 대비는 시기적절하고 효과적인 사고 대응을 위해 매우 중요합니다. 준비는 세 가지 영역에서 이루어집니다.

- 직원: 보안 사고에 대비하려면 사고 대응을 위한 관련 이해 관계자를 식별하고 사고 대응 및 클라우드 기술에 대해 교육해야 합니다.
- 프로세스:: 보안 사고에 대비하여 프로세스를 준비하려면 아키텍처 문서화, 철저한 사고 대응 계획 개발, 보안 이벤트에 대한 일관된 대응을 위한 플레이북 작성이 포함됩니다.
- 기술: 보안 사고에 대비한 기술 준비에는 액세스 설정, 필요한 로그 집계 및 모니터링, 효과적인 경고 메커니즘 구현, 대응 및 조사 기능 개발이 포함됩니다.

이러한 각 영역은 효과적인 사고 대응을 위해 동일하게 중요합니다. 이 세 가지가 모두 없으면 사고 대응 프로그램이 완전하거나 효과적이지 않습니다. 사고에 대비하려면 긴밀한 통합을 통해 직원, 프로세스 및 기술을 준비해야 합니다.

모범 사례

- SEC10-BP01 주요 직원과 외부 리소스 파악
- SEC10-BP02 인시던트 관리 계획 개발
- SEC10-BP03 포렌식 역량 확보
- SEC10-BP04 보안 사고 대응 플레이북 개발 및 테스트
- SEC10-BP05 액세스 권한 사전 프로비저닝

- SEC10-BP06 도구 사전 배포
- SEC10-BP07 시뮬레이션 실행

SEC10-BP01 주요 직원과 외부 리소스 파악

조직이 인시던트에 대응하는 데 도움이 될 수 있는 내/외부 직원, 리소스, 법적 의무를 파악합니다.

다른 팀(예: 법률 자문, 리더십, 비즈니스 이해관계자, AWS Support Services 등)과 함께 클라우드 인시던트 대응 방식을 정할 경우 주요 직원, 이해관계자 및 관련 연락처를 파악해야 합니다. 종속성을 줄이고 응답 시간을 단축하려면 사용하는 서비스에 대해 팀, 전문 보안팀, 응답자를 교육하고 실습 기회를 제공해야 합니다.

외부의 전문 지식 그리고 대응 능력을 강화할 수 있는 다른 관점을 제공할 수 있는 외부 AWS 보안 파트너를 찾는 것이 좋습니다. 신뢰할 수 있는 보안 파트너는 익숙하지 않은 잠재적 위험 또는 위협을 식별하는 데 도움을 줄 수 있습니다.

이 모범 사례가 확립되지 않을 경우 노출되는 위험 수준: 높음

구현 가이드

- 조직의 주요 직원 파악: 인시던트 대응 및 인시던트 이후의 복구에 참여해야 하는 조직 내의 직원 연락처 목록을 유지 관리합니다.
- 외부 파트너 파악: 필요한 경우 인시던트 대응 및 인시던트 이후의 복구를 지원할 수 있는 외부 파트너와 협력합니다.

리소스

관련 문서:

- AWS 인시던트 대응 안내서

관련 동영상:

- AWS 환경에서 보안 인시던트 준비 및 대응

관련 예시:

SEC10-BP02 인시던트 관리 계획 개발

인시던트 대응을 위해 작성해야 할 첫 번째 문서는 인시던트 대응 계획입니다. 인시던트 대응 계획은 인시던트 대응 프로그램 및 전략의 기초가 되도록 설계되었습니다.

이 모범 사례 확립의 이점: 철저하고 명확하게 정의된 인시던트 대응 프로세스를 개발하는 것은 성공적이고 확장 가능한 인시던트 대응 프로그램의 핵심입니다. 보안 이벤트가 발생하면 명확한 단계 및 워크플로가 적시에 대응하는 데 도움이 됩니다. 기존 인시던트 대응 프로세스가 이미 있을 수 있습니다. 현재 상태에 관계없이 인시던트 대응 프로세스를 정기적으로 업데이트, 반복, 테스트하는 것이 중요합니다.

이 모범 사례가 확립되지 않았을 경우의 위험 수준: 높음

구현 가이드

인시던트 관리 계획은 보안 인시던트의 잠재적 영향에 대한 대응, 완화 및 복구에 매우 중요합니다. 인시던트 관리 계획은 보안 인시던트를 적시에 파악하고 해결 및 대응하기 위한 구조화된 프로세스입니다.

클라우드에는 온프레미스 환경에서 볼 수 있는 수많은 동일한 운영 역할과 요구 사항이 있습니다. 인시던트 관리 계획을 수립할 때는 비즈니스 성과와 규정 준수 요구 사항에 가장 잘 맞는 대응 및 복구 전략을 고려하는 것이 중요합니다. 예를 들어, 미국 내 FedRAMP 규정을 준수하는 AWS에서 워크로드를 운영하는 경우 NIST SP 800-61 Computer Security Handling Guide(컴퓨터 보안 처리 안내서). 이와 유사하게 유럽 개인 식별 정보(PII) 데이터가 있는 워크로드를 운영하는 경우 유럽 연합 일반 데이터 보호 규정(GDPR)단축할 수 있습니다.

AWS에서 워크로드에 대한 인시던트 관리 계획을 구축하는 경우, 인시던트 대응에 대한 심층 방어 방식을 구축하기 위해 AWS 공동 책임 모델의 시작합니다. 이 모델에서 AWS는 클라우드 자체의 보안을 관리하지만 클라우드 내에서 보안을 유지하는 것은 고객의 책임입니다. 즉, 고객은 구현을 선택하는 보안 제어에 대한 제어 권한을 보유하며 이에 대한 책임이 있습니다. 클라우드 중심 인시던트 관리 계획 구축에 대한 핵심 개념 및 기본 지침은 AWS 보안 인시던트 대응 안내서 에서 자세히 설명하고 있습니다.

효과적인 인시던트 관리 계획은 클라우드 운영 목표와 함께 끊임없이 반복되고 항상 최신 상태를 유지해야 합니다. 인시던트 관리 계획을 수립 및 발전시킬 때 아래에서 자세히 설명하는 구현 계획의 사용을 고려해 볼 수 있습니다.

구현 단계

역할과 책임 정의

보안 이벤트를 처리하려면 조직 간 규율과 행동 성향이 필요합니다. 조직 구조 내에는 인사(HR) 담당자, 경영진, 법무 담당자와 같이 인시던트 발생 시 책임이 있거나(Responsible) 책임을 지거나(Accountable) 자문을 받거나(Consulted) 최신 정보를 제공받는(Informed) 사람들이 많이 있어야 합니다. 이러한 역할 및 책임과 제3자가 개입해야 하는지 여부를 고려하십시오. 많은 지역에는 해야 할 일과 하지 말아야 할 일을 규정하는 현지 법률이 있습니다. 보안 대응 계획을 위해 Responsible, Accountable, Consulted, Informed(RACI) 차트를 작성하는 것이 불필요해 보일 수 있지만, 그렇게 하면 신속하고 직접적인 커뮤니케이션이 촉진되고 이벤트의 여러 단계에서 리더십의 윤곽을 명확하게 파악할 수 있습니다.

인시던트 발생 시 영향을 받는 애플리케이션 및 리소스의 소유자와 개발자를 포함하는 것이 중요합니다. 이들은 영향을 측정하는 데 도움이 되는 정보와 컨텍스트를 제공할 수 있는 주제 전문가(SME)이기 때문입니다. 개발자 및 애플리케이션 소유자의 인시던트 대응 전문 지식에 의존하기 전에 이들과 함께 연습하고 관계를 구축해야 합니다. 클라우드 관리자 또는 엔지니어와 같은 애플리케이션 소유자 또는 SME는 환경이 익숙하지 않거나 복잡하거나 대응자가 액세스할 수 없는 상황에서 조치를 취해야 할 수 있습니다.

마지막으로 신뢰할 수 있는 파트너는 추가적인 전문 지식과 가치 있는 조사를 제공할 수 있으므로 조사 또는 대응에 참여할 수 있습니다. 팀에 이러한 기술이 없다면 외부 담당자를 고용하여 도움을 받는 것이 좋습니다.

AWS 대응 팀 및 지원 이해

- AWS Support
  - AWS Support는 AWS 솔루션의 성공 및 운영 상태를 지원하는 도구와 전문 지식을 이용할 수 있는 다양한 플랜을 제공합니다. AWS 환경을 계획, 배포, 최적화하는 데 도움이 되는 기술 지원 및 추가 리소스가 필요한 경우 AWS 사용 사례에 가장 적합한 Support 플랜을 선택할 수 있습니다.
  - 그리고 지원 센터 (AWS Management Console에 있으며 로그인이 필요함)를 AWS 리소스에 영향을 미치는 문제에 대한 지원을 받을 수 있는 중앙 연락 창구로 고려하십시오. AWS Support에 대한 액세스는 AWS Identity and Access Management으로 제어됩니다. AWS Support 기능에 액세스하는 방법에 대한 자세한 내용은 AWS Support 시작하기를참조하십시오.
- AWS 고객 인시던트 대응 팀(CIRT)
  - AWS 고객 인시던트 대응 팀(CIRT)은 24/7로 운영되는 전문 글로벌 AWS 팀으로 AWS 공동 책임 모델의고객 측에서 보안 이벤트가 진행되는 동안 고객을 지원합니다.
  - 고객을 지원할 때 AWS CIRT는 AWS에서의 활성 보안 이벤트 분류 및 복구를 지원합니다. 팀은 AWS 서비스 로그를 사용하여 근본 원인 분석을 지원하고 복구를 위한 권장 사항을 제공할 수 있습니다. 또한 향후 보안 이벤트를 방지하는 데 도움이 되는 보안 권장 사항 및 모범 사례를 제공할 수 있습니다.

  - AWS 고객은 AWS CIRT의 지원을 요청할 수 있습니다(AWS Support 사례를 통해).
- DDoS 대응 지원
  - AWS는 AWS Shield를제공합니다. 이 서비스는 AWS에서 실행 중인 웹 애플리케이션을 보호하는 관리형 분산 서비스 거부(DDoS) 보호 서비스를 제공합니다. Shield는 애플리케이션 가동 중지 시간과 지연 시간을 최소화할 수 있는 상시 탐지 및 자동 인라인 방어 기능을 제공하므로 DDoS 보호 혜택을 받기 위해 AWS Support의 지원을 요청할 필요가 없습니다. Shield에는 다음 두 가지 계층이 있습니다. AWS Shield Standard 및 AWS Shield Advanced. 이 두 계층의 차이점에 대해 알아보려면 Shield 기능 설명서를참조하십시오.
- AWS Managed Services(AMS)
  - AWS Managed Services(AMS)에서 AWS 인프라를 지속적으로 관리하므로 사용자는 애플리케이션에 집중할 수 있습니다. AMS는 인프라를 유지 관리하기 위한 모범 사례를 구현함으로써 운영 오버헤드 및 위험을 줄이도록 지원합니다. AMS는 변경 요청, 모니터링, 패치 관리, 보안, 백업 서비스 등과 같은 일반적인 활동을 자동화하고 인프라를 프로비저닝, 운영 및 지원하기 위한 전체 수명 주기 서비스를 제공합니다.
  - AMS는 일련의 보안 탐지 제어를 배포하고 경고에 대한 일차 대응을 연중무휴로 제공합니다. 경고가 시작되면 AMS는 일련의 표준 자동 및 수동 플레이북에 따라 일관된 응답을 확인합니다. 이러한 플레이북은 온보딩 중에 AMS 고객과 공유되므로 고객이 AMS를 통해 대응 방안을 개발하고 조정할 수 있습니다.

인시던트 대응 계획 개발

인시던트 대응 계획은 인시던트 대응 프로그램 및 전략의 기초가 되도록 설계되었습니다. 인시던트 대응 계획은 공식 문서에 포함되어야 합니다. 인시던트 대응 계획에는 일반적으로 다음 섹션이 포함됩니다.

- 인시던트 대응 팀 개요: 인시던트 대응 팀의 목표 및 기능을 간략하게 설명합니다.
- 역할 및 책임: 인시던트 대응 이해 관계자를 나열하고 인시던트 발생 시 해당 이해 관계자의 역할을 자세히 설명합니다.
- 커뮤니케이션 계획: 연락처 정보 및 인시던트 발생 시 커뮤니케이션 방법을 자세히 설명합니다.
- 백업 커뮤니케이션 방법: 인시던트 커뮤니케이션의 백업으로 대역 외 통신을 사용하는 것이 가장 좋습니다. 안전한 대역 외 통신 채널을 제공하는 애플리케이션의 예는 AWS Wickr입니다.
- 인시던트 대응 단계 및 취해야 할 조치: 인시던트 대응의 단계(예: 탐지, 분석, 제거, 억제, 복구)를 열거하며, 여기에는 해당 단계 내에서 취해야 할 상위 수준 조치가 포함됩니다.
- 인시던트 심각도 및 우선순위 정의: 인시던트의 심각도를 분류하는 방법, 인시던트의 우선순위를 지정하는 방법, 심각도 정의가 에스컬레이션 절차에 미치는 영향을 자세히 설명합니다.

이러한 섹션은 규모 및 업종이 다른 회사 간에 공통적으로 사용되지만 각 조직의 인시던트 대응 계획은 고유합니다. 조직에 가장 적합한 인시던트 대응 계획을 수립해야 합니다.

리소스

관련 모범 사례:

- SEC04(보안 관련 이벤트를 어떻게 탐지하나요?)

관련 문서:

- AWS 보안 인시던트 대응 안내서
- NIST: 컴퓨터 보안 인시던트 처리 안내서

SEC10-BP03 포렌식 역량 확보

보안 사고에 앞서 보안 이벤트 조사를 지원하기 위한 포렌식 기능을 개발하는 것이 좋습니다.

이 모범 사례가 확립되지 않을 경우 노출되는 위험 수준: 보통

기존 온프레미스 포렌식의 개념이 AWS에 적용됩니다. AWS 클라우드에서 포렌식 기능 구축을 시작하기 위한 주요 정보는 AWS 클라우드의 포렌식 조사 환경 전략을 참조하십시오.

포렌식을 위한 환경과 AWS 계정 계정 구조를 설정한 후에는 네 단계에 걸쳐 포렌식 방법론을 효과적으로 수행하는 데 필요한 기술을 정의하세요.

- 수집: AWS CloudTrail, AWS Config, VPC 흐름 로그, 호스트 수준 로그 등 관련 AWS 로그를 수집합니다. 가능한 경우 영향을 받은 AWS 리소스의 스냅샷, 백업, 메모리 덤프를 수집합니다.
- 검사: 관련 정보를 추출하고 평가하여 수집한 데이터를 검사합니다.
- 분석: 수집된 데이터를 분석하여 인시던트를 이해하고 결론을 도출합니다.
- 보고: 분석 단계의 결과 정보를 제시합니다.

구현 단계

포렌식 환경 준비

AWS Organizations 는 AWS 리소스의 성장과 확장에 따라 AWS 환경을 중앙에서 관리하고 제어할 수 있도록 도와줍니다. AWS 조직은 AWS 계정을 통합하여 단일 단위로 관리할 수 있도록 합니다. 조직 단위(OU)를 사용하여 계정을 그룹으로 묶어 단일 단위로 관리할 수 있습니다.

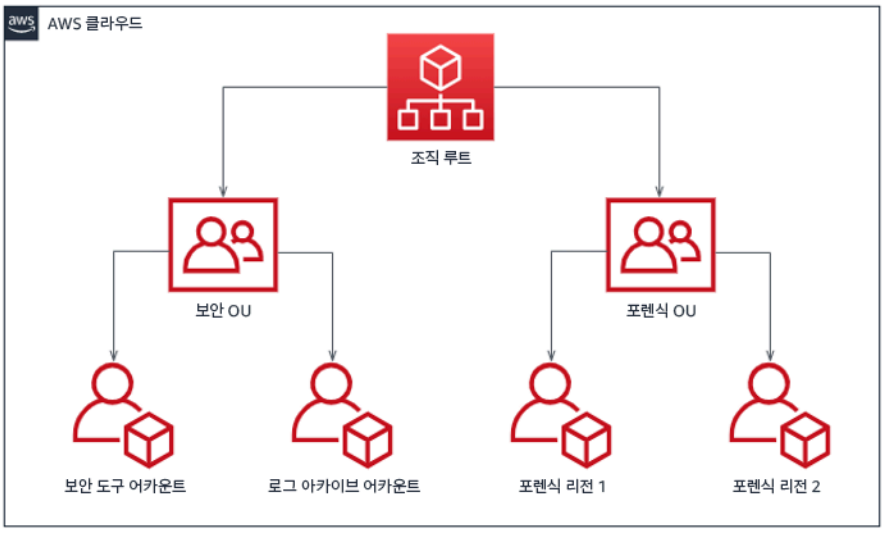
인시던트 대응에는 다음을 포함하는 인시던트 대응 기능을 지원하는 AWS 계정 구조를 갖는 것이 도움이 됩니다. 보안 OU 및 포렌식 OU. 보안 OU 내에는 다음에 대한 계정이 있어야 합니다.

- 로그 아카이브: 제한된 권한으로 로그 아카이브 AWS 계정에서 로그를 집계합니다.
- 보안 도구: 보안 도구 AWS 계정에서 보안 서비스를 중앙 집중화합니다. 이 계정은 보안 서비스에 대한 위임된 관리자 역할을 합니다.

포렌식 OU 내에서, 비즈니스와 운영 모델에 가장 적합한 것에 따라 운영하는 각 리전에 대해 단일 포렌식 계정 또는 여러 개의 계정을 구현할 수 있는 옵션이 있습니다. 리전별로 포렌식 계정을 생성하면 해당 리전 외부에서 AWS 리소스를 생성하는 것을 차단하고 리소스가 의도하지 않은 리전으로 복사되는 위험을 줄일 수 있습니다 예를 들어 US East (N. Virginia) Region(`us-east-1`) 및 US West (Oregon) (`us-west-2`)에서만 운영하는 경우 포렌식 OU에 두 개의 계정이 생깁니다. 하나는 `us-east-1` , 다른 하나는 `us-west-2`입니다.

여러 리전에 대한 포렌식 AWS 계정 계정을 만들 수 있습니다. 데이터 주권 요구 사항을 준수하고 있는지 확인하기 위해 AWS 리소스를 해당 계정으로 복사할 때는 주의해야 합니다. 새 계정을 프로비저닝하는 데는 시간이 걸리므로, 사고 발생 훨씬 전에 포렌식 계정을 생성하고 계측하여 대응 담당자가 대응에 효과적으로 사용할 수 있도록 준비하는 것이 필수적입니다.

다음 다이어그램은 리전별 포렌식 계정이 있는 포렌식 OU를 포함한 샘플 계정 구조를 보여줍니다.



인시던트 대응을 위한 리전별 계정 구조

백업 및 스냅샷 캡처

주요 시스템과 데이터베이스의 백업을 설정하는 것은 보안 인시던트 복구 및 포렌식 용도로 매우 중요합니다. 백업을 설정하면 시스템을 이전의 안전한 상태로 복원할 수 있습니다. AWS에서는 다양한 리

소스의 스냅샷을 생성할 수 있습니다. 스냅샷은 해당 리소스의 특정 시점 백업을 제공합니다. 백업 및 복구를 지원할 수 있는 많은 AWS 서비스가 있습니다. 백업 및 복구를 위한 이러한 서비스 및 접근 방식에 대한 자세한 내용은 백업 및 복구 규범 지침 및 백업을 사용한 보안 인시던트 복구를 참조하십시오.

특히 랜섬웨어와 같은 상황에서는 백업의 보안을 잘 유지하는 것이 중요합니다. 백업 보안에 대한 지침은 AWS에서의 백업 보안을 위한 10가지 주요 보안 모범 사례를 참조하십시오. 백업의 보안을 유지하는 것 외에도 정기적으로 백업 및 복원 프로세스를 테스트하여 보유한 기술과 프로세스가 정상적으로 작동하는지 확인해야 합니다.

포렌식 자동화

보안 이벤트가 발생하는 동안 사고 대응팀은 이벤트와 관련된 기간 동안 정확성을 유지하면서 증거를 신속하게 수집하고 분석할 수 있어야 합니다(예: 특정 이벤트 또는 리소스와 관련된 로그 캡처 또는 Amazon EC2 인스턴스의 메모리 덤프 수집). 특히 많은 인스턴스와 계정에서 관련 증거를 수동으로 수집하는 작업은 인시던트 대응팀에게 어렵고 시간이 많이 소요되는 업무입니다. 또한 수작업으로 수집하는 경우 인적 오류가 발생하기 쉽습니다. 이러한 이유로 포렌식을 위한 자동화를 최대한 개발하고 구현해야 합니다.

AWS는 포렌식을 위한 여러 가지 자동화 리소스를 제공하며 이는 아래 리소스 섹션에 열거되어 있습니다. 이러한 리소스는 AWS가 개발하고 고객이 구현한 포렌식 패턴의 예시입니다. 시작하기에 유용한 참조 아키텍처가 될 수 있지만, 환경, 요구 사항, 도구, 포렌식 프로세스에 따라 이를 수정하거나 새로운 포렌식 자동화 패턴을 생성하는 것을 고려하세요.

리소스

관련 문서:

- AWS 보안 인시던트 대응 가이드 - 포렌식 기능 개발
- AWS 보안 인시던트 대응 가이드 - 포렌식 리소스
- AWS 클라우드의 포렌식 조사 환경 전략
- AWS에서의 포렌식 디스크 수집을 자동화하는 방법
- AWS 권장 가이드 - 인시던트 대응 및 포렌식 자동화

관련 동영상:

- 인시던트 대응 및 포렌식 자동화

관련 예시:

- 자동화된 인시던트 대응 및 포렌식 프레임워크
- Amazon EC2에 대한 자동 포렌식 오케스트레이터

SEC10-BP04 보안 사고 대응 플레이북 개발 및 테스트

인시던트 대응 프로세스를 준비하는 데 있어 가장 중요한 부분은 플레이북을 개발하는 것입니다. 인시던트 대응 플레이북은 보안 이벤트가 발생했을 때 따라야 할 일련의 권장 가이드와 단계를 제공합니다. 명확한 구조와 단계를 갖추면 대응 프로세스가 간소화되고 인적 오류의 가능성이 줄어듭니다.

이 모범 사례가 확립되지 않을 경우 노출되는 위험 수준: 보통

구현 가이드

다음과 같은 인시던트 시나리오에 대한 플레이북을 만들어야 합니다.

- 예상 인시던트: 예상되는 인시던트에 대한 플레이북을 만들어야 합니다. 여기에는 서비스 거부(DoS), 랜섬웨어, 자격 증명 유출과 같은 위협이 포함됩니다.
- 알려진 보안 조사 결과 또는 경고: 알려진 보안 조사 결과 및 경고(예: GuardDuty 조사 결과)에 대한 플레이북을 만들어야 합니다. GuardDuty 조사 결과를 받고 "이제 어떡하지?"라고 생각할 수 있습니다. GuardDuty 조사 결과를 잘못 처리하거나 무시하는 것을 방지하려면 잠재적인 GuardDuty 조사 결과에 대한 플레이북을 만드세요. 일부 해결 세부 사항과 지침은 GuardDuty 설명서에서 확인할 수 있습니다. GuardDuty는 기본적으로 활성화되어 있지 않으며 사용 시 별도의 비용이 발생한다는 점에 유의하세요. GuardDuty에 대한 자세한 내용은 부록 A: 클라우드 기능 정의 - 가시성 및 알림을 참조하십시오.

플레이북에는 보안 분석가가 잠재적인 보안 사고를 적절히 조사하고 대응하기 위해 완료해야 할 기술 단계가 포함되어야 합니다.

구현 단계

플레이북에 포함할 항목은 다음과 같습니다.

- 플레이북 개요: 이 플레이북은 어떤 위험 또는 인시던트 시나리오를 다루고 있나요? 플레이북의 목표는 무엇인가요?
- 사전 요구 사항이 인시던트 시나리오에 어떤 로그, 탐지 메커니즘 및 자동화된 도구가 필요한가요? 예상되는 알림은 무엇인가요?
- 커뮤니케이션 및 에스컬레이션 정보: 참여자는 누구이며 연락처 정보는 어떻게 되나요? 관련된 각 이해 관계자의 책임은 무엇인가요?

- 대응 단계인시던트 대응 단계 전반에서 어떤 전술적 단계를 수행해야 하나요? 분석가는 어떤 쿼리를 실행해야 하나요? 원하는 결과를 얻으려면 어떤 코드를 실행해야 하나요?
  - 감지: 어떻게 인시던트를 감지하나요?
  - 분석: 어떻게 영향 범위를 결정하나요?
  - 포함: 범위를 제한하기 위해 어떻게 인시던트를 격리하나요?
  - 근절: 환경에서 위협을 어떻게 제거하나요?
  - 복구: 영향을 받은 시스템이나 리소스를 어떻게 프로덕션 환경으로 복구하나요?
- 예상 결과: 쿼리와 코드가 실행된 후 플레이북의 예상 결과는 무엇인가요?

리소스

관련 Well-Architected 모범 사례:

- SEC10-BP02 인시던트 관리 계획 개발

관련 문서:

- 인시던트 대응 플레이북을 위한 프레임워크
- 자체 인시던트 대응 플레이북 개발
- 인시던트 대응 플레이북 샘플
- Jupyter 플레이북 및 CloudTrail Lake를 사용하여 AWS 인시던트 대응 런북 빌드

SEC10-BP05 액세스 권한 사전 프로비저닝

인시던트 응답자에게 AWS에 사전 프로비저닝된 올바른 액세스 권한이 있는지 확인하여 조사 및 복구 시간을 단축할 수 있도록 합니다.

일반적인 안티 패턴:

- 인시던트 대응을 위해 루트 계정을 사용합니다.
- 기존 사용자 계정을 변경합니다.
- 적시 권한 승격을 제공할 때 IAM 권한을 직접 조작합니다.

이 모범 사례가 확립되지 않을 경우 노출되는 위험의 수준: 보통

구현 가이드

AWS는 가능한 경우 장기 보안 인증에 대한 의존도를 줄이거나 제거할 것을 권장합니다. 그 대신, 임시 보안 인증 및 적시 권한 승격 메커니즘을 사용하는 것이 좋습니다. 장기 보안 인증은 보안 위험에 노출되기 쉽고, 운영 오버헤드가 증가합니다. 따라서 대부분의 관리 작업에서 인시던트 대응 작업 뿐만 아니라 아이덴티티 페더레이션 과 함께 관리 액세스를 위한 임시 승격을 구현하는 것을 권장합니다. 이 모델에서는 사용자가 더 높은 수준의 권한(인시던트 대응 역할 등)을 요청하고, 사용자가 권한 승격에 적합한 경우 요청이 승인자에게 전송됩니다. 요청이 승인되면 사용자는 일련의 임시 AWS 보안 인증을 받아 자신의 작업을 완료하는 데 사용합니다. 이러한 보안 인증이 만료되면 사용자는 새로운 승격 요청을 제출해야 합니다.

대부분의 인시던트 대응 시나리오에서는 임시 권한 승격을 사용하는 것이 좋습니다. 이를 위한 올바른 방법은 AWS Security Token Service 및 세션 정책 을 사용하여 액세스의 범위를 지정하는 것입니다.

페더레이션형 ID를 사용할 수 없는 시나리오는 다음과 같습니다.

- ID 제공업체(idP)의 침해로 인한 중단
- 잘못된 구성 또는 인적 오류로 인한 페더레이션 액세스 관리 시스템의 손상
- DDoS(분산 서비스 거부) 이벤트 배포 또는 시스템을 사용할 수 없도록 렌더링하는 등의 악의적인 활동

위와 같은 사례의 경우, 긴급 브레이크 글라스 액세스를 구성하여 인시던트에 대한 조사 및 적시 개선이 이루어지도록 해야 합니다. 또한 적절한 권한이 있는 IAM 사용자 를 사용하여 작업을 수행하고 AWS 리소스에 액세스하는 것이 좋습니다. 루트 보안 인증은 루트 사용자 액세스가 필요한 작업에 사용합니다. 인시던트 응답자가 AWS 및 기타 관련 시스템에 대한 올바른 수준의 액세스 권한이 있는지 확인할 수 있도록, 전용 사용자 계정을 사전 프로비저닝하는 것이 좋습니다. 사용자 계정에는 권한이 있는 액세스가 필요하며, 엄격하게 제어 및 모니터링해야 합니다. 계정은 필요한 작업을 수행하기 위한 가장 최소한의 권한만으로 구축해야 하며, 액세스의 수준은 인시던트 관리 계획의 일부로 생성된 플레이북을 기준으로 해야 합니다.

모범 사례는 목적별 전용 사용자 및 역할을 사용하는 것입니다. IAM 정책 추가를 통해 사용자나 역할 액세스 권한을 임시 승격할 경우 인시던트가 발생하는 동안 사용자의 액세스 대상이 불명확해질 뿐만 아니라 승격된 권한이 취소되지 않는 위험이 발생합니다.

최대한 많은 수의 실패 시나리오에서 액세스를 얻을 수 있는지 확인할 수 있도록 가능한 많은 종속성을 제거하는 것이 중요합니다. 이를 지원하기 위해, 인시던트 대응 담당자가 전용 보안 계정에서 AWS Identity and Access Management 사용자로 생성되었는지, 그리고 기존 페더레이션 또는 Single Sign-On(SSO) 솔루션을 통해 관리되고 있지 않은지 확인할 수 있는 플레이북을 생성합니다. 각 개별 대응

담당자는 자신만의 명명된 계정을 가지고 있어야 합니다. 계정 구성은 강력한 암호 정책 및 다중 인증(MFA)을 적용해야 합니다. 인시던트 대응 플레이북에서 AWS Management Console에 대한 액세스 권한만 요구할 경우, 사용자는 구성된 액세스 키를 가지고 있지 않아야 하며 액세스 키 생성이 명시적으로 허용되지 않아야 합니다. 이것은 IAM 정책 또는 서비스 제어 정책(SCP)을 통해서만 구성해야 하며, AWS 보안 모범 사례(AWS Organizations SCP)를 따를 것을 권장합니다. 사용자는 다른 계정에서 인시던트 대응 역할을 수임할 수 있는 기능 외에 다른 권한이 없어야 합니다.

인시던트 과정에서 조사, 개선 조치 또는 복구 활동을 지원하기 위해 기타 내부 또는 외부 인력에게 액세스 권한을 부여해야 할 수 있습니다. 이 경우, 이전에 언급한 플레이북 메커니즘을 사용해야 하며, 인시던트가 완료된 후 모든 추가 액세스 권한이 즉시 취소되었는지 확인할 수 있는 프로세스가 반드시 있어야 합니다.

인시던트 대응 역할의 사용이 적절히 모니터링 및 감사되고 있는지 확인할 수 있도록, 이 목적을 위해 생성된 IAM 사용자 계정이 인력 간에 공유되지 않도록 하고, AWS 계정 루트 사용자가 특정 작업에 필요하지 않다면사용되지 않습니다. 루트 사용자가 필요한 경우(예를 들어, 특정 계정에 대한 IAM 액세스를 사용할 수 없는 경우), 사용 가능한 플레이북을 통해 별도의 프로세스를 사용하여 루트 사용자 암호 및 MFA 토큰의 가용성을 확인해야 합니다.

인시던트 대응 역할을 위한 IAM 정책을 구성하기 위해 IAM Access Analyzer 를 사용하여 AWS CloudTrail 로그를 기반으로 정책을 생성하는 것을 고려해 볼 수 있습니다. 이를 위해서는 비 프로덕션 계정에서 인시던트 대응 역할에 대한 관리자 액세스 권한을 부여하고 플레이북에 따라 실행해야 합니다. 완료되면 수행된 작업만 허용하는 정책을 생성할 수 있습니다. 그 후 이 정책은 모든 계정의 모든 인시던트 대응 역할에 적용할 수 있습니다. 더욱 쉬운 관리 및 감사를 허용할 수 있도록 각 플레이북에 대한 별도의 IAM 정책을 생성할 수 있습니다. 플레이북에 포함할 수 있는 예로는 랜섬웨어, 데이터 침해, 프로덕션 액세스의 손실 및 기타 시나리오에 대한 대응 계획이 있을 수 있습니다.

인시던트 대응 사용자 계정을 사용하여 전용 인시던트 대응 IAM 역할(다른 AWS 계정 계정 내)을 수임합니다. 이러한 역할은 보안 계정의 사용자만 수임할 수 있도록 구성되어야 하며, 신뢰 관계에서는 호출하는 보안 주체가 MFA를 사용하여 인증해야 합니다. 역할은 범위가 좁은 IAM 정책을 사용하여 액세스를 제어해야 합니다. 이러한 역할에 대한 모든 `AssumeRole` 요청은 CloudTrail에 로깅되고 알림이 생성되며, 이러한 역할을 사용하여 수행된 모든 작업이 로깅되도록 합니다.

IAM 사용자 계정과 IAM 역할의 이름을 모두 명확하게 지정하여 CloudTrail 로그에서 쉽게 찾을 수 있도록 하는 것이 좋습니다. 그 예로는 IAM 계정 *<USER_ID>*-BREAK-GLASS 및 IAM 역할 BREAK-GLASS-ROLE이 있습니다.

CloudTrail 은 AWS 계정의 API 활동을 기록하는 데 사용되며 인시던트 대응 역할의 사용에 대한 알림을 구성하는 데 사용해야 합니다. 루트 키 사용 시 알림 구성에 대한 블로그 게시물을 참조하세요. 지침

을 수정하여 Amazon CloudWatch 지표 필터를 필터로 구성할 수 있으며 이 경우 AssumeRole 이벤트에 구성되며 인시던트 대응 IAM 역할과 관련됩니다.

```
{ $.eventName = "AssumeRole" && $.requestParameters.roleArn =
 "<INCIDENT_RESPONSE_ROLE_ARN>" && $.userIdentity.invokedBy NOT EXISTS && $.eventType !
= "AwsServiceEvent" }
```

인시던트 대응 역할은 높은 수준의 액세스 권한을 가질 가능성이 높기 때문에, 이러한 알림은 광범위한 그룹에 전달되고 신속하게 조치되는 것이 중요합니다.

인시던트 발생 시 응답자는 IAM에 의해 직접 보호되지 않는 시스템에 대한 액세스가 필요할 수 있습니다. 여기에는 Amazon Elastic Compute Cloud 인스턴스, Amazon Relational Database Service 데이터베이스 또는 서비스형 소프트웨어(SaaS) 플랫폼이 포함됩니다. SSH 또는 RDP와 같은 기본 프로토콜을 사용하는 것보다는 AWS Systems Manager Session Manager 를 Amazon EC2 인스턴스에 대한 모든 관리 액세스에 사용하는 것이 좋습니다. 이 액세스는 보안 및 감사 기능이 있는 IAM을 사용하여 제어할 수 있습니다. 또한 AWS Systems Manager Run Command 문서를 사용하여 플레이북의 일부를 자동화할 수 있으며, 이를 통해 사용자 오류를 줄이고 복구 시간을 개선할 수 있습니다. 데이터베이스 및 서드 파티 도구에 대한 액세스를 위해, AWS Secrets Manager에 액세스 보안 인증을 저장하고 인시던트 응답자 역할에 액세스 권한을 부여하는 것이 좋습니다.

마지막으로, 인시던트 대응 IAM 사용자 계정의 관리를 입사, 전근 및 퇴사 프로세스 에 추가하고 정기적으로 검토 및 테스트하여 대상 액세스만 허용되는지 확인해야 합니다.

리소스

관련 문서:

- AWS 환경에 대한 임시 승격된 액세스 관리
- AWS 보안 인시던트 대응 안내서
- AWS Elastic Disaster Recovery
- AWS Systems Manager Incident Manager
- IAM 사용자의 계정 암호 정책 설정
- AWS에서 다중 인증(MFA) 사용
- MFA를 통한 크로스 계정 액세스 구성
- IAM Access Analyzer를 사용하여 IAM 정책 생성
- 다중 계정 환경에서의 AWS Organizations 서비스 제어 정책 모범 사례
- AWS 계정의 루트 액세스 키가 사용될 때 알림을 받는 방법

- IAM 관리형 정책을 사용하여 세분화된 세션 권한 생성

관련 동영상:

- AWS의 인시던트 대응 및 포렌식 자동화
- 런북, 인시던트 보고서, 인시던트 대응에 대한 DIY 가이드
- AWS 환경에서 보안 인시던트 준비 및 대응

관련 예시:

- 실습: AWS 계정 설정 및 루트 사용자
- 실습: AWS 콘솔 및 CLI를 사용한 인시던트 대응

SEC10-BP06 도구 사전 배포

보안 담당자가 조사부터 복구까지 소요되는 시간을 단축할 수 있는 올바른 도구를 미리 배포했는지 확인합니다.

이 모범 사례가 확립되지 않을 경우 노출되는 위험 수준: 보통

구현 가이드

보안 대응 및 운영 기능을 자동화하기 위해 AWS의 포괄적인 API 및 도구 세트를 사용할 수 있습니다. 자격 증명 관리, 네트워크 보안, 데이터 보호, 모니터링 기능을 완전히 자동화하고 이미 사용하고 있는 대중적인 소프트웨어 개발 방법을 사용하여 제공할 수 있습니다. 보안 자동화를 구축하면 직원이 보안 상태를 모니터링하면서 수동으로 이벤트에 대응하는 것이 아니라 시스템이 모니터링 및 검토하고 대응을 시작할 수 있습니다.

인시던트 대응팀은 같은 방식으로 계속 알림에 대응할 경우 알림에 대한 피로감을 느낄 위험이 있습니다. 시간이 지남에 따라 팀이 알림에 무감각한 상태가 되어 일상적인 상황을 처리하는 데 실수하거나 비정상적인 알림을 놓칠 수 있습니다. 자동화는 반복적이고 일상적인 알림을 처리하는 기능을 사용함으로써 알림에 대한 피로감을 방지하며, 중요하고 특별한 인시던트만 사람이 직접 처리하도록 합니다. Amazon GuardDuty, AWS CloudTrail Insights, Amazon CloudWatch Anomaly Detection과 같은 이상 탐지 시스템을 통합하면 일반적인 임계값 기반 알림의 부담을 줄일 수 있습니다.

프로세스의 단계를 프로그래밍 방식으로 자동화하여 수동 프로세스를 개선할 수 있습니다. 이벤트에 대한 수정 패턴을 정의한 후 해당 패턴을 실행 가능한 로직으로 분해하고 코드를 작성하여 해당 로직을

수행할 수 있습니다. 그런 다음, 응답자가 해당 코드를 실행하여 문제를 해결할 수 있습니다. 시간이 지남에 따라 점점 더 많은 단계를 자동화할 수 있으며, 궁극적으로 일반적인 인시던트의 전체 클래스를 자동으로 처리할 수 있습니다.

보안 조사 중에 관련 로그를 검토하여 인시던트의 전체 범위와 타임라인을 기록하고 이해할 수 있어야 합니다. 관심 있는 특정 작업이 발생했음을 나타내는 알림 생성에도 로그가 필요합니다. 쿼리 및 검색 메커니즘을 선택, 활성화, 저장 및 설정하고 경보를 설정하는 것이 중요합니다. 또한 로그 데이터를 검색할 수 있는 도구를 제공하는 효과적인 방법은 Amazon Detective입니다.

AWS는 200개 이상의 클라우드 서비스와 수천 개의 기능을 제공합니다. 인시던트 대응 전략을 지원하고 간소화할 수 있는 서비스를 검토하는 것이 좋습니다.

로깅 외에도 태그 지정 전략을 개발하고 구현해야 합니다. 태그 지정은 AWS 리소스의 목적에 대한 컨텍스트를 제공하는 데 도움이 될 수 있습니다. 태그 지정은 자동화를 위해서도 사용할 수 있습니다.

구현 단계

분석 및 알림을 위한 로그 선택 및 설정

인시던트 대응을 위한 로깅 구성에 대한 다음 설명서를 참조하세요

- 보안 인시던트 대응을 위한 로깅 전략
- SEC04-BP01 서비스 및 애플리케이션 로깅 구성

보안 서비스를 사용하여 탐지 및 대응 지원

AWS는 기본 탐지, 예방 및 대응 기능을 제공하며, 기타 서비스를 사용하여 맞춤형 보안 솔루션을 설계할 수 있습니다. 보안 인시던트 대응과 가장 관련성이 높은 서비스 목록은 클라우드 기능 정의를 참조하십시오.

태그 지정 전략 개발 및 구현

AWS 리소스를 사용하는 비즈니스 사용 사례와 관련 내부 이해 관계자에 대한 컨텍스트 정보를 얻는 것은 어려울 수 있습니다. 한 가지 방법은 AWS 리소스에 메타데이터를 할당하고 사용자 정의 키와 값으로 구성되는 태그의 형태를 사용하는 것입니다. 태그를 생성하여 리소스를 목적, 소유자, 환경, 처리되는 데이터 유형 및 기타 원하는 기준에 따라 분류할 수 있습니다.

일관된 태그 전략을 사용하면 AWS 리소스에 대한 컨텍스트 정보를 신속하게 식별할 수 있으므로 응답 시간을 단축하고 조직 컨텍스트에 소요되는 시간을 최소화할 수 있습니다. 태그는 응답 자동화를 시작하는 메커니즘으로도 사용할 수 있습니다. 태그를 지정할 항목에 대한 자세한 내용은 AWS 리소스에

태그 지정을 참조하십시오. 먼저 조직 전체에 구현할 태그를 정의하는 것이 좋습니다. 그런 다음 이 태그 지정 전략을 구현하고 적용합니다. 구현 및 적용에 대한 자세한 내용은 AWS 태그 정책 및 서비스 제어 정책(SCP)을 사용한 AWS 리소스 태그 지정 전략 구현을 참조하십시오.

리소스

관련 Well-Architected 모범 사례:

- SEC04-BP01 서비스 및 애플리케이션 로깅 구성
- SEC04-BP02 로그, 결과 및 지표를 중앙에서 분석

관련 문서:

- 보안 인시던트 대응을 위한 로깅 전략
- 인시던트 대응 클라우드 기능 정의

관련 예시:

- Amazon GuardDuty 및 Amazon Detective를 사용한 위협 탐지 및 대응
- 보안 허브 워크숍
- Amazon Inspector를 사용한 취약성 관리

SEC10-BP07 시뮬레이션 실행

시간이 지나면서 조직이 성장하고 발전함에 따라 위협 환경도 변화하므로 인시던트 대응 능력을 지속적으로 검토하는 것이 중요합니다. 시뮬레이션(게임 데이라고도 함)을 실행하는 것도 이 평가를 수행하는 데 사용할 수 있는 방법 중 하나입니다. 시뮬레이션은 위협 행위자의 전술, 기술 및 절차(TTP)를 모방하도록 설계된 실제 보안 이벤트 시나리오를 사용하며, 이를 통해 조직은 이러한 모의 사이버 이벤트에 실제 상황과 같이 대응하여 인시던트 대응 능력을 발휘하고 평가할 수 있습니다.

이 모범 사례 수수립의 이점: 시뮬레이션에는 다양한 이점이 있습니다.

- 사이버 대비 상태를 검증하고 인시던트 대응자의 자신감을 높입니다.
- 도구 및 워크플로의 정확성과 효율성을 테스트합니다.
- 인시던트 대응 계획에 맞춰 커뮤니케이션 및 에스컬레이션 방법을 개선합니다.
- 덜 일반적인 벡터에 대응할 수 있는 기회를 제공합니다.

이 모범 사례를 따르지 않을 경우 노출 위험도: 중간

구현 가이드

시뮬레이션에는 다음과 같은 세 가지 주요 유형이 있습니다.

- 탁상 연습: 시뮬레이션에 대한 탁상 접근 방식은 다양한 인시던트 대응 이해 관계자가 참여하여 책임진 역할을 연습하고 확립된 커뮤니케이션 도구과 플레이북을 사용하는 토론 기반 세션입니다. 연습은 일반적으로 가상 장소, 실제 장소 또는 이들 장소의 조합에서 하루 종일 수행할 수 있어 언제든 촉진시킬 수 있습니다. 토론을 기반으로 하기 때문에 탁상 연습은 프로세스, 사람, 협업에 중점을 둡니다. 기술은 토론의 핵심 부분이지만 인시던트 대응 도구 또는 스크립트의 실제 사용은 일반적으로 탁상 연습의 일부가 아닙니다.
- 퍼플 팀 연습: 퍼플 팀 연습은 인시던트 대응 담당자(블루 팀)와 시뮬레이션된 위협 행위자(레드 팀) 간의 협업 수준을 높입니다. 블루 팀은 보안 운영 센터(SOC)의 직원으로 구성되지만 실제 사이버 이벤트 중에 관여하게 될 다른 이해 관계자들도 포함될 수 있습니다. 레드 팀은 보안 공격 교육을 받은 침투 테스트 팀 또는 주요 이해 관계자로 구성됩니다. 레드 팀은 시나리오를 설계할 때 연습 진행자와 협력하여 시나리오가 정확하고 실현 가능한지 확인합니다. 퍼플 팀 연습에서는 인시던트 대응 작업을 지원하는 탐지 메커니즘, 도구 및 표준 운영 절차(SOP)에 주로 초점을 맞춥니다.
- 레드 팀 연습: 레드 팀 연습 중에 공격 팀(레드 팀)은 미리 정해진 범위에서 특정 목표 또는 일련의 목표를 달성하기 위해 시뮬레이션을 수행합니다. 방어 팀(블루 팀)은 훈련의 범위와 기간을 꼭 알 필요가 없습니다. 이를 모르면 실제 인시던트에 어떻게 대응하는지에 대한 더 현실적인 평가를 받을 수 있습니다. 레드 팀 연습은 침습적 테스트일 수 있으므로 주의가 필요하고 해당 연습이 환경에 실제로 해를 끼치지 않는지 확인하기 위한 관리 조치를 취해야 합니다.

정기적으로 사이버 시뮬레이션을 진행하는 것이 좋습니다. 각 연습 유형에는 참가자와 조직 전체에 대한 고유한 이점이 있으므로 덜 복잡한 시뮬레이션 유형(예: 탁상 연습)에서 시작하여 더 복잡한 시뮬레이션 유형(레드 팀 연습)으로 진행할 수 있습니다. 보안 성숙도, 리소스, 원하는 결과에 따라 시뮬레이션 유형을 선택해야 합니다. 일부 고객은 복잡성과 비용 때문에 레드 팀 연습을 선택하지 않을 수 있습니다.

구현 단계

선택한 유형에 관계없이 시뮬레이션은 일반적으로 다음 구현 단계를 따릅니다.

1. 핵심 연습 요소 정의: 시뮬레이션의 시나리오와 목표를 정의합니다. 이 두 가지 모두 리더의 승인을 받아야 합니다.
2. 주요 이해 관계자 식별: 연습에는 최소한 연습 진행자와 참가자가 필요합니다. 시나리오에 따라 법무, 커뮤니케이션 또는 경영진과 같은 추가 이해 관계자가 참여할 수 있습니다.

3. 시나리오 구축 및 테스트: 특정 요소가 실현 가능하지 않은 경우 구축 중인 시나리오를 재정의해야 할 수 있습니다. 이 단계의 결과로 최종 시나리오가 도출될 것으로 예상됩니다.
4. 시뮬레이션 진행: 시뮬레이션 유형에 따라 어떤 방법으로 진행시킬지 결정됩니다(종이를 사용한 시나리오 또는 고도로 기술적인 시뮬레이션 시나리오). 진행자는 연습 목표에 맞게 촉진 전략을 조정해야 하며 가능한 한 모든 연습 참가자를 참여시켜 최대한의 이점을 확보해야 합니다.
5. 사후 조치 보고서(AAR) 작성: 잘 운영된 영역, 개선이 필요한 영역, 잠재적인 격차를 파악합니다. AAR은 시뮬레이션의 효과와 시뮬레이션된 이벤트에 대한 팀의 반응을 측정하여 향후 시뮬레이션을 통해 시간의 흐름에 따른 진행 상황을 추적할 수 있도록 해야 합니다.

리소스

관련 문서:

- AWS 인시던트 대응 안내서

관련 동영상:

- AWS GameDay - Security Edition(보안 에디션)

운영

운영은 사고 대응 수행의 핵심입니다. 여기서 보안 사고 대응 및 해결 조치가 이루어집니다. 운영에는 다음 다섯 단계가 포함됩니다. 탐지, 분석, 격리, 근절 및 복구. 이러한 단계 및 목표에 대한 설명은 다음 표에 나와 있습니다.

단계	목표
탐지	잠재적 보안 이벤트 파악
분석	보안 이벤트가 사고인지 판단하고 사고 범위를 평가하세요.
컨테인먼트	보안 이벤트의 범위를 최소화하고 제한합니다.
근절	보안 이벤트와 관련된 승인되지 않은 리소스 또는 아티팩트를 제거합니다. 보안 사고를 일으킨 완화 조치를 구현하세요.

단계	목표
복구:	시스템을 알려진 안전 상태로 복원하고 이러한 시스템을 모니터링하여 위협이 다시 발생하지 않는지 확인합니다.

이 단계는 효과적이고 강력한 방식으로 대응하기 위해 보안 사고에 대응하고 이를 운영할 때 지침으로 활용해야 합니다. 실제로 취하는 조치는 사고에 따라 달라집니다. 예를 들어 랜섬웨어와 관련된 인시던트는 퍼블릭 Amazon S3 버킷과 관련된 인시던트와는 다른 대응 단계를 따라야 합니다. 또한 이러한 단계가 반드시 순차적으로 발생하는 것은 아닙니다. 격리 및 근절 후에는 분석 작업으로 돌아가 자신의 행동이 효과적이었는지 파악해야 할 수도 있습니다.

직원, 프로세스 및 기술 전반의 철저한 준비가 효과적인 운영을 위한 핵심입니다. 활성 보안 이벤트에 효과적으로 대응할 수 있도록 준비 섹션의 모범 사례를 따르세요.

자세한 내용을 보려면 운영 AWS 보안 인시던트 대응 안내서를 보세요.

인시던트 사후 활동입니다.

위협 환경은 끊임없이 변화하므로 환경을 효과적으로 보호할 수 있는 조직의 역량도 그에 못지않게 역동적으로 대처하는 것이 중요합니다. 지속적인 개선의 핵심은 발생 가능한 보안 사고를 효과적으로 탐지, 대응 및 조사할 수 있는 능력을 향상시키고, 발생 가능한 취약성을 줄이고, 대응 시간을 단축하고, 안전한 운영으로 돌아갈 수 있도록 인시던트 및 시뮬레이션의 결과를 반복해서 검토하는 것입니다. 다음 메커니즘은 조직이 상황에 관계없이 효과적으로 대응할 수 있는 최신 역량과 지식을 갖추고 있는지 확인하는 데 도움이 될 수 있습니다.

모범 사례

- SEC10-BP08 인시던트로부터 학습하기 위한 프레임워크 구축

SEC10-BP08 인시던트로부터 학습하기 위한 프레임워크 구축

이때 학습한 내용 프레임워크와 근본 원인 분석 기능을 구현하면 인시던트 대응 능력을 개선하는 데 도움이 될 뿐만 아니라 인시던트 재발을 방지하는 데도 도움이 됩니다. 각 인시던트에서 교훈을 얻음으로써 동일한 실수, 노출 또는 잘못된 구성을 반복하지 않도록 하여 보안 태세를 개선할 뿐만 아니라 사전에 방지 가능한 상황으로 인한 시간 손실을 최소화할 수 있습니다.

이 모범 사례가 확립되지 않을 경우 노출되는 위험 수준: 보통

구현 가이드

개략적인 수준에서 다음 사항을 설정하고 달성하는 학습한 내용 프레임워크를 구현하는 것이 중요합니다.

- 학습한 교훈은 언제 적용하게 되나요?
- 학습한 교훈 과정에는 무엇이 포함되나요?
- 학습한 교훈은 어떻게 수행되나요?
- 누가, 어떻게 이 과정에 참여하나요?
- 개선이 필요한 부분은 어떻게 확인할 수 있나요?
- 개선 사항을 효과적으로 추적하고 구현할 수 있도록 어떻게 해야 할까요?

프레임워크는 개인에게 초점을 맞추거나 개인을 비난하는 것이 아니라 도구와 프로세스를 개선하는 데 초점을 맞춰야 합니다.

구현 단계

앞에서 설명한 개략적인 결과 외에도, 프로세스에서 최대한의 가치(실행 가능한 개선으로 이어지는 정보)를 이끌어낼 수 있도록 올바른 질문을 하는 것이 중요합니다. 다음 질문을 고려하면 학습한 교훈 토론을 시작하는 데 도움이 됩니다.

- 어떤 인시던트였나요?
- 인시던트가 언제 처음 확인되었나요?
- 어떻게 식별되었나요?
- 어떤 시스템에서 해당 활동에 대해 경고했나요?
- 어떤 시스템, 서비스 및 데이터가 관련되어 있나요?
- 구체적으로 어떤 일이 발생했나요?
- 어떤 점이 잘 작동했나요?
- 어떤 점이 잘 작동하지 않았나요?
- 인시던트에 대응하기 위해 어떤 프로세스 또는 절차가 실패했거나 확장하지 못했나요?
- 다음 영역에서 개선할 수 있는 사항:
  - 직원
    - 연락이 필요한 직원이 실제로 연락이 가능했고 연락처 목록이 최신 상태였나요?
    - 인시던트에 효과적으로 대응하고 조사하는 데 필요한 교육이나 역량을 갖춘 직원이 없었나요?

    - 적절한 리소스가 준비되어 있고 이용 가능했나요?
- 프로세스:
    - 프로세스와 절차를 준수했나요?
    - 이 (유형의) 인시던트에 대한 프로세스와 절차가 문서화되어 있고 사용 가능했나요?
    - 필요한 프로세스 및 절차가 누락되지는 않았나요?
    - 대응 담당자가 문제를 대응하는 데 필요한 정보에 적시에 액세스할 수 있었나요?
- 기술
    - 기존 경고 시스템이 활동을 효과적으로 식별하고 경고했나요?
    - 어떻게 하면 탐지 시간을 50%까지 줄일 수 있을까요?
    - 기존 경고 시스템을 개선해야 하나요, 아니면 이 (인시던트 유형) 사고에 대해 새로운 경고 시스템을 구축해야 하나요?
    - 기존 도구로 인시던트를 효과적으로 조사(검색/분석)할 수 있었나요?
    - 이 (유형의) 인시던트를 더 빨리 식별하려면 어떻게 해야 할까요?
    - 이 (유형의) 인시던트가 재발하는 것을 방지하려면 어떻게 해야 할까요?
    - 개선 계획의 담당자는 누구이며 개선 계획이 실행되었는지 어떻게 테스트할 예정인가요?
    - 추가 모니터링 또는 예방적 통제 및 프로세스를 구현하고 테스트할 일정은 어떻게 되나요?

이 목록은 모든 것을 포함하지는 않지만, 조직 및 비즈니스 요구 사항이 무엇인지 식별하고 인시던트로부터 가장 효과적으로 학습하고 보안 태세를 지속적으로 개선하기 위해 이를 분석할 수 있는 방법을 식별하기 위한 출발점이 될 수 있습니다. 가장 중요한 것은 인시던트 대응 프로세스, 문서화 및 이해 관계자 전반의 기대치에서 학습한 교훈을 표준으로 삼아 통합하는 것부터 시작하는 것입니다.

리소스

관련 문서:

- AWS 보안 인시던트 대응 가이드 - 인시던트를 통해 배울 수 있는 프레임워크 수립
- NCSC CAF 지침 - 학습한 교훈

애플리케이션 보안

애플리케이션 보안(AppSec)은 개발하는 워크로드의 보안 속성을 설계, 구축 및 테스트하는 전반적인 프로세스를 설명합니다. 조직에 적절한 교육을 받은 직원이 있어야 하며, 빌드 및 릴리스 인프라의 보안 속성을 이해하고, 자동화를 사용하여 보안 문제를 식별해야 합니다.

소프트웨어 개발 수명 주기(SDLC) 및 릴리스 후 프로세스에 정기적으로 애플리케이션 보안 테스트를 수행하면 프로덕션 환경에 유입되는 애플리케이션 보안 문제를 식별, 수정 및 방지할 수 있는 구조화된 메커니즘을 갖출 수 있습니다.

애플리케이션 개발 방법에는 워크로드를 설계, 구축, 배포 및 운영할 때의 보안 제어 기능이 포함되어야 합니다. 이와 동시에 지속적으로 결함을 줄이고 기술 부채를 최소화하도록 프로세스를 조정하세요. 예를 들어 설계 단계에서 위협 모델링을 사용하면 설계 결함을 조기에 발견할 수 있으므로 기다렸다가 나중에 문제를 완화하는 것보다 수정이 더 쉽고 비용이 적게 듭니다.

SDLC에서 결함은 보통 일찍 해결해야 비용과 복잡성이 줄어듭니다. 문제를 해결하는 가장 쉬운 방법은 애초에 문제가 발생하지 않게 하는 것입니다. 따라서 위협 모델로 시작하면 설계 단계부터 올바른 결과에 집중하는 데 도움이 됩니다. AppSec 프로그램이 발전을 거듭하면서 자동화를 사용하여 수행되는 테스트의 양을 늘리고, 빌더에 대한 피드백의 충실도를 개선하며, 보안 검토에 필요한 시간을 줄일 수 있습니다. 이러한 모든 작업은 구축하는 소프트웨어의 품질을 개선하고, 프로덕션에 기능을 도입하는 속도를 높입니다.

이러한 구현 지침은 조직 및 문화, 파이프라인의 보안, 파이프라인 내 보안, 종속성 관리라는 4가지 영역에 중점을 둡니다. 각 영역은 구현할 수 있는 일련의 원칙을 제공하며, 워크로드를 설계, 개발, 구축, 배포 및 운영하는 방법을 아우르는 전체적인 관점을 제공합니다.

AWS에는 애플리케이션 보안 프로그램을 다룰 때 사용할 수 있는 여러 방법이 있습니다. 이러한 접근 방식 중 일부는 기술에 의존하며, 일부는 애플리케이션 보안 프로그램의 인력 및 조직 측면에 중점을 두고 있습니다.

모범 사례

- SEC11-BP01 애플리케이션 보안 교육
- SEC11-BP02 개발 및 릴리스 수명 주기를 통한 테스트 자동화
- SEC11-BP03 정기적인 침투 테스트 시행
- SEC11-BP04 수동 코드 검토
- SEC11-BP05 패키지 및 종속성 서비스의 중앙 집중화
- SEC11-BP06 프로그래밍 방식으로 소프트웨어 배포

- SEC11-BP07 정기적으로 파이프라인의 보안 속성 평가
- SEC11-BP08 보안 소유권이 워크로드 팀에 귀속되도록 프로그램 구축

SEC11-BP01 애플리케이션 보안 교육

애플리케이션의 안전한 개발 및 운영을 위해 조직 내 빌더에게 일반적인 사례를 교육합니다. 보안 중점 개발 관행을 도입하면 보안 검토 단계에서만 탐지되는 문제의 발생 가능성을 줄이는 데 도움이 됩니다.

원하는 결과: 소프트웨어는 보안을 염두에 두어 설계되고 구축되어야 합니다. 조직의 빌더가 위협 모델로 시작하는 보안 개발 관행에 대해 교육을 받으면 제작되는 소프트웨어의 전반적인 품질과 보안이 향상됩니다. 이 접근 방식은 보안 검토 단계 이후에 재작업의 필요성을 덜어주어 소프트웨어나 기능 납품 시간을 줄일 수 있습니다.

이 모범 사례에서 보안 개발은 작성 중인 소프트웨어와 소프트웨어 개발 수명 주기(SDLC)를 지원하는 도구 또는 시스템을 의미합니다.

일반적인 안티 패턴:

- 보안 검토가 끝날 때까지 기다린 다음 시스템의 보안 속성을 고려합니다.
- 보안 팀에 모든 보안 결정을 맡깁니다.
- SDLC에서 내린 결정이 조직의 전반적인 보안 기대치 또는 정책과 어떤 관련이 있는지를 전달하지 못합니다.
- 보안 검토 프로세스에 너무 늦게 참여합니다.

이 모범 사례 확립의 이점:

- 개발 주기 초기에 조직의 보안 요구 사항을 보다 효과적으로 이해합니다.
- 잠재적인 보안 문제를 보다 신속하게 식별하고 해결하여 기능을 발 빠르게 제공할 수 있습니다.
- 소프트웨어 및 시스템의 품질이 향상됩니다.

이 모범 사례를 따르지 않을 경우 노출 위험도: 중간

구현 가이드

조직의 빌더를 교육합니다. 위협 모델링 과정부터 시작하면 보안 교육의 굳건한 토대를 마련할 수 있습니다. 이상적으로는 빌더가 워크로드와 관련된 정보에 자체적으로 액세스할 수 있어야 합니다. 이러한

액세스를 통해 구축하는 시스템의 보안 속성에 대해 다른 팀에 문의할 필요 없이 정보에 기반한 의사 결정을 내릴 수 있습니다. 검토에 보안 팀을 참여시키는 프로세스가 명확하게 정의되어야 하며, 이를 간단하게 따를 수 있어야 합니다. 검토 프로세스의 단계는 보안 교육에 포함되어야 합니다. 알려진 구현 패턴 또는 템플릿을 사용할 수 있는 경우 찾기가 쉬워야 하며 전반적인 보안 요구 사항과 연결되어야 합니다. 직접 맞춤 구성할 필요성을 줄이려면 AWS CloudFormation, AWS Cloud Development Kit (AWS CDK) Constructs, Service Catalog 또는 기타 템플릿 도구 사용을 고려해 보세요.

구현 단계

- 빌더를 대상으로 위협 모델링 과정을 교육하면 좋은 토대를 마련하고 보안을 대하는 방식을 교육할 수 있습니다.
- AWS 교육 and Certification, 산업 또는 AWS 파트너 교육에 대한 액세스 권한을 제공합니다.
- 보안 팀, 워크로드 팀 및 기타 이해 관계자 간의 책임 분담을 명확히 하는 조직의 보안 검토 프로세스를 교육합니다.
- 가능한 경우 코드 예시 및 템플릿을 포함하여 보안 요구 사항을 충족하는 방법을 다루는 자습형 지침을 게시합니다.
- 보안 검토 프로세스 및 교육을 받은 빌더 팀의 경험에 대해 정기적으로 피드백을 얻고, 피드백을 바탕으로 개선합니다.
- 게임 데이 또는 버그 배쉬 캠페인을 사용하여 문제 수를 줄이고 빌더의 기술을 강화합니다.

리소스

관련 모범 사례:

- SEC11-BP08 보안 소유권이 워크로드 팀에 귀속되도록 프로그램 구축

관련 문서:

- AWS 교육 and Certification
- How to think about cloud security governance(클라우드 보안 거버넌스를 대하는 방식)
- How to approach threat modeling(위협 모델링 접근 방식)
- Accelerating training – The AWS Skills Guild(교육 가속화 - AWS Skills Guild)

관련 동영상:

- Proactive security: Considerations and approaches(사전 예방적 보안: 고려 사항 및 접근 방법)

관련 예시:

- Workshop on threat modeling(위협 모델링 워크숍)
- Industry awareness for developers(개발자를 위한 업계 인지도)

관련 서비스:

- AWS CloudFormation
- AWS Cloud Development Kit (AWS CDK)(AWS CDK) Constructs
- Service Catalog
- AWS BugBust

SEC11-BP02 개발 및 릴리스 수명 주기를 통한 테스트 자동화

개발 및 릴리스 수명 주기 전반에 걸쳐 보안 속성 테스트를 자동화하세요. 자동화를 구현하면 릴리스에 앞서 소프트웨어의 잠재적인 문제를 일관되고 반복적으로 손쉽게 식별할 수 있어 소프트웨어 제공 중에 보안 문제가 발생할 위험이 줄어듭니다.

원하는 결과: 자동화된 테스트의 목표는 개발 수명 주기 전반에 걸쳐 잠재적인 문제를 조기에 자주 탐지할 수 있는 프로그래밍 방식을 제공하는 것입니다. 회귀 테스트를 자동화하면 기능 테스트 및 비기능 테스트를 다시 실행하여 이전에 테스트한 소프트웨어가 변경 후에도 예상대로 작동하는지 확인할 수 있습니다. 보안 장치 테스트를 정의하여 인증 정보 손상 또는 누락과 같은 일반적인 구성 오류를 확인하면 이러한 문제를 개발 프로세스 초기에 식별하고 해결할 수 있게 됩니다.

테스트 자동화는 애플리케이션의 요구 사항과 원하는 기능을 기반으로 애플리케이션 검증을 위해 특별히 제작된 테스트 사례를 사용합니다. 자동화된 테스트 결과는 생성된 테스트 출력을 각각의 예상 출력과 비교하여 전체 테스트 수명 주기를 가속화합니다. 회귀 테스트 및 장치 테스트 세트와 같은 테스트 방법론이 자동화에 가장 적합합니다. 보안 속성 테스트를 자동화하면 빌더가 보안 검토를 기다리지 않고도 자동으로 피드백을 받을 수 있습니다. 정적 또는 동적 코드 분석의 형태로 자동화된 테스트는 코드 품질을 개선하고 개발 수명 주기 초기에 잠재적인 소프트웨어 문제를 탐지하는 데 도움이 됩니다.

일반적인 안티 패턴:

- 자동화된 테스트의 테스트 사례 및 테스트 결과를 전달하지 않습니다.
- 릴리스 직전에만 자동화된 테스트를 수행합니다.
- 요구 사항이 자주 변경되는 테스트 사례를 자동화합니다.

- 보안 테스트 결과를 처리하는 방법에 대한 지침을 제공하지 못합니다.

이 모범 사례 확립의 이점:

- 시스템의 보안 속성을 평가하는 사용자에 대한 의존도를 낮춥니다.
- 여러 작업 흐름에서 일정한 결과를 도출하여 일관성이 향상됩니다.
- 프로덕션 소프트웨어에 보안 문제가 발생할 가능성이 줄어듭니다.
- 소프트웨어 문제를 조기에 발견하여 탐지부터 해결에 걸리는 시간이 단축됩니다.
- 여러 작업 흐름에 걸쳐 체계적이거나 반복적인 동작에 대한 가시성이 향상되어 조직 전체의 개선을 추진하는 데 유용합니다.

이 모범 사례를 따르지 않을 경우 노출 위험도: 중간

구현 가이드

소프트웨어를 구축할 때 다양한 소프트웨어 테스트 메커니즘을 채택하여 애플리케이션의 비즈니스 논리에 바탕을 둔 기능적 요구 사항과 애플리케이션 신뢰성, 성능, 보안에 중점을 둔 비기능적 요구 사항을 기반으로 애플리케이션을 테스트합니다.

정적 애플리케이션 보안 테스트(SAST)는 비정상적인 보안 패턴에 대한 소스 코드를 분석하고 결함이 발생하기 쉬운 코드를 표시해 줍니다. SAST는 문서(요구 사항 사양, 설계 설명서, 설계 사양)와 같은 정적 입력 및 애플리케이션 소스 코드를 전적으로 사용하여 알려진 여러 보안 문제를 테스트합니다. 정적 코드 분석기는 대량의 코드를 신속하게 분석하는 데 도움이 됩니다. NIST Quality Group은 바이트 코드 스캐너 및 이진 코드 스캐너용 오픈 소스 도구를 포함하는 소스 코드 보안 분석기의 비교 정보를 제공합니다.

실행 중인 애플리케이션을 테스트하여 잠재적으로 예상치 못한 동작을 식별하는 동적 분석 보안 테스트(DAST) 방법론으로 정적 테스트를 보완합니다. 동적 테스트를 사용하면 정적 분석을 통해 탐지할 수 없는 잠재적 문제를 감지할 수 있습니다. 코드 리포지토리, 구축 및 파이프라인 단계에서 테스트하면 코드 입력 시 발생 가능한 여러 유형의 잠재적 문제를 확인할 수 있습니다. Amazon CodeWhisperer는 빌더의 IDE에서 보안 검색을 포함한 코드 권장 사항을 제공합니다. Amazon CodeGuru Reviewer는 애플리케이션 개발 중 중대한 문제, 보안 문제 및 찾기 어려운 버그를 식별할 수 있으며 코드 품질을 개선하기 위한 권장 사항을 제공합니다.

Security for Developers 워크숍에서는 AWS CodeBuild, AWS CodeCommit, AWS CodePipeline 등의 AWS 개발자 도구를 사용하여 SAST 및 DAST 테스트 방법론이 포함된 릴리스 파이프라인 자동화를 수행합니다.

SDLC를 진행하면서 보안 팀과 함께 정기적인 애플리케이션 검토를 포함하는 반복 프로세스를 수립하세요. 이러한 보안 검토에서 수집된 피드백은 릴리스 준비 상태 검토 과정에서 해결하고 검증해야 합니다. 검토를 통해 강력한 애플리케이션 보안 태세를 확립하고, 빌더에게 잠재적인 문제를 해결하는 데 도움이 되는 실용적인 피드백을 제공할 수 있습니다.

구현 단계

- 보안 테스트가 포함된 IDE, 코드 검토 및 CI/CD 도구를 일관성 있게 구현합니다.
- 문제를 해결해야 한다고 빌더에게 통보하는 대신 SDLC의 어느 지점에서 파이프라인을 차단하는 것이 적절한지 고려해 보세요.
- Security for Developers 워크숍에서는 정적 및 동적 테스트를 릴리스 파이프라인에 통합하는 예를 제공합니다.
- 개발자 IDE와 통합된 Amazon CodeWhisperer 및 커밋 시점에 코드를 스캔하는 Amazon CodeGuru Reviewer와 같은 자동화된 도구를 사용하여 테스트 또는 코드 분석을 수행하면 빌더가 적시에 피드백을 받을 수 있습니다.
- AWS Lambda를 사용하여 구축하면 Amazon Inspector를 통해 함수의 애플리케이션 코드를 스캔할 수 있습니다.
- AWS CI/CD 워크숍은 AWS에서 CI/CD 파이프라인을 구축하기 위한 시작점을 제공합니다.
- CI/CD 파이프라인에 자동화된 테스트가 포함된 경우 티켓팅 시스템을 사용하여 소프트웨어 문제의 알림 및 해결 방법을 추적해야 합니다.
- 결과를 생성할 수 있는 보안 테스트의 경우 해결 지침에 연결하면 빌더가 코드 품질을 개선하는 데 도움이 됩니다.
- 자동화된 도구의 결과를 정기적으로 분석하여 다음 자동화, 빌더 교육 또는 인식 캠페인의 우선순위를 지정합니다.

리소스

관련 문서:

- 지속적 전달 및 지속적 배포

- AWS DevOps 컴피턴시 파트너
- AWS 보안 컴피턴시 파트너(애플리케이션 보안용)
- Choosing a Well-Architected CI/CD approach(Well-Architected CI/CD 접근 방법 선택)

- Monitoring CodeCommit events in Amazon EventBridge and Amazon CloudWatch Events(Amazon EventBridge와 Amazon CloudWatch Events에서 CodeCommit 이벤트 모니터링)
- Secrets detection in Amazon CodeGuru Review(Amazon CodeGuru 검토의 비밀 탐지)
- Accelerate deployments on AWS with effective governance(효과적인 거버넌스를 통한 AWS의 배포 가속화)
- How AWS approaches automating safe, hands-off deployments(AWS 접근 방식으로 안전하게 자동 배포를 자동화하는 방법)

관련 동영상:

- Hands-off: Automating continuous delivery pipelines at Amazon(자동: Amazon에서 지속적 전달 파이프라인 자동화)
- Automating cross-account CI/CD pipelines(크로스 계정 CI/CD 파이프라인 자동화)

관련 예시:

- Industry awareness for developers(개발자를 위한 업계 인지도)
- AWS CodePipeline 거버넌스(GitHub)
- Security for Developers workshop(Security for Developers 워크숍)
- AWS CI/CD Workshop(AWS CI/CD 워크숍)

SEC11-BP03 정기적인 침투 테스트 시행

정기적으로 소프트웨어 침투 테스트를 시행하세요. 이러한 메커니즘은 자동화된 테스트나 수동 코드 검토로 탐지할 수 없는 잠재적인 소프트웨어 문제를 식별하는 데 도움이 됩니다. 또한 탐지 컨트롤의 효율성을 이해하는 데 도움이 될 수 있습니다. 침투 테스트를 통해 보호해야 하는 데이터를 노출하거나 예상보다 더 광범위한 권한을 부여하는 등 소프트웨어가 예기치 않은 방식으로 작동할 가능성이 있는지 확인해야 합니다.

원하는 결과: 침투 테스트는 애플리케이션의 보안 속성을 탐지, 문제 해결 및 검증하는 데 사용됩니다. 소프트웨어 개발 수명 주기(SDLC)의 일부로 일정을 정해 정기적인 침투 테스트를 수행해야 합니다. 침투 테스트로 인해 발견한 결과는 소프트웨어 출시 전에 해결해야 합니다. 침투 테스트의 결과를 분석하여 자동화를 통해 찾을 수 있는 문제가 있는지 확인해야 합니다. 능동적 피드백 메커니즘을 포함하는 정기적이고 반복 가능한 침투 테스트 프로세스를 통해 빌더에게 지침을 제공하고 소프트웨어 품질을 개선할 수 있습니다.

일반적인 안티 패턴:

- 알려진 보안 문제 또는 일반적인 보안 문제에 대한 침투 테스트만 수행합니다.
- 종속된 타사 도구 및 라이브러리가 없는 애플리케이션에 대한 침투 테스트만 수행합니다.
- 패키지 보안 문제에 대한 침투 테스트만 수행하고 구현된 비즈니스 논리는 평가하지 않습니다.

이 모범 사례 확립의 이점:

- 릴리스 전에 소프트웨어의 보안 속성에 대한 신뢰도가 높아집니다.
- 선호하는 애플리케이션 패턴을 식별할 수 있는 기회를 제공하여 소프트웨어 품질을 개선합니다.
- 피드백 루프를 통해 자동화 또는 추가 교육으로 소프트웨어의 보안 속성을 개선할 여지가 있는 개발 주기의 초기 단계를 식별할 수 있습니다.

이 모범 사례를 따르지 않을 경우 노출 위험도: 높음

구현 가이드

침투 테스트는 계획된 보안 위반 시나리오를 실행하여 보안 제어 기능을 탐지, 문제 해결 및 검증하는 체계적인 보안 테스트 활동입니다. 침투 테스트는 정찰부터 시작되는데, 이때 애플리케이션의 현재 설계와 종속성을 기반으로 데이터가 수집됩니다. 보안 관련 테스트 시나리오의 선별 목록도 작성되고 실행됩니다. 침투 테스트의 주요 목적은 환경이나 데이터에 무단으로 액세스할 권한을 얻는 데 악용될 수 있는 애플리케이션의 보안 문제를 파악하는 것입니다. 새 기능을 출시할 때 또는 애플리케이션 기능이나 기술 구현이 크게 변경될 때마다 침투 테스트를 수행해야 합니다.

침투 테스트를 수행하려면 개발 수명 주기에서 가장 적합한 단계를 식별해야 합니다. 시스템의 기능이 원하는 릴리스 상태에 근접했을 정도로 늦은 시점에 수행하되, 이때 문제를 해결할 시간이 충분히 남아 있어야 합니다.

구현 단계

- 침투 테스트의 범위를 파악하는 체계적인 프로세스를 수립해야 합니다. 위협 모델을 기반으로 이 프로세스를 수행하면 컨텍스트를 유지할 수 있습니다.
- 침투 테스트를 수행하기 위한 개발 주기의 적절한 시점을 파악합니다. 애플리케이션에 예상되는 변경이 최소한만 남아 있는 동시에 문제를 해결하기에 시간이 충분한 시점이어야 합니다.
- 빌더에게 침투 테스트 결과에서 기대할 수 있는 정보를 알려주고 문제 해결 정보를 얻는 방법을 교육합니다.
- 도구를 통해 공통 또는 반복 가능한 테스트를 자동화하여 침투 테스트 프로세스를 가속화합니다.

- 침투 테스트 결과를 분석하여 시스템 보안 문제를 식별하고, 이 데이터를 바탕으로 자동화된 테스트를 추가로 수행하고 빌더를 꾸준히 교육합니다.

리소스

관련 모범 사례:

- SEC11-BP01 애플리케이션 보안 교육
- SEC11-BP02 개발 및 릴리스 수명 주기를 통한 테스트 자동화

관련 문서:

- AWS 침투 테스트(AWS의 침투 테스트에 대한 자세한 지침 제공)
- Accelerate deployments on AWS with effective governance(효과적인 거버넌스를 통한 AWS의 배포 가속화)
- AWS 보안 컴피턴시 파트너
- Modernize your penetration testing architecture on AWS Fargate(AWS Fargate의 침투 테스트 아키텍처 현대화)
- AWS Fault injection Simulator

관련 예시:

- AWS CodePipeline을 통한 API 테스트 자동화(GitHub)
- 자동화된 보안 헬퍼(GitHub)

SEC11-BP04 수동 코드 검토

개발할 소프트웨어의 코드를 수동으로 검토하세요. 이 프로세스를 통해 코드를 작성한 개발자 이외의 사람이 코드 품질을 검사하게 됩니다.

원하는 결과: 개발 중에 수동 코드 검토 단계를 포함하면 작성 중인 소프트웨어의 품질이 높아지고, 경험이 부족한 팀원의 기술이 향상되며, 자동화를 사용할 수 있는 부분을 식별할 기회를 얻을 수 있습니다. 수동 코드 검토는 자동화된 도구 및 테스트를 통해 지원됩니다.

일반적인 안티 패턴:

- 배포 전에 코드 검토를 수행하지 않습니다.
- 같은 사람이 코드를 작성하고 검토하도록 합니다.
- 코드 검토를 지원하거나 조율하는 데 자동화를 사용하지 않습니다.
- 빌더가 코드를 검토하기 전에 애플리케이션 보안 교육을 받지 않습니다.

이 모범 사례 확립의 이점:

- 코드 품질이 향상됩니다.
- 공통된 접근 방식을 재사용할 수 있어 코드 개발의 일관성이 높아집니다.
- 침투 테스트 및 이후 단계에서 발견되는 문제의 수가 줄어듭니다.
- 팀 내 지식 전달이 개선됩니다.

이 모범 사례를 따르지 않을 경우 노출 위험도: 중간

구현 가이드

검토 단계는 전체 코드 관리 흐름의 일부로 구현되어야 합니다. 자세한 내용은 분기, 풀 요청 및 병합에 사용되는 전략에 따라 달라집니다. AWS CodeCommit 또는 GitHub, GitLab, Bitbucket 등 타사 솔루션을 사용하고 있을 수 있습니다. 어떤 방법을 사용하든 프로덕션 환경에 배포하는 데 앞서 프로세스에 코드 검토가 필요한지 확인하는 것이 중요합니다. Amazon CodeGuru Reviewer와 같은 도구를 사용하면 코드 검토 프로세스를 보다 쉽게 조율할 수 있습니다.

구현 단계

- 코드 관리 흐름의 일부로 수동 검토 단계를 구현하고, 계속하기 전에 검토를 수행합니다.
- 코드 검토를 관리하고 지원하는 Amazon CodeGuru Reviewer 사용을 고려해 보세요.
- 코드를 다음 단계로 진행하려면 코드 검토를 완료해야 하는 승인 흐름을 구현합니다.
- 수동 코드 검토 중에 자동으로 탐지 가능한 문제를 식별하는 프로세스가 있는지 확인합니다.
- 코드 개발 관행에 맞게 수동 코드 검토 단계를 통합합니다.

리소스

관련 모범 사례:

- SEC11-BP02 개발 및 릴리스 수명 주기를 통한 테스트 자동화

관련 문서:

- Working with pull requests in AWS CodeCommit repositories(AWS CodeCommit 리포지토리에서 풀 요청 작업)
- Working with approval rule templates in AWS CodeCommit(AWS CodeCommit에서 승인 규칙 템플릿 작업)
- GitHub의 풀 요청 정보
- Automate code reviews with Amazon CodeGuru Reviewer(Amazon CodeGuru Reviewer를 사용한 코드 검토 자동화)
- Automating detection of security vulnerabilities and bugs in CI/CD pipelines using Amazon CodeGuru Reviewer CLI(Amazon CodeGuru Reviewer CLI를 사용하여 CI/CD 파이프라인의 보안 취약성 및 버그 탐지 자동화)

관련 동영상:

- Continuous improvement of code quality with Amazon CodeGuru(Amazon CodeGuru를 통한 코드 품질의 지속적인 개선)

관련 예시:

- Security for Developers workshop(Security for Developers 워크숍)

SEC11-BP05 패키지 및 종속성 서비스의 중앙 집중화

빌더 팀이 소프트웨어 패키지 및 기타 종속성을 확보할 수 있도록 서비스를 중앙 집중화하세요. 서비스를 중앙 집중화하면 작성하는 소프트웨어에 포함하기 전에 패키지를 검증할 수 있습니다. 또한 조직에서 사용 중인 소프트웨어 분석에 쓰일 데이터를 하나의 소스에서 얻을 수 있습니다.

원하는 결과: 소프트웨어는 작성 중인 코드 외에 다양한 소프트웨어 패키지 세트로 구성됩니다. 따라서 JSON 구문 분석기 또는 암호화 라이브러리와 같이 반복적으로 사용되는 기능 구현을 간편하게 사용할 수 있습니다. 이러한 패키지 및 종속성에 대한 소스를 논리적으로 중앙 집중화하면 패키지를 사용하기 전에 패키지의 속성을 검증하는 메커니즘을 보안 팀에 제공할 수 있습니다. 또한 이러한 전략은 기존 패키지의 변경이나 빌더 팀(예: 인터넷에서 바로 임의 패키지 다운로드)으로 인해 예상치 못한 문제가 발생할 위험을 줄여줍니다. 수동 및 자동 테스트 흐름과 함께 이 전략을 사용하면 개발 중인 소프트웨어의 품질에 대한 신뢰도를 높일 수 있습니다.

일반적인 안티 패턴:

- 인터넷의 임의 리포지토리에서 패키지를 가져옵니다.
- 빌더에게 새 패키지를 제공하기 전에 테스트하지 않습니다.

이 모범 사례 확립의 이점:

- 구축 중인 소프트웨어에서 어떤 패키지가 사용되고 있는지 효과적으로 이해할 수 있습니다.
- 누가 무엇을 사용하고 있는지 파악한 후에 패키지를 업데이트해야 할 때 워크로드 팀에 알릴 수 있습니다.
- 소프트웨어에 문제가 포함된 패키지의 위험을 줄입니다.

이 모범 사례를 따르지 않을 경우 노출 위험도: 중간

구현 가이드

패키지 및 종속성에 대한 중앙 집중식 서비스를 빌더가 쉽게 사용할 수 있는 방식으로 제공합니다. 중앙 집중식 서비스는 단일 시스템으로 구현되기보다는 논리적으로 중앙에 배치될 수 있습니다. 이러한 접근 방식을 통해 빌더의 요구를 충족하는 방향으로 서비스를 제공할 수 있습니다. 업데이트가 발생하거나 새로운 요구 사항이 나타날 때 패키지를 리포지토리에 추가하는 효율적인 방법을 구현해야 합니다. AWS CodeArtifact와 같은 AWS 서비스 또는 이와 유사한 AWS 파트너 솔루션은 이러한 기능을 제공하는 방법을 안내합니다.

구현 단계:

- 소프트웨어가 개발되는 모든 환경에서 사용할 수 있는 논리적으로 중앙 집중화된 리포지토리 서비스를 구현합니다.
- 리포지토리에 대한 액세스를 AWS 계정 벤딩 프로세스의 일부로 포함합니다.
- 패키지를 리포지토리에 게시하기 전에 테스트하는 자동화를 구축합니다.
- 가장 일반적으로 사용되는 패키지, 언어 및 변경 사항이 제일 많은 팀의 지표를 유지 관리합니다.
- 빌더 팀이 새 패키지를 요청하고 피드백을 줄 수 있도록 자동화된 메커니즘을 제공합니다.
- 리포지토리의 패키지를 정기적으로 스캔하여 새로 발견된 문제의 잠재적 영향을 식별합니다.

리소스

관련 모범 사례:

- SEC11-BP02 개발 및 릴리스 수명 주기를 통한 테스트 자동화

관련 문서:

- Accelerate deployments on AWS with effective governance(효과적인 거버넌스를 통한 AWS의 배포 가속화)
- Tighten your package security with CodeArtifact Package Origin Control toolkit(CodeArtifact Package Origin Control 도구 키트로 패키지 보안 강화)
- Detecting security issues in logging with Amazon CodeGuru Reviewer(Amazon CodeGuru Reviewer를 사용한 로깅 내 보안 문제 탐지)
- Supply chain Levels for Software Artifacts (SLSA)(소프트웨어 아티팩트의 공급망 수준(SLSA))

관련 동영상:

- Proactive security: Considerations and approaches(사전 예방적 보안: 고려 사항 및 접근 방법)
- The AWS Philosophy of Security (re:Invent 2017)(AWS 보안 철학(re:Invent 2017))
- When security, safety, and urgency all matter: Handling Log4Shell(보안, 안전 및 긴급성이 모두 중요한 경우: Log4Shell 처리)

관련 예시:

- 다중 리전 패키지 게시 파이프라인(GitHub)
- AWS CodePipeline을 사용하여 AWS CodeArtifact에 Node.js 모듈 게시(GitHub)
- AWS CDK Java CodeArtifact 파이프라인 샘플(GitHub)
- AWS CodeArtifact를 사용한 프라이빗 .NET NuGet 패키지 배포(GitHub)

SEC11-BP06 프로그래밍 방식으로 소프트웨어 배포

가능한 한 프로그래밍 방식으로 소프트웨어를 배포하세요. 이 접근 방식을 통해 인적 오류로 배포 실패나 예기치 않은 문제가 발생할 가능성을 줄일 수 있습니다.

원하는 결과: AWS 클라우드에서의 안전한 구축을 위해서는 데이터에 대한 사람의 접근을 막는 것이 핵심 원칙입니다. 이러한 원칙에는 소프트웨어 배포 방법이 포함됩니다.

인력에 의존하여 소프트웨어를 배포하지 않으면 테스트한 것이 배포되고 매번 일관성 있게 배포가 이루어진다는 장점이 있습니다. 다른 환경에서 작동되도록 소프트웨어를 변경할 필요가 없습니다. 12가지 요소로 구성된 애플리케이션 개발의 원칙, 특히 구성의 외부화를 적용하면 변경하지 않고 동일한 코드를 여러 환경에 배포할 수 있습니다. 암호화된 서명 소프트웨어 패키지는 서로 다른 환경 간에 변경

된 내용이 없음을 확인하는 좋은 방법입니다. 이 접근 방식을 통해 변경 프로세스의 위험을 줄이고 소프트웨어 릴리스의 일관성을 개선하는 결과를 얻을 수 있습니다.

일반적인 안티 패턴:

- 프로덕션에 소프트웨어를 수동으로 배포합니다.
- 다양한 환경에 맞게 소프트웨어를 수동으로 변경합니다.

이 모범 사례 확립의 이점:

- 소프트웨어 릴리스 프로세스에 대한 신뢰도가 높아집니다.
- 비즈니스 기능에 영향을 미치는 변경 실패 위험을 줄여줍니다.
- 변경 위험 감소로 인해 릴리스 주기가 길어집니다.
- 배포 중 예기치 않은 이벤트에 대한 자동 롤백 기능이 지원됩니다.
- 테스트된 소프트웨어가 배포된 소프트웨어임을 암호화하여 증명하는 기능이 제공됩니다.

이 모범 사례를 따르지 않을 경우 노출 위험도: 높음

구현 가이드

AWS 계정 구조를 구성하여 환경에서 영구적인 인적 액세스를 제거하고 CI/CD 도구를 사용하여 배포를 수행합니다. AWS Systems Manager Parameter Store와 같은 외부 소스에서 환경별 구성 데이터를 가져오도록 애플리케이션을 설계합니다. 패키지를 테스트한 후 서명하고, 배포하는 동안 서명을 검증합니다. 애플리케이션 코드를 푸시하도록 CI/CD 파이프라인을 구성하고 Canary를 사용하여 배포 성공을 확인합니다. AWS CloudFormation 또는 AWS CDK와 같은 도구를 사용하여 인프라를 정의한 다음, AWS CodeBuild 및 AWS CodePipeline을 사용하여 CI/CD 작업을 수행합니다.

구현 단계

- 효과적으로 정의된 CI/CD 파이프라인을 구축하여 배포 프로세스를 간소화합니다.
- AWS CodeBuild 및 AWS Code Pipeline을 사용하여 CI/CD 기능을 제공하면 보안 테스트를 파이프라인에 쉽게 통합할 수 있습니다.
- 여러 계정을 사용하여 AWS 환경 구성 백서의 환경 분리 지침을 따릅니다.
- 프로덕션 워크로드가 실행 중인 환경에 대한 영구적인 인적 액세스 권한이 없는지 확인합니다.
- 구성 데이터의 외부화를 지원하도록 애플리케이션을 설계합니다.
- 블루/그린 배포 모델을 사용한 배포를 고려합니다.

- Canary를 구현하여 소프트웨어의 배포 성공을 검증합니다.
- AWS Signer 또는 AWS Key Management Service(AWS KMS)와 같은 암호화 도구를 사용하여 배포 중인 소프트웨어 패키지에 서명하고 확인합니다.

리소스

관련 모범 사례:

- SEC11-BP02 개발 및 릴리스 수명 주기를 통한 테스트 자동화

관련 문서:

- AWS CI/CD Workshop(AWS CI/CD 워크숍)
- Accelerate deployments on AWS with effective governance(효과적인 거버넌스를 통한 AWS의 배포 가속화)
- 안전한 자동 배포 자동화
- Code signing using AWS Certificate Manager Private CA and AWS Key Management Service asymmetric keys(AWS Certificate Manager Private CA 및 AWS Key Management Service 비대칭 키를 사용한 코드 서명)
- Code Signing, a Trust and Integrity Control for AWS Lambda(코드 서명, AWS Lambda의 신뢰 및 무결성 제어)

관련 동영상:

- Hands-off: Automating continuous delivery pipelines at Amazon(자동: Amazon에서 지속적 전달 파이프라인 자동화)

관련 예시:

- Blue/Green deployments with AWS Fargate(AWS Fargate를 사용한 블루/그린 배포)

SEC11-BP07 정기적으로 파이프라인의 보안 속성 평가

특히 권한 분리에 주의를 기울여 Well-Architected 보안 원칙을 파이프라인에 적용하세요. 파이프라인 인프라의 보안 속성을 정기적으로 평가합니다. 파이프라인의 보안을 효율적으로 관리하면 파이프라인을 통과하는 소프트웨어의 보안을 보장할 수 있습니다.

원하는 결과: 소프트웨어를 구축하고 배포하는 데 사용되는 파이프라인은 환경의 다른 워크로드와 동일한 권장 사례를 따라야 합니다. 파이프라인에서 구현되는 테스트는 이를 사용하는 빌더가 편집할 수 없어야 합니다. 파이프라인은 빌더가 수행 중인 배포에 필요한 권한만 보유해야 하며, 잘못된 환경에 배포되지 않도록 안전 조치를 구현해야 합니다. 파이프라인은 장기 보안 인증 정보에 의존하지 않아야 하며, 구축 환경의 무결성을 확인할 수 있게 상태를 전송하도록 구성해야 합니다.

일반적인 안티 패턴:

- 빌더가 보안 테스트를 우회할 수 있습니다.
- 배포 파이프라인에 대한 권한이 지나치게 광범위합니다.
- 입력을 검증하도록 파이프라인을 구성하지 않습니다.
- CI/CD 인프라와 관련된 권한을 정기적으로 검토하지 않습니다.
- 장기 또는 하드코딩된 보안 인증 정보를 사용합니다.

이 모범 사례 확립의 이점:

- 파이프라인을 통해 구축 및 배포되는 소프트웨어의 무결성에 대한 신뢰도가 높아집니다.
- 의심스러운 활동이 있을 때 배포를 중지할 수 있습니다.

이 모범 사례를 따르지 않을 경우 노출 위험도: 높음

구현 가이드

IAM 역할을 지원하는 관리형 CI/CD 서비스로 시작하면 보안 인증 정보 유출 위험이 줄어듭니다. CI/CD 파이프라인 인프라에 보안 원칙을 적용하면 보안을 개선할 수 있는 부분을 가려내는 데 도움이 됩니다. CI/CD 환경을 구축하기 시작할 때 AWS 배포 파이프라인 참조 아키텍처를 따르는 것이 좋습니다. 파이프라인 구현을 정기적으로 검토하고 예기치 않은 동작에 대한 로그를 분석하면 소프트웨어 배포에 사용되는 파이프라인의 사용 패턴을 이해하는 데 유용합니다.

구현 단계

- 먼저 AWS 배포 파이프라인 참조 아키텍처부터 시작합니다.

- 파이프라인에 대한 최소 권한 IAM 정책을 프로그래밍 방식으로 생성하려면 AWS IAM Access Analyzer를 사용하는 것이 좋습니다.
- 파이프라인을 모니터링 및 알림과 통합하여 예기치 않거나 비정상적인 활동이 발생할 경우 알림을 받을 수 있습니다. AWS 관리형 서비스의 경우 Amazon EventBridge를 사용하면 AWS Lambda 또는 Amazon Simple Notification Service(Amazon SNS)와 같은 대상으로 데이터를 라우팅할 수 있습니다.

리소스

관련 문서:

- AWS 배포 파이프라인 참조 아키텍처
- Monitoring AWS CodePipeline(AWS CodePipeline 모니터링)
- Security best practices for AWS CodePipeline(AWS CodePipeline 보안 모범 사례)

관련 예시:

- DevOps 모니터링 대시보드(GitHub)

SEC11-BP08 보안 소유권이 워크로드 팀에 귀속되도록 프로그램 구축

빌더 팀이 개발하는 소프트웨어의 보안을 결정할 수 있도록 지원하는 프로그램 또는 메커니즘을 구축합니다. 보안 팀에서 검토 시 이러한 결정을 다시금 검증해야 하지만, 빌더 팀이 보안 소유권을 가지면 더욱 신속하고 안전하게 워크로드를 구축할 수 있습니다. 또한 이 메커니즘은 구축하는 시스템의 운영에 좋은 영향을 미치는 주인 의식 문화를 강화합니다.

원하는 결과: 보안 소유권과 의사 결정을 빌더 팀에 귀속하려면 빌더에게 보안을 대하는 방식을 교육하면 됩니다. 빌더 팀에 속하거나 연계된 보안 담당자를 활용하여 빌더 대상 교육을 강화할 수도 있습니다. 두 전략 모두 유효하며, 이를 통해 빌더 팀이 개발 주기 초기에 더 현명한 보안 결정을 내리도록 지원할 수 있습니다. 이 소유권 모델은 애플리케이션 보안 교육을 기반으로 합니다. 특정 워크로드에 대한 위협 모델부터 시작하면 적절한 컨텍스트에 초점을 맞춰 설계 사고를 집중할 수 있습니다. 보안에 주력하는 빌더 커뮤니티 또는 빌더 팀과 협력하는 보안 엔지니어 그룹이 있으면 소프트웨어가 작성되는 방식을 보다 깊이 이해할 수 있다는 이점도 있습니다. 이러한 이해도를 바탕으로 자동화 기능의 다음 개선 영역을 결정할 수 있습니다.

일반적인 안티 패턴:

- 보안 팀에 모든 보안 설계 결정을 맡깁니다.
- 개발 프로세스에서 보안 요구 사항을 조기에 해결하지 못합니다.
- 빌더 및 보안 담당자로부터 프로그램 운영 피드백을 받지 못합니다.

이 모범 사례 확립의 이점:

- 보안 검토를 빠르게 완료할 수 있습니다.
- 보안 검토 단계에서만 탐지되는 보안 문제가 감소합니다.
- 작성 중인 소프트웨어의 전반적인 품질이 향상됩니다.
- 체계 문제 또는 유의미한 개선 영역을 식별하고 이해할 수 있습니다.
- 보안 검토 결과로 인해 필요한 재작업의 양이 줄어듭니다.
- 보안 기능에 대한 인식이 개선됩니다.

이 모범 사례를 따르지 않을 경우 노출되는 위험 수준: 낮음

구현 가이드

SEC11-BP01 애플리케이션 보안 교육의 지침부터 시작합니다. 그런 다음 조직에 가장 적합하다고 생각되는 프로그램의 운영 모델을 파악합니다. 2가지 주요 패턴은 빌더를 교육하거나 빌더 팀에 보안 인력을 투입하는 것입니다. 초기 접근 방식을 정한 후에는 단일 또는 소규모 워크로드 팀과 함께 시범 운영하여 해당 모델이 조직에 적합한지 입증해야 합니다. 조직의 빌더 및 보안 부문에서 리더십의 지원이 있으면 프로그램의 제공과 성공에 도움이 됩니다. 이 프로그램을 개발할 때 프로그램의 가치를 보여주는 데 사용할 수 있는 지표를 선택해야 합니다. AWS가 이러한 문제에 어떻게 접근했는지 알아보면 큰 도움이 됩니다. 이 모범 사례는 조직의 변화와 문화를 집중적으로 조명합니다. 빌더와 보안 커뮤니티 간의 협업을 지원하는 도구를 사용해야 합니다.

구현 단계

- 먼저 빌더에게 애플리케이션 보안 교육을 제공합니다.
- 빌더를 교육하기 위한 커뮤니티와 온보딩 프로그램을 개발합니다.
- 프로그램 이름을 선택합니다. Guardians, Champions, Advocates가 주로 사용됩니다.
- 빌더를 교육하거나, 보안 엔지니어를 합류시키거나, 보안 담당자를 연계하는 등 사용할 모델을 결정합니다.

- 보안, 빌더 및 기타 관련 그룹의 프로젝트 후원 주체를 결정합니다.
- 프로그램에 참여한 인원 수, 검토에 소요된 시간, 빌더 및 보안 인력의 피드백 지표를 추적합니다. 이러한 지표를 바탕으로 개선합니다.

리소스

관련 모범 사례:

- SEC11-BP01 애플리케이션 보안 교육
- SEC11-BP02 개발 및 릴리스 수명 주기를 통한 테스트 자동화

관련 문서:

- How to approach threat modeling(위협 모델링 접근 방식)
- How to think about cloud security governance(클라우드 보안 거버넌스를 대하는 방식)

관련 동영상:

- Proactive security: Considerations and approaches(사전 예방적 보안: 고려 사항 및 접근 방법)

결론

보안을 유지하려면 지속적인 작업을 수행해야 합니다. 인시던트가 실제로 발생한 경우, 이를 아키텍처 보안을 개선할 기회로 간주해야 합니다. 강력한 자격 증명 제어를 적용하고, 보안 인시던트에 대한 대응을 자동화하고, 여러 수준에서 인프라를 보호하고, 암호화를 통해 적절하게 분류된 데이터를 관리하면 모든 조직에서 구현해야 하는 심층 방어 기능이 제공됩니다. 이 백서에서 설명한 프로그래밍 방식 함수와 AWS 기능 및 서비스를 사용하면 이 작업을 더 쉽게 수행할 수 있습니다.

AWS는 비즈니스 가치를 제공하는 동시에 정보, 시스템 및 자산을 보호하는 아키텍처를 구축하고 운영하는 과정을 지원합니다.

기여자

다음은 본 문서를 작성하는 데 도움을 준 개인 및 조직입니다.

- Sarita Dharankar, Amazon Web Services Well-Architected 부문 보안 원칙 책임자
- Byron Pogson, Amazon Web Services 선임 솔루션스 아키텍트
- Bill Shinn, CISO 오피스 선임 수석, Amazon Web Services
- Brigid Johnson, Amazon Web Services AWS Identity 부문 선임 소프트웨어 개발 관리자
- Byron Pogson, 수석 솔루션스 아키텍트, Amazon Web Services
- Charlie Hammell, Amazon Web Services 수석 엔터프라이즈 아키텍트
- Darran Boyd, 금융 서비스 수석 보안 솔루션 아키텍트, Amazon Web Services
- Dave Walker, 보안 및 규정 준수 부문 수석 전문가 솔루션 아키텍트, Amazon Web Services
- John Formento, Amazon Web Services 선임 솔루션 아키텍트
- Paul Hawkins, Amazon Web Services 수석 CISO
- Sam Elmalak, 선임 기술 책임자, Amazon Web Services
- Pat Gaw, Amazon Web Services 수석 보안 컨설턴트
- Daniel Begimher, Amazon Web Services 선임 보안 컨설턴트
- Danny Cortegaca, Amazon Web Services 선임 보안 솔루션스 아키텍트
- Ana Malhotra, Amazon Web Services 보안 솔루션스 아키텍트
- Debashis Das, Amazon Web Services 수석 CISO
- Reef Dsouza, Principal Solutions Architect, Amazon Web Services
- Brad Burnett, Security Solutions Architect, Identity, Amazon Web Services
- Anna McAbee, Senior Security Solutions Architect, Threat Detection and Incident Response, Amazon Web Services
- Jason Garman, Principal Security Solutinos Architect, Amazon Web Services

추가 자료

자세한 내용은 다음 출처를 참조하십시오.

- AWS Well-Architected Framework 백서
- AWS 아키텍처 센터

문서 개정

이 백서의 업데이트에 대한 알림을 받으려면 RSS 피드를 구독하세요.

변경 사항	설명	날짜
모범 사례 지침 제공	워크로드의 안전한 운영 및 전송 중 데이터 보호 영역에 대한 새로운 지침으로 모범 사례가 업데이트되었습니다.	December 6, 2023
모범 사례 지침 제공	인시던트 대응 지침 및 모범 사례에 주요 업데이트가 있었습니다. 준비에 여러 모범 사례가 업데이트되었습니다. 인시던트 대응에 운영 및 인시던트 사후 활동이라는 두 가지 영역이 새롭게 추가되었습니다. 새로운 모범 사례 SEC10-BP08 인시던트로부터 학습하기 위한 프레임워크 구축이 추가되었습니다.	October 3, 2023
모범 사례 지침 제공	준비 및 시뮬레이션 영역에 대한 새로운 지침으로 모범 사례가 업데이트되었습니다.	July 13, 2023
새 프레임워크 관련 업데이트	모범 사례를 권장 가이드와 함께 업데이트하고 새로운 모범 사례를 추가했습니다. 애플리케이션 보안(AppSec)의 새로운 모범 사례 영역이 추가되었습니다.	April 10, 2023

백서 업데이트	모범 사례를 새로운 구현 가이드와 함께 업데이트했습니다.	December 15, 2022
백서 업데이트	모범 사례를 확대하고 개선 계획을 추가했습니다.	October 20, 2022
마이너 업데이트	현재 모범 사례를 반영하도록 IAM 정보를 업데이트했습니다.	June 28, 2022
마이너 업데이트	AWS PrivateLink 정보를 추가하고 연결되지 않는 링크를 수정했습니다.	May 19, 2022
마이너 업데이트	AWS PrivateLink를 추가했습니다.	May 6, 2022
마이너 업데이트	포용적이지 않은 표현을 삭제했습니다.	April 22, 2022
마이너 업데이트	VPC Network Access Analyzer에 대한 정보를 추가했습니다.	February 2, 2022
마이너 업데이트	소개에 지속 가능성 원칙을 추가했습니다.	December 2, 2021
마이너 업데이트	연결되지 않는 링크를 수정했습니다.	May 27, 2021
마이너 업데이트	문서 전반에 편집상 변경 사항을 적용했습니다.	May 17, 2021
주요 업데이트	거버넌스에 대한 섹션을 추가하고 다양한 섹션에 대한 세부 정보를 추가했으며 전반적으로 신규 기능 및 서비스를 추가했습니다.	May 7, 2021
마이너 업데이트	링크를 업데이트했습니다.	March 10, 2021

마이너 업데이트	연결되지 않는 링크를 수정했습니다.	July 15, 2020
새 프레임워크 관련 업데이트	계정, 자격 증명 및 권한 관리에 관한 지침을 업데이트했습니다.	July 8, 2020
새 프레임워크 관련 업데이트	모든 영역에서 더 많은 조언을 제공하고 새로운 모범 사례, 서비스 및 기능을 추가하기 위해 업데이트했습니다.	April 30, 2020
백서 업데이트	최신 AWS 서비스 및 기능과 새로워진 참조를 반영하여 업데이트했습니다.	July 1, 2018
백서 업데이트	새로운 AWS 서비스 및 기능을 반영하기 위해 시스템 보안 구성 및 유지 관리 섹션을 업데이트했습니다.	May 1, 2017
최초 게시	보안 원칙 - AWS Well-Architected Framework를 게시했습니다.	November 1, 2016

고지 사항

고객은 본 문서에 포함된 정보를 독자적으로 평가할 책임이 있습니다. 본 문서는 (a) 정보 제공만을 위한 것이며, (b) 사전 고지 없이 변경될 수 있는 현재의 AWS 제품 제공 서비스 및 사례를 보여 주며, (c) AWS 및 자회사, 공급업체 또는 라이선스 제공자로부터 어떠한 약정 또는 보증도 하지 않습니다. AWS 제품 또는 서비스는 명시적이든 묵시적이든 어떠한 종류의 보증, 진술 또는 조건 없이 '있는 그대로' 제공됩니다. 고객에 대한 AWS의 책임과 법적 책임은 AWS 계약서에 준하며 본 문서는 AWS와 고객 간의 계약에 포함되지 않고 계약을 변경하지도 않습니다.