보안 부문

AWS Well-Architected 프레임워크

2020년 7월

This paper has been archived.

The latest version is now available at:

https://docs.aws.amazon.com/ko_kr/wellarchitected/latest/security-pillar/welcome.html

고지 사항

고객은 본 문서에 포함된 정보를 독립적으로 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.

목차

개요

Well-Architected 프레임워크의 보안 부문을 중점으로 다룬 이 백서에서는 안전한 AWS 워크로드 설계, 제공 및 유지 관리에 모범 사례 및 현재 권장 사항을 적용할 때 참조할 수 있는 지침을 제공합니다.

서문

AWS Well-Architected 프레임워크는 AWS에서 워크로드를 구축할 때 내리는 의사 결정의 상충 관계를 이해하는 데 도움이 됩니다. 이 프레임워크를 사용하면 클라우드에서 안정적이고 안전하며 효율적이고 경제적인 워크로드를 설계하고 운영하기 위한 최신 설계 모범 사례를 알아볼 수 있습니다. 이 프레임워크는 모범 사례에 대해 아키텍처를 일관적으로 측정하고 개선 영역을 식별할 수 있는 방법을 제공합니다. Well-Architected 워크로드를 갖추면 비즈니스 성공 가능성이 높아집니다.

이 프레임워크에는 다음의 5가지 원칙이 포함됩니다.

- 운영 우수성
- 보안
- 안정성
- 성능 효율성
- 비용 최적화

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

이 백서의 내용을 확인하고 나면 보안을 염두에 두고 클라우드 아키텍처를 설계할 때 사용할 AWS 현재 권장 사항과 전략을 파악할 수 있습니다. 이 백서는 구현 세부 정보나 아키텍처 패턴을 직접적으로 제시하지는 않았지만, 이러한 정보를 얻을 수 있는 적절한 리소스를 제공하는 참조 자료를 기재하였습니다. 이 백서에서 설명하는 사례를 도입하면 데이터와 시스템을 보호하고, 액세스를 제어하고, 보안 이벤트에 자동으로 대응하는 아키텍처를 구축할 수 있습니다.

보안

보안 부문은 클라우드 기술을 활용하여 보안 상태를 개선할 수 있는 방식으로 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.
- **추적 기능 활성화:** 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 제어와 함께 심층 방어 접근 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 안전한 확장 능력을 빠르고 비용 효율적으로 향상시킵니다. 버전 제어가 가능한 템플릿에서 코드로 정의 및 관리되는 제어 기능의 구현을 비롯한 보안 아키텍처를 생성합니다.
- **전송 및 보관 중인 데이터 보호**: 데이터를 민감도 수준에 따라 분류하고 적절한 경우 암호화, 토큰화 및 액세스 제어와 같은 메커니즘을 사용합니다.
- **사람들이 데이터에 쉽게 접근할 수 없도록 유지:** 데이터에 대한 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 통해 사고에 대비합니다. 인시던트 대응 시뮬레이션을 실행하고 자동화된 도구를 사용하여 감지, 조사 및 복구 속도를 높입니다.

정의

클라우드의 보안은 다음의 5가지 영역으로 구성됩니다.

1. Identity and Access Management
2. 탐지
3. 인프라 보호
4. 데이터 보호
5. 인시던트 대응

보안 및 규정 준수는 AWS와 고객의 공동 책임입니다. 이 공유 모델은 운영 부담을 줄이는 데 도움이 됩니다. 사용하는 서비스, 서비스를 IT 환경에 통합하는 과정, 준거법과 규제에 따라 책임 범위가 다르기 때문에 선택하고자 하는 서비스에 대해 신중히 검토해야 합니다. 또한 이러한 공동 책임의 특성상 배포를 유연하게 관리할 수 있습니다.

워크로드를 안전하게 운영

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 자동화함으로써 보안 작업을 확장할 수 있습니다.

위협 모델을 사용하여 위험 식별 및 우선순위 지정: 보안 위협 모델을 사용하여 가장 최근에 등록된 잠재적 위협을 파악하고 유지 관리합니다. 위협 우선순위를 지정하고 보안 제어를 조정하여 방지, 탐지 및 대응합니다. 진화하는 보안 환경에 맞춰 위협 모델을 재검토하고 유지 관리합니다.

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

최신 보안 위협 파악: 적절한 제어를 정의하고 구현하는 데 도움이 되도록 최신 보안 위협 정보를 기반으로 공격 벡터를 파악합니다.

최신 보안 권장 사항 파악: 워크로드의 보안 상태를 개선할 수 있도록 AWS 및 업계 보안 권장 사항을 모두 확인합니다.

새로운 보안 서비스 및 기능을 정기적으로 평가 및 구현: 워크로드의 보안 상태를 개선할 수 있는 AWS 및 APN 파트너의 보안 서비스와 기능을 평가하고 구현합니다.

파이프라인에서 보안 제어의 테스트 및 검증 자동화: 빌드, 파이프라인, 프로세스에 포함되어 테스트되고 검증되는 보안 메커니즘에 대한 보안 기준 및 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.

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.

리소스

워크로드를 안전하게 운영하는 방법에 대해 자세히 알아보려면 다음 리소스를 참조하십시오.

동영상

- Well-Architected 방식의 보안 모범 사례
- 자동화 및 거버넌스를 통해 대규모 AWS 도입 지원
- AWS Security Hub: 보안 알림 관리 및 규정 준수 자동화

- AWS에서 보안 자동화

설명서

- 보안 프로세스 개요
- 보안 공지
- 보안 블로그
- AWS의 새로운 내용
- AWS 보안 감사 지침
- AWS에서 CI/CD 파이프라인 설정

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 리소스를 보관할, 경계가 뚜렷한 제로 트러스트 컨테이너입니다. 예를 들어 프로덕션 워크로드를 개발 및 테스트 워크로드에서 격리하려면 계정 수준의 분리를 적극 권장합니다.

계정을 사용하여 워크로드 구분: 워크로드가 증가함에 따라 조직에서 공통 가드레일을 설정할 수 있도록 보안 및 인프라를 염두에 두고 시작합니다. 그러면 여러 워크로드 사이에 경계를 정해 제어할 수 있습니다. 프로덕션 환경을 개발 및 테스트 환경과 격리하거나 외부 규정 준수 요구 사항(예: PCI-DSS 또는 HIPAA)에 정의된 대로 서로 다른 민감도 수준의 데이터를 처리하는 워크로드와 그렇지 않은 워크로드 간에 강력한 논리적 경계를 표시하려는 경우에는 계정 수준의 분리를 권장합니다.

AWS 계정 보호: AWS 계정을 보호하는 데는 루트 사용자를 사용하지 않고 보안을 확보하고, 연락처 정보를 최신 상태로 유지하는 등 여러 가지 측면이 있습니다. AWS Organizations를 사용하면 워크로드가 증가되고 이를 확장함에 따라 계정을 중앙에서 관리할 수 있습니다. AWS Organizations는 계정을 관리하고 제어를 설정하며 여러 계정에 걸쳐 서비스를 구성하는 데 도움을 줍니다.

중앙에서 계정 관리: AWS Organizations는 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 조직 단위(OU)로 그룹화하여 워크로드의 요구 사항과 용도에 따라 서로 다른 환경을 나타내도록 할 수 있습니다.

중앙에서 제어 설정: AWS 계정에 특정 서비스, 리전과 적절한 수준에서의 서비스 작업만 허용하면 AWS 계정의 능력 범위를 제어할 수 있습니다. AWS Organizations에서는 서비스 제어 정책(SCP)을 사용해 조직, 조직 산하 단위 또는 계정 수준에서 권한 가드레일을 적용함으로써 이를 모든 AWS Identity and Access Management(IAM) 사용자와 역할에 적용할 수 있습니다. 예를 들어 사용자가 명시적으로 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 서비스를 구성할 수 있습니다. 예를 들어 AWS CloudTrail을 사용하면 조직 전체에서 수행되는 모든 작업을 중앙에서 로깅하도록 구성하고 멤버 계정이 로깅을 비활성화하지 않도록 할 수 있습니다. 또한 정의한 규칙에 대해 AWS Config를 사용하여 중앙에서 데이터를 취합할 수 있으므로 워크로드를 감사하여 규정 준수 여부를 확인하고, 변화에 신속하게 반응하도록 지원할 수 있습니다. AWS CloudFormation StackSets를 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.

리소스

여러 AWS 계정을 배포하고 관리하기 위한 AWS 권장 사항에 대해 자세히 알아보려면 다음 리소스를 참조하십시오.

동영상

- AWS Organizations를 사용하여 다중 계정 AWS 환경 관리

- AXA: 글로벌 랜딩 존을 통한 도입 확장
- AWS Control Tower를 사용하여 다중 계정 AWS 환경 관리

설명서

- 모범 사례 AWS 환경 구축
- AWS Organizations
- AWS Control Tower
- AWS CloudFormation StackSets 사용
- AWS Organization의 여러 계정에서 서비스 제어 정책을 사용해 권한 가드레일을 설정하는 방법

실습

- 실습: AWS 계정 및 루트 사용자

Identity and Access Management

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. AWS는 인적 자격 증명과 시스템 자격 증명, 그리고 해당 권한을 관리하는 데 도움이 되는 다양한 기능을 제공합니다. 이러한 기능에 대한 모범 사례는 두 가지 주요 영역으로 나뉩니다.

- 자격 증명 관리
- 권한 관리

자격 증명 관리

안전한 AWS 워크로드 운영에 접근할 때 관리해야 하는 두 가지 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 하려면(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.

중앙 집중식 자격 증명 공급자 사용

인력 자격 증명의 경우 중앙 집중식 위치에서 자격 증명을 관리할 수 있는 자격 증명 공급자를 사용합니다. 이렇게 하면 단일 위치에서 액세스를 생성, 관리 및 취소하므로 여러 애플리케이션과 서비스에 대한 액세스를 더 쉽게 관리할 수 있습니다. 예를 들어 누군가가 퇴사한다면 한 곳에서 모든 애플리케이션 및 서비스(AWS 포함)에 대한 액세스 권한을 취소할 수 있습니다. 이렇게 하면 자격 증명을 여러 개 만들 필요가 없고, 기존 인사(HR) 프로세스와 통합할 수도 있습니다.

개인 AWS 계정과 연동하려면 AWS IAM의 SAML 2.0 기반 공급자를 통해 AWS용 중앙 집중식 자격 증명을 사용할 수 있습니다. SAML 2.0 프로토콜과 호환되기만 하면 AWS에서 직접 호스팅하든, AWS 외부에서 호스팅하든, AWS Partner Network(APN)에서 제공하든, 어느 공급자를 사용해도 됩니다. AWS 계정과 선택한 공급자 사이의 연동을 활용하면 SAML 어설션을 사용해 임시 보안 자격 증명을 얻어 사용자나 애플리케이션에 AWS API 작업을 호출할 액세스 권한을 부여할 수 있습니다. 웹 기반 SSO(Single Sign-On)도 지원되므로 사용자가 로그인 포털에서 AWS Management Console에 로그인할 수 있습니다.

AWS Organization 내 여러 계정에 연동하려면 AWS Single Sign-On(AWS SSO)에서 자격 증명 소스를 구성하여 사용자와 그룹이 저장될 위치를 지정하면 됩니다. 구성이 완료되면 자격 증명 공급자가 실제 소스가 되며, System for Cross-domain Identity Management(SCIM) v2.0 프로토콜을 사용해 정보를 동기화할 수 있습니다. 그러면 사용자나 그룹을 검색하여 이들에게 AWS 계정, 클라우드 애플리케이션 또는 둘 모두에 SSO 액세스 권한을 부여할 수 있습니다.

AWS SSO는 AWS Organizations와 통합되므로 자격 증명 공급자를 한 번 구성하면 조직에서 관리하는 기존 및 새 계정에 대한 액세스 권한을 부여할 수 있습니다. AWS SSO는 사용자 및 그룹을 관리하는 데 사용할 수 있는 기본 스토어를 제공합니다. AWS SSO 스토어를 사용하는 경우, 최소 권한의 모범 사례를 염두에 두고 사용자와 그룹을 생성하고 해당 액세스 수준을 AWS 계정 및 애플리케이션에 할당합니다. 또는 SAML 2.0을 사용하여 외부 자격 증명 공급자에 연결하거나 AWS Directory Services를 사용하여 Microsoft AD Directory에 연결할 수 있습니다. 구성이 완료되면 중앙 자격 증명 공급자를 통해 인증한 후 AWS Management Console, 명령줄 인터페이스 또는 AWS 모바일 앱에 로그인할 수 있습니다.

최종 사용자 또는 워크로드 소비자(예: 모바일 앱)를 관리할 때는 Amazon Cognito를 사용할 수 있습니다. 이는 웹 및 모바일 앱의 인증, 권한 부여 및 사용자 관리 등의 기능을 제공합니다. 사용자는 사용자 이름과 암호를 통해 직접 로그인하거나, Amazon, Apple, Facebook 또는 Google 등 타사를 통해 로그인할 수 있습니다.

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 SSO를 사용하여 사용자 그룹 및 속성을 관리할 수 있습니다. AWS SSO는 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.

강력한 로그인 메커니즘 사용

최소 암호 길이를 적용하고, 사용자에게 일반적인 암호나 재사용된 암호를 사용하지 않도록 교육합니다. 추가적인 확인 계층을 제공하기 위해 소프트웨어 또는 하드웨어 메커니즘을 사용하여 MFA(Multi-Factor Authentication)를 적용합니다. 예를 들어 AWS SSO를 자격 증명 소스로 사용하는 경우, MFA에 "컨텍스트 인식" 또는 "상시 작동" 설정을 구성하고 사용자가 자신의 MFA 디바이스를 등록하여 도입 속도를 향상할 수 있습니다. 외부 자격 증명 공급자(IdP)를 사용하는 경우 MFA에 IdP를 구성합니다.

임시 자격 증명 사용

임시 자격 증명을 동적으로 획득하려면 자격 증명(ID)이 필요합니다. 인력 자격 증명의 경우 AWS SSO를 사용하거나 IAM과의 연동을 사용하여 AWS 계정에 액세스합니다. 시스템 자격 증명(예: EC2 인스턴스 또는 Lambda 함수)의 경우 장기적인 액세스 키를 포함한 IAM 사용자 대신 IAM 역할을 사용해야 합니다.

AWS Management Console을 사용하는 인적 자격 증명의 경우, 사용자가 임시 자격 증명을 획득하여 AWS에 연동해야 합니다. 이렇게 하려면 AWS SSO 사용자 포털을 사용하거나 IAM과의 연동을 구성하면 됩니다. CLI 액세스가 필요한 사용자의 경우, AWS Single Sign-On(AWS SSO)과의 직접적인 통합을 지원하는 AWS CLI v2를 사용해야 합니다. 사용자는 AWS SSO 계정과 역할에 연결된 CLI 프로필을 생성할 수 있습니다. 그러면 CLI가 자동으로 AWS SSO에서 AWS 자격 증명을 검색하여 사용자를 대신해 새로 고칩니다. 따라서 AWS SSO 콘솔에서 임시 AWS 자격 증명을 복사해 붙여넣지 않아도 됩니다. SDK의 경우, 사용자는 AWS STS를 사용하여 임시

자격 증명을 수신할 역할을 수임해야 합니다. 경우에 따라 임시 자격 증명이 실용적이지 않을 수 있습니다. 액세스 키를 저장하는 데 수반되는 위험을 알고 있어야 하고, 키를 자주 교체해 사용해야 하며, 가능한 경우 MFA를 필수 조건으로 지정해야 합니다.

소비자에게 AWS 리소스에 대한 액세스 권한을 부여해야 하는 경우, Amazon Cognito 자격 증명 풀을 사용해 임시적이고 권한이 제한된 일련의 자격 증명을 할당하여 AWS 리소스에 액세스합니다. 각 사용자에 대한 권한은 생성한 IAM 역할을 통해 제어됩니다. 사용자의 ID 토큰에 있는 클레임에 따라 각 사용자의 역할을 선택하는 규칙을 정의할 수 있습니다. 인증된 사용자에 대한 기본 역할을 정의할 수 있습니다. 또한 인증되지 않은 게스트 사용자의 경우, 권한이 제한된 별도의 IAM 역할을 정의할 수 있습니다.

시스템 자격 증명의 경우, IAM 역할을 사용하여 AWS에 대한 액세스 권한을 부여해야 합니다. EC2 인스턴스의 경우, Amazon EC2 역할을 사용할 수 있습니다. EC2 인스턴스에 IAM 역할을 연결하면 Amazon EC2에서 실행 중인 애플리케이션에서 AWS가 생성, 배포 및 교체하는 임시 보안 자격 증명을 자동으로 사용하도록 할 수 있습니다. 키 또는 암호를 사용하여 EC2 인스턴스에 액세스하는 경우, AWS Systems Manager를 사용하면 저장된 보안 암호 없이 사전 설치된 에이전트를 통해 보다 안전하게 인스턴스에 액세스하고 이를 관리할 수 있습니다. 또한 AWS Lambda와 같은 다른 AWS 서비스를 사용하여 IAM 서비스 역할을 구성하고, 이를 통해 임시 자격 증명을 사용해 AWS 작업을 수행하도록 서비스 권한을 부여할 수 있습니다.

정기적으로 자격 증명 감사 및 교체

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 자격 증명을 사용하도록 규정해야 합니다. 또한 자격 증명 공급자의 MFA 설정을 지속적으로 모니터링하는 것이 좋습니다. AWS Config 규칙을 설정하여 이러한 설정을 모니터링할 수 있습니다. 시스템 자격 증명의 경우, IAM 역할을 사용한 임시 자격 증명을 사용해야 합니다. 이러한 방법이 불가능한 경우, 액세스 키를 자주 감사하고 교체해 사용해야 합니다.

안전하게 보안 암호 저장 및 사용

IAM과 관련이 없는 자격 증명의 경우(예: 데이터베이스 로그인), AWS Secrets Manager와 같이 보안 암호 관리 작업을 처리하도록 설계된 서비스를 사용합니다. AWS Secrets Manager를 사용하면 지원 서비스를 통해 간편하게 암호화된 보안 암호를 관리 및 교체하고 안전하게 저장할 수 있습니다. 보안 암호에 액세스하기 위한 호출은 감사를 위해 CloudTrail에 기록되며, IAM 권한은 이에 액세스하기 위한 최소 권한을 부여할 수 있습니다.

리소스

AWS 자격 증명 보호와 관련된 AWS 모범 사례에 대해 자세히 알아보려면 다음 리소스를 참조하십시오.

동영상

- Mastering identity at every layer of the cake
- Managing user permissions at scale with AWS SSO
- Best Practices for Managing, Retrieving, & Rotating Secrets at Scale

설명서

- AWS 계정 루트 사용자
- AWS 계정 루트 사용자 자격 증명과 IAM 사용자 자격 증명
- IAM 모범 사례
- IAM 사용자의 계정 암호 정책 설정
- AWS Secrets Manager 시작하기
- 인스턴스 프로파일 사용
- 임시 보안 자격 증명
- 자격 증명 공급자 및 연동

권한 관리

AWS 및 워크로드에 액세스해야 하는 사람 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 true여야 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.

조직에 대한 권한 가드레일 정의

AWS에서 추가 워크로드를 확장하고 관리하면서, 계정을 사용하여 이러한 워크로드를 구분하고 AWS Organizations를 사용하여 해당 계정을 관리해야 합니다. 이 경우 조직 내 모든 자격 증명에 대한 액세스를 제한하는 공용 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.

AWS Organizations를 사용하여 계정을 그룹화하고 각 계정 그룹에 대한 공통 제어를 설정할 수 있습니다. 이러한 공통 제어를 설정하려면 AWS Organizations와 통합된 서비스를 사용할 수 있습니다. 즉, 계정 그룹에 대한 액세스를 제한하려면 서비스 제어 정책(SCP)을 사용하면 됩니다. 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 예외 사항을 정의할 수 있습니다. 예를 들어 주어진 계정에서 특정 관리자 역할을 제외한 모든 IAM 엔터티에 대해 서비스 작업을 제한할 수 있습니다.

최소 권한 액세스 부여

최소 권한 원칙을 수립하면 사용 가능성과 효율성을 적절하게 절충하면서 자격 증명이 특정 작업을 처리하는 데 필요한 최소한의 기능 세트만 수행하도록 할 수 있습니다. 이 원칙에 따라

운영하면 의도치 않은 액세스를 제한하고, 누가 어떤 리소스에 액세스할 수 있는지 감사할 수 있습니다. AWS에서 자격 증명에는 기본적으로 권한이 없습니다. 다만 루트 사용자는 예외로, 일부 특정 작업에만 사용해야 합니다.

IAM 또는 리소스 엔터티(예: 연동된 자격 증명, 시스템 또는 리소스(예: S3 버킷)에서 사용하는 IAM 역할)에 연결된 권한을 명시적으로 부여하는 경우 정책을 사용합니다. 정책을 생성하여 연결할 때 서비스 작업, 리소스는 물론 AWS가 액세스를 허용하려면 true여야 하는 조건을 지정할 수 있습니다. AWS는 액세스 범위를 좁히는 데 도움이 되는 다양한 조건을 지원합니다. 예를 들어 `PrincipalOrgID`조건 키를 사용하면 AWS Organizations의 식별자를 확인하므로 AWS Organization 내에서 액세스 권한을 부여할 수 있습니다. 또한 `CalledVia` 조건 키를 사용하여 AWS CloudFormation에서 AWS Lambda 함수를 생성하는 등 AWS 서비스가 사용자를 대신하여 수행하는 요청을 제어할 수도 있습니다. 이렇게 하면 AWS 전체에서 인적 및 시스템 자격 증명에 대해 세분화된 권한을 설정할 수 있습니다.

AWS에는 권한 관리를 확장하고 최소 권한을 준수할 수 있는 기능도 있습니다.

권한 경계: 권한 경계를 사용하여 관리자가 설정할 수 있는 최대 권한을 설정할 수 있습니다. 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): AWS에서는 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.

퍼블릭 및 교차 계정 액세스 분석

AWS에서는 다른 계정의 리소스에 대한 액세스 권한을 부여할 수 있습니다. 리소스에 연결된 정책(예: S3 버킷 정책)을 사용하거나 자격 증명이 다른 계정의 IAM 역할을 수임하도록 허용하여

교차 계정 액세스 권한을 부여합니다. 리소스 정책을 사용할 때는 조직의 자격 증명에 대한 액세스 권한을 부여하고, 리소스 공개 시점을 의도적으로 설정할 수 있어야 합니다. 리소스를 공개 리소스로 설정하면 모든 사람이 해당 리소스에 액세스할 수 있게 되므로 가급적 지양하는 것이 좋습니다. IAM Access Analyzer는 수학적 방식(즉, 증명 가능한 보안)을 사용해 자격 증명에 계정 외부의 리소스에 대한 전체적인 액세스 경로를 제공합니다. 따라서 리소스 정책을 지속적으로 검토하고, 공개 또는 교차 계정 액세스 결과를 보고하여 잠재적인 광범위한 액세스를 쉽게 분석할 수 있습니다.

안전하게 리소스 공유

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 조직 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.

지속적으로 권한 축소

팀과 프로젝트를 막 시작하는 경우, 혁신과 민첩성을 이끌어내기 위해 광범위한 액세스 권한을 부여하고자 하는 경우가 있습니다. 액세스를 지속적으로 평가하여 필요한 권한으로만 액세스를 제한하고 최소 권한을 달성하는 것이 좋습니다. AWS는 미사용 액세스를 파악하는 데 도움이 되는 액세스 분석 기능을 제공합니다. AWS에서 액세스 활동을 분석하여 액세스 키와 역할이 마지막으로 사용된 사례의 정보를 제공하므로 미사용된 사용자와 역할을 파악하는 데 도움이 됩니다. 즉 마지막으로 액세스한 타임스탬프를 사용하면 미사용된 사용자와 역할을 파악하여 제거할 수 있습니다. 또한 서비스와 작업의 마지막 액세스 정보를 검토하면 특정 사용자와 역할을 대상으로 권한을 강화할 수도 있습니다. 예를 들어 마지막 액세스 정보를 사용하면 애플리케이션 역할에 필요한 특정 S3 작업을 파악하여 그러한 작업에 대해서만 액세스를 제한할 수 있습니다. 이러한 기능은 콘솔에서 프로그램 방식으로 제공되므로 인프라 워크플로 및 자동화된 도구에 손쉽게 통합할 수 있습니다.

긴급 액세스 프로세스 설정

실제로 일어날 가능성은 희박하지만 자동화된 프로세스나 파이프라인에 문제가 발생할 경우에 대비하여 워크로드, 특히 AWS 계정에 대한 긴급 액세스를 허용하는 프로세스를 마련하는 것이 좋습니다. 이 프로세스에는 액세스를 위한 긴급 AWS 교차 계정 역할 또는 관리자가 긴급 요청을 검증하고 승인하기 위해 따라야 하는 특정 프로세스 등 다양한 기능의 조합이 포함될 수 있습니다.

리소스

세분화된 권한 부여와 관련된 최신 AWS 모범 사례에 대해 자세히 알아보려면 다음 리소스를 참조하십시오.

동영상

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

설명서

- 최소 권한 부여
- 정책 사용
- IAM 사용자, 그룹 및 자격 증명을 관리하기 위한 권한 위임
- IAM Access Analyzer
- 불필요한 자격 증명 삭제
- MFA를 사용하여 CLI에서 역할 배정
- 권한 경계
- 속성 기반 액세스 제어(ABAC)

실습

- 실습: 역할 생성을 위임하는 IAM 권한 경계
- 실습: EC2에 대한 IAM 태그 기반 액세스 제어
- 실습: Lambda 교체 계정 IAM 역할 위임

탐지

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.

- 구성
- 조사

구성

서비스 및 애플리케이션 로깅 구성: 기초적인 방법은 계정 수준에서 탐지 메커니즘 세트를 설정하는 것입니다. 이 기본 메커니즘 세트는 계정의 모든 리소스에서 광범위한 작업을 기록하고 탐지하는 것을 목표로 합니다. 이를 통해 자동화된 수정, 기능 추가를 위한 파트너 통합 등의 옵션이 포함된 포괄적인 탐지 기능을 구축할 수 있습니다.

AWS에서 이 기본 세트의 서비스는 다음과 같습니다.

- AWS CloudTrail은 AWS Management Console, AWS SDK, 명령줄 도구 및 기타 AWS 서비스를 통해 수행된 작업을 비롯하여 AWS 계정 활동의 이벤트 이력을 제공합니다.
- AWS Config의 경우 AWS 리소스 구성을 모니터링 및 기록하며, 원하는 구성을 기준으로 자동으로 평가하고 수정할 수 있습니다.
- Amazon GuardDuty는 악성 활동 및 무단 행위를 지속적으로 모니터링하여 AWS 계정 및 워크로드를 보호하는 위협 탐지 서비스입니다.

- AWS Security Hub는 여러 AWS 서비스 및 타사 제품(선택 사항)의 보안 알림 또는 탐지 결과를 집계하고 정리하고 우선순위를 지정함으로써 보안 알림 및 규정 준수 상태를 종합적으로 파악할 수 있는 단일 장소를 제공합니다.

Amazon Virtual Private Cloud(VPC)를 위시한 대다수의 주요 AWS 서비스는 근본적으로 계정 수준을 기반으로 구축되어 서비스 수준 로깅 기능을 제공합니다. VPC 흐름 로그를 사용하면 네트워크 인터페이스에서 송수신되는 IP 트래픽에 대한 정보를 캡처할 수 있습니다. 이러한 정보는 연결 기록에 대한 중요한 통찰력을 제공하고 변칙적 동작에 대한 자동화된 작업을 트리거할 수 있습니다.

EC2 인스턴스와 AWS 서비스에서 시작되지 않은 애플리케이션 기반 로깅의 경우, Amazon CloudWatch Logs를 사용하여 로그를 저장하고 분석할 수 있습니다. 에이전트는 실행 중인 운영 체제 및 애플리케이션에서 로그를 수집하여 자동으로 저장합니다. CloudWatch Logs에서 로그를 사용할 수 있게 되면 실시간으로 처리하거나, 분석 정보를 사용해 바로 분석 작업을 진행할 수 있습니다.

로그 수집과 집계 작업도 중요하지만, 복잡한 아키텍처에서 생성되는 대량의 로그 및 이벤트 데이터에서 의미 있는 분석 정보를 추출하는 기능도 중요합니다. 자세한 내용은 안정성 부문 백서의 모니터링 섹션을 참조하십시오. 로그 자체에 민감한 것으로 간주되는 데이터가 포함될 수 있습니다. CloudWatch Logs 에이전트가 캡처하는 로그 파일로 가는 경로를 애플리케이션 데이터가 잘못 찾은 경우 또는 로그 병합을 위해 교차 리전 로깅이 구성되어 있는데 경계를 넘어 특정 종류의 정보를 전송하는 데 대한 법률적 고려 사항이 있는 경우입니다.

한 가지 방식은 로그가 전송될 때 이벤트에서 트리거되는 Lambda 함수를 사용하여 S3 버킷과 같은 중앙 로깅 위치로 전달하기 전에 로그 데이터를 필터링하고 교정하는 것입니다. 수정되지 않은 로그는 "합리적인 시간"(규정 및 법무팀에서 결정함)이 경과할 때까지 로컬 버킷에 보존될 수 있으며, 해당 시점에 도달하면 S3 수명 주기 규칙이 자동으로 로그를 삭제할 수 있습니다. 또한 S3 Object Lock을 사용하여 Amazon S3에서 로그를 추가적으로 보호할 수도 있는데, 이 경우 WORM(Write-Once-Read-Many) 모델을 사용해 객체를 저장할 수 있습니다.

로그, 결과, 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/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/협업 및 개발자 생산성 플랫폼에 통합하고 있습니다. 자동화에 착수하는 조직의 경우, API 기반의 지연 시간이 짧은 티켓팅 시스템은 "먼저 자동화할 대상"을 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 IAM 및 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.

GuardDuty 및 Security Hub는 다른 AWS 서비스를 통해서도 사용할 수 있는 로그 레코드에 대한 집계, 중복 제거, 분석 메커니즘을 제공합니다. 특히 GuardDuty는 VPC DNS 서비스의 정보와 CloudTrail 및 VPC 흐름 로그를 통해 볼 수 있는 정보를 수집, 집계 및 분석합니다. Security Hub는 GuardDuty, AWS Config, Amazon Inspector, Macie, AWS Firewall Manager 그리고 AWS Marketplace에서 사용할 수 있는 많은 타사 보안 제품의 출력은 물론 알맞게 작성된 경우 자체 코드도 수집, 집계 및 분석할 수 있습니다. GuardDuty 및 Security Hub 모두 여러 계정의 결과와 분석 정보를 집계할 수 있는 마스터-멤버 모델을 보유하고 있으며, Security Hub는 온프레미스 SIEM을 AWS 측 로그 및 알림 프리프로세서와 어그리게이터로 사용하는 고객들이 주로 사용합니다. 이러한 고객들은 Lambda 기반 프로세서와 전달자를 통해 Amazon EventBridge에 수집할 수 있습니다.

리소스

로그 캡처 및 분석과 관련된 최신 AWS 권장 사항에 대해 자세히 알아보려면 다음 리소스를 참조하십시오.

동영상

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

설명서

- Amazon GuardDuty 설정
- AWS Security Hub
- 시작하기: Amazon CloudWatch Logs
- Amazon EventBridge
- CloudTrail 로그를 분석하도록 Athena 구성
- Amazon CloudWatch
- AWS Config
- CloudTrail의 추적 생성
- 로깅 솔루션 중앙 집중화

실습

- 실습: Security Hub 활성화
- 실습: 탐지 제어 자동 배포
- 실습: Amazon GuardDuty 실습

조사

실행 가능한 보안 이벤트 구현: 보유한 각 탐지 메커니즘에 대해 런북 또는 플레이북 형태의 조사 프로세스도 있어야 합니다. 예를 들어 Amazon GuardDuty를 활성화하면 서로 다른 결과가 생성됩니다. 각 결과 유형에 대한 런북 항목이 있어야 합니다. 예를 들어 트로이 목마를 발견한 경우에는 누군가에게 조사 및 수정을 지시하는 간단한 지침이 런북 안에 있습니다.

이벤트에 대한 응답 자동화: AWS에서는 Amazon EventBridge를 사용하여 관심 있는 이벤트와 자동화된 워크플로에 대해 예기치 않은 잠재적 변경 사항에 대한 정보를 조사할 수 있습니다. 이

서비스는 CloudTrail 이벤트 등의 기본 AWS 이벤트 형식과 애플리케이션에서 생성 가능한 사용자 지정 이벤트를 중개하도록 설계된 확장 가능 규칙 엔진을 제공합니다. 또한 Amazon EventBridge를 사용하면 인시던트 대응 시스템(Step Functions)을 구축하는 워크플로 시스템 또는 중앙 보안 계정에 이벤트를 라우팅하거나, 추가 분석을 위해 버킷에 이벤트를 라우팅할 수 있습니다.

AWS Config 규칙을 사용하여 변경 사항을 감지하고 이 정보를 올바른 워크플로로 라우팅할 수도 있습니다. AWS Config는 범위 내 서비스(지연 시간은 Amazon EventBridge보다 더 김)의 변경 사항을 감지한 다음, 롤백, 규정 준수 정책 적용, 변경 관리 플랫폼 및 운영 티켓팅 시스템 등으로 정보 전달을 위해 AWS Config 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 규칙 라이브러리를 활용할 수도 있습니다.

리소스

알림 및 워크플로와 감사 제어를 통합하는 방법과 관련된 최신 AWS 모범 사례에 대해 자세히 알아보려면 다음 리소스를 참조하십시오.

동영상

- Amazon Detective
- Remediating Amazon GuardDuty and AWS Security Hub Findings
- Best Practices for Managing Security Operations on AWS
- Achieving Continuous Compliance using AWS Config

설명서

- Amazon Detective
- Amazon EventBridge
- AWS Config 규칙
- AWS Config 규칙 리포지토리(오픈 소스)
- AWS Config 규칙 개발 키트

실습

- 솔루션: AWS 계정 활동에 대한 실시간 인사이트
- 솔루션: 중앙 집중식 로깅

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.

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

- 네트워크 보호
- 컴퓨팅 보호

네트워크 보호

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. 서버리스 보안에 대한 구체적인 지침은 AWS Well-Architected 서버리스 애플리케이션 렌즈를 참조하십시오.

네트워크 계층 생성: 연결성 요구 사항을 공유하는 EC2 인스턴스, RDS 데이터베이스 클러스터, Lambda 함수와 같은 구성 요소를 서브넷으로 구성된 계층으로 분할할 수 있습니다. 예를 들어 인터넷에 액세스할 필요가 없는 VPC의 RDS 데이터베이스 클러스터는 인터넷과 연결되는 경로가 없는 서브넷에 배치해야 합니다. 이러한 계층적 제어 방식은 의도하지 않은 액세스를 허용할 수 있는 단일 계층 구성 오류로 인한 영향을 완화합니다. AWS Lambda의 경우, VPC에서 함수를 실행하여 VPC 기반 제어를 진행할 수 있습니다.

수천 개의 VPC, AWS 계정, 온프레미스 네트워크를 포함할 수 있는 네트워크 연결의 경우 AWS Transit Gateway를 사용해야 합니다. AWS Transit Gateway는 스포크처럼 작동하는 모든 연결된 네트워크 간에 트래픽이 라우팅되는 방식을 제어하는 허브 역할을 합니다. Amazon VPC와 AWS Transit Gateway 간의 트래픽은 AWS 프라이빗 네트워크에 유지되므로 DDoS(Distributed Denial of Service) 공격과 같은 외부 위협 벡터 그리고 SQL 주입, 교차 사이트 스크립팅, 교차 사이트 요청 위조, 손상된 인증 코드 남용과 같은 일반적인 악용을 줄입니다. 또한 AWS Transit Gateway 리전 간 피어링은 단일 장애 지점이나 대역폭 병목 없이 리전 간 트래픽을 암호화합니다.

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, 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. 비 TCP 프로토콜을 사용하지 않는 한, 로드 밸런서 또는 CloudFront 없이 인터넷에서 직접 EC2 인스턴스에 액세스할 필요가 없습니다(보안 그룹에 의해 제한된 포트를 사용하는 경우에도). 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 API를 호출하려면 인터넷에 액세스하여 API를 호출하는 구성 요소가 필요하지만(AWS API 엔드포인트 위치) 다른 AWS 서비스는 VPC 안에 있는 엔드포인트를 사용합니다. Amazon S3 및 DynamoDB를 비롯한 여러 AWS 서비스가 VPC 엔드포인트를 지원하며, 이 기술은 AWS PrivateLink에서 일반화되었습니다. 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.

검사 및 보호 구현: 각 계층에서 트래픽을 검사하고 필터링합니다. HTTP 기반 프로토콜을 통해 트랜잭션되는 구성 요소의 경우, 웹 애플리케이션 방화벽이 일반 공격으로부터 보호하는 데 도움을 줄 수 있습니다. AWS WAF는 Amazon API Gateway API, Amazon CloudFront, Application Load Balancer로 전달되는 구성 가능한 규칙과 일치하는 HTTP(s) 요청을 모니터링하고 차단할 수 있는 웹 애플리케이션 방화벽입니다. AWS WAF를 시작하려면 자체 AWS 관리형 규칙을 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. AWS Shield Advanced나 웹 애플리케이션에 대한 원치 않는 요청을 자동으로 차단할 수 있는 솔루션을 사용하여 공격에 신속하게 대응할 수도 있습니다.

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.

리소스

네트워크 보호와 관련된 AWS 모범 사례에 대해 자세히 알아보려면 다음 리소스를 참조하십시오.

동영상

- AWS Transit Gateway reference architectures for many VPCs
- Application Acceleration and Protection with Amazon CloudFront, AWS WAF, and AWS Shield
- DDoS Attack Detection at Scale

설명서

- Amazon VPC 설명서
- AWS WAF 시작하기
- 네트워크 액세스 제어 목록
- VPC의 보안 그룹
- VPC에 권장되는 네트워크 ACL 규칙
- AWS Firewall Manager
- AWS PrivateLink
- VPC 엔드포인트
- Amazon Inspector

실습

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

컴퓨팅 보호

취약성 관리 수행: 코드, 종속 관계, 인프라의 취약성을 자주 검색하고 패치하여 새 위협으로부터 워크로드를 보호합니다.

빌드 및 배포 파이프라인을 사용하여 취약성 관리의 여러 부분을 자동화할 수 있습니다.

- 타사 정적 코드 분석 도구를 사용하여 확인되지 않은 함수 입력 범위와 최신 CVE와 같은 일반적인 보안 문제를 식별합니다. 지원되는 언어에 대해 Amazon CodeGuru를 사용할 수 있습니다.
- 타사 종속성 확인 도구를 사용하여 코드가 링크된 라이브러리가 최신 버전인지, 해당 라이브러리에 CVE가 없는지, 소프트웨어 정책 요구 사항에 부합하는 라이선스 조건이 있는지를 확인합니다.
- Amazon Inspector를 사용하면 인스턴스에 대해 알려진 일반 취약성 및 노출(CVE)을 확인하는 구성 평가를 수행하고, 보안 벤치마크를 기준으로 평가하고, 결함 알림을 완전히 자동화할 수 있습니다. 프로덕션 인스턴스 또는 빌드 파이프라인에서 실행되는 Amazon Inspector는 확인된 정보가 있으면 개발자와 엔지니어에게 알림을 보냅니다. 프로그래밍 방식으로 확인된 정보에 액세스할 수 있으며, 팀에게 백로그 및 버그 추적 시스템 액세스 권한을 제공할 수 있습니다. EC2 Image Builder는 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 Scanning을 구현하여 컨테이너에서 CVE를 찾습니다.
- Amazon Inspector 및 기타 도구는 존재하는 구성 및 CVE를 식별하는 데 효과적이지만, 애플리케이션 수준에서 워크로드를 테스트하려면 다른 방법이 필요합니다. Fuzzing은 자동화를 사용하여 잘못된 형식의 데이터를 입력 필드 및 애플리케이션의 기타 영역에 주입함으로써 버그를 찾는 유명한 방법입니다.

이러한 기능 중 상당수는 AWS 서비스, AWS Marketplace에 있는 제품 또는 오픈 소스 도구를 사용하여 수행할 수 있습니다.

공격 대상 영역 감소: 운영 체제를 강화하고 사용 중인 구성 요소, 라이브러리 및 외부 사용 서비스를 최소화하여 공격 대상 영역을 줄입니다. 공격 대상 영역을 줄이려면 발생할 수 있는 진입점과 잠재적 위협을 식별하는 위협 모델이 필요합니다. 공격 대상 영역을 줄이는 일반적인 방법은 운영 체제 패키지이든, 또는 애플리케이션이나(EC2 기반 워크로드의 경우) 코드의 외부 소프트웨어 모듈이든(모든 워크로드의 경우) 사용하지 않는 구성 요소를 줄이는 것입니다. 일반적인 운영 체제 및 서버 소프트웨어에 대한 여러 가지 강화 및 보안 구성 가이드가

있습니다(예: Center for Internet Security에서 제공하는 자료). 이러한 가이드를 출발점으로 사용하고 반복해서 활용할 수 있습니다.

사람이 한 발 떨어져서 작업하도록 지원: 대화형 액세스 기능을 제거하면 인적 오류의 위험과 수동으로 구성하거나 관리해야 할 가능성이 줄어듭니다. 예를 들어 변경 관리 워크플로를 사용하여 직접 액세스를 허용하는 대신 AWS Systems Manager와 같은 도구를 통해 또는 배스천 호스트를 통해 EC2 인스턴스를 관리할 수 있습니다. AWS Systems Manager는 자동화 워크플로, 문서(플레이북) 및 실행 명령 등의 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.

관리형 서비스 구현: 공유 책임 모델의 일환으로 보안 유지 관리 작업을 줄일 수 있도록 Amazon RDS, AWS Lambda, 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 프레임워크에 설명된 다른 방법으로 애플리케이션을 보호하는 데 더 많은 시간을 할애할 수 있습니다. AWS Lambda를 사용하면 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있으므로 인프라나 운영 체제가 아니라 코드 수준의 연결, 호출, 보안에만 집중하면 됩니다.

소프트웨어 무결성 검증: 워크로드에 사용되는 소프트웨어, 코드, 라이브러리가 신뢰할 수 있는 소스에서 온 것이며 변조되지 않았는지를 검증하기 위한 메커니즘(예: 코드 서명)을 구현합니다. 예를 들어 바이너리 및 스크립트의 코드 서명 인증서를 검토하여 작성자를 확인하고 작성자가 생성한 후에 변조되지 않았는지 확인해야 합니다. 또한 다운로드한 소프트웨어의 체크섬을 공급자의 체크섬과 비교하면 변조되지 않았는지를 확인하는 데 도움이 될 수 있습니다.

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

리소스

컴퓨팅 보호와 관련된 AWS 모범 사례에 대해 자세히 알아보려면 다음 리소스를 참조하십시오.

동영상

- Security best practices for the Amazon EC2 instance metadata service
- Securing Your Block Storage on AWS
- Securing Serverless and Container Services
- Running high-security workloads on Amazon EKS
- Architecting Security through Policy Guardrails in Amazon EKS

설명서

- AWS Lambda 보안 개요
- Amazon EC2의 보안
- AWS Systems Manager
- Amazon Inspector
- Writing your own AWS Systems Manager documents
- 배스천 호스트를 Amazon EC2 Systems Manager로 대체

실습

- 실습: EC2 웹 애플리케이션 자동 배포

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.

- 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.

워크로드 안에서 데이터 식별: 워크로드에서 처리 중인 데이터의 유형 및 분류, 관련 비즈니스 프로세스, 데이터 소유자, 적용 가능한 법률 및 규정 준수 요구 사항, 저장 위치, 적용해야 할 최종 제어를 이해해야 합니다. 여기에는 데이터가 공개적으로 사용 가능한지, 데이터가 PII(고객 개인 식별 정보)처럼 내부에서만 사용되는지, 지적 재산처럼 데이터에 대한 액세스가 더 엄격히 제한되는지, 법적으로 권한이 있는지, 민감한 데이터로 표시되는지 등을 나타내는 분류가 포함될 수 있습니다. 적절한 데이터 분류 시스템과 각 워크로드의 보호 요구 사항 수준을 철저하게 관리하면 데이터에 적합한 제어 기능과 액세스/보호 수준을 적용할 수 있습니다. 예를 들어 공개 콘텐츠는 누구나 액세스할 수 있지만 중요한 콘텐츠는 암호화하여 보호된 방식(콘텐츠 암호를 해독하려면 키에 대한 액세스 권한이 부여되어야 함)으로 저장할 수 있습니다.

데이터 보호 제어 정의: 리소스 태그, 중요도별(주의/영역/커뮤니티별로도 가능) 개별 AWS 계정, IAM 정책, Organizations SCP, AWS KMS, AWS CloudHSM을 사용함으로써 데이터 분류 및 암호화를 통한 보호를 위한 정책을 정의하고 구현할 수 있습니다. 예를 들어 기밀 데이터를

처리하는 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.

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.

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.

리소스

데이터 분류에 대해 자세히 알아보려면 다음 리소스를 참조하십시오.

설명서

- 데이터 분류 백서
- Amazon EC2 리소스 태그 지정
- Amazon S3 객체 태그 지정

저장된 데이터 보호

저장된 데이터란 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 둘 모두 중요한 데이터 보호 체계이나 별개의 개념입니다.

*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.

보안 키 관리 구현: 키의 저장, 교체, 액세스 제어를 포함하는 암호화 방식을 정의함으로써 권한이 없는 사용자로부터 콘텐츠를 보호하고 권한이 있는 사용자에게도 불필요한 콘텐츠 노출이 발생하는 것을 방지할 수 있습니다. AWS KMS를 사용하면 암호화 키를 관리하고 다양한 AWS 서비스와 통합할 수 있습니다. 이 서비스는 마스터 키 용도로 내구성과 보안성이 뛰어난 중복 스토리지를 제공합니다. 키 별칭과 키 수준 정책을 정의할 수 있습니다. 정책을 사용하면 키 관리자와 키 사용자를 정의할 수 있습니다. 또한 AWS CloudHSM은 AWS 클라우드에서 고유한

암호화 키를 쉽게 생성하여 사용할 수 있는 클라우드 기반 하드웨어 보안 모듈(HSM)입니다. HSM을 사용하면 FIPS 140-2 수준 3 확인된 HSM을 통해 데이터 보안 관련 회사, 계약 및 규정 준수 요구 사항을 충족할 수 있습니다.

저장 시 암호화 적용: 데이터를 저장할 때 반드시 암호화를 사용하도록 해야 합니다. AWS KMS는 여러 AWS 서비스와 원활하게 통합되므로 모든 저장된 데이터를 쉽게 암호화할 수 있습니다. 예를 들어 Amazon S3의 경우 버킷에 기본 암호화를 설정하여 새 객체가 모두 자동으로 암호화되도록 하면 됩니다. 또한 Amazon EC2는 전체 리전에 대해 기본 암호화 옵션을 설정함으로써 암호화를 적용하는 것을 지원합니다.

액세스 제어 적용: 액세스(최소 권한 사용), 백업(안정성 백서 참조), 격리, 버전 관리 등의 다양한 제어는 저장된 데이터를 보호하는 데 모두 도움이 될 수 있습니다. 데이터에 대한 액세스 권한은 CloudTrail 등 본 백서의 앞부분에서 다룬 탐지 메커니즘과 S3 액세스 로그와 같은 서비스 수준 로그를 사용하여 감사해야 합니다. 공개적으로 액세스할 수 있는 데이터의 인벤토리를 만들고, 사용 가능한 데이터의 양을 점차 줄이는 방법을 계획해야 합니다. Amazon S3 Glacier 저장소 잠금 및 S3 객체 잠금은 필수 액세스 제어를 제공하는 기능입니다. 규정 준수 옵션으로 저장소 정책을 잠그면 루트 사용자도 잠금이 만료되기 전까지는 변경할 수 없습니다. 이 메커니즘은 SEC, CFTC 및 FINRA의 장부 및 기록 관리 요구 사항에 부합합니다. 자세한 내용은 이 백서를 참조하십시오.

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.

데이터에 접근하지 못하도록 하는 메커니즘 사용: 정상적인 운영 환경에서 모든 사용자가 민감한 데이터와 시스템에 직접 액세스하지 못하도록 합니다. 예를 들어 변경 관리 워크플로를 사용하여 직접 액세스를 허용하는 대신 도구나 배스천 호스트를 사용하여 EC2 인스턴스를 관리할 수 있습니다. 이렇게 하려면 AWS Systems Manager Automation을 사용할 수 있으며, 이 경우 작업을 수행할 때 사용하는 단계를 포함한 자동화 문서를 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 있습니다. CI/CD 파이프라인이 사용되지 않는 경우,

정상적으로 비활성화된 브레이크-글라스 액세스 메커니즘을 적절하게 제공하기 위해 필요한 제어 및 프로세스를 결정합니다.

저장된 데이터 보호 자동화: 자동화된 도구를 사용하여 저장된 데이터 제어를 지속적으로 검증하고 적용합니다(예: 암호화된 스토리지 리소스만 있는지 확인함). AWS Config 규칙을 사용하면 모든 EBS 볼륨이 암호화되었는지 검증을 자동화할 수 있습니다. 또한 AWS Security Hub를 사용하면 보안 표준을 기준으로 자동화된 검사를 통해 여러 가지 제어의 유효성을 확인할 수 있습니다. 또한 AWS Config 규칙은 자동으로 규정 미준수 리소스를 수정할 수 있습니다.

리소스

저장된 데이터 보호와 관련된 AWS 모범 사례에 대해 자세히 알아보려면 다음 리소스를 참조하십시오.

동영상

- How Encryption Works in AWS
- Securing Your Block Storage on AWS
- Achieving security goals with AWS CloudHSM
- Best Practices for Implementing AWS Key Management Service
- A Deep Dive into AWS Encryption Services

설명서

- 암호화를 사용하여 Amazon S3 데이터 보호
- Amazon EBS 암호화
- Amazon RDS 리소스 암호화
- 암호화를 사용하여 데이터 보호
- AWS 서비스에서 AWS KMS를 사용하는 방법
- Amazon EBS 암호화
- AWS Key Management Service
- AWS CloudHSM

- AWS KMS 암호화 세부 정보 백서
- AWS KMS에서 키 정책 사용
- 버킷 정책 및 사용자 정책 사용
- AWS 암호화 도구

전송 중 데이터 보호

전송 중 데이터는 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.

보안 키 및 인증서 관리 구현: 암호화 키와 인증서를 안전하게 저장하고 엄격하게 액세스 제어를 통해 적절한 간격으로 교체합니다. 이를 위한 가장 좋은 방법은 AWS Certificate Manager(ACM)와 같은 관리형 서비스를 사용하는 것입니다. AWS 서비스 및 연결된 내부 리소스에 사용할 공인 및 사설 TLS(Transport Layer Security) 인증서를 손쉽게 프로비저닝, 관리 및 배포할 수 있습니다. TLS 인증서는 네트워크 통신을 보호하고 인터넷에서는 웹 사이트 그리고 프라이빗 네트워크에서 리소스의 ID를 설정하기 위해 사용됩니다. ACM은 Elastic Load Balancer, Amazon CloudFront 배포, API Gateway의 API 등 AWS 리소스와 통합되며 자동 인증서 갱신도 처리합니다. ACM을 사용하여 사설 루트 CA를 배포하는 경우 EC2 인스턴스, 컨테이너 등에서 사용할 수 있도록 인증서와 비공개 키가 제공될 수 있습니다.

전송 중 암호화 적용: 조직, 법률 및 규정 준수 요구 사항을 충족하기 위해 적절한 표준 및 권장 사항에 따라 정의된 암호화 요구 사항을 적용합니다. AWS 서비스는 통신에 TLS를 사용하는 HTTPS 엔드포인트를 제공함으로써 AWS API와 통신할 때 전송 중 암호화 기능을 제공합니다. HTTP와 같은 비보안 프로토콜은 보안 그룹을 사용하여 VPC에서 감사 및 차단할 수 있습니다. HTTP 요청은 Amazon CloudFront의 HTTPS로나 Application Load Balancer에서 자동으로 리디렉션될 수도 있습니다. 컴퓨팅 리소스를 완전하게 제어하여 서비스 간에 전송 중 암호화를 구현할 수 있습니다. 외부 네트워크로부터 특정 VPC로의 VPN 연결을 사용하여 트래픽을 쉽게 암호화할 수도 있습니다. 특별한 요구 사항이 있는 경우 AWS Marketplace에서 타사 솔루션을 사용할 수 있습니다.

네트워크 통신 인증: 인증을 지원하는 네트워크 프로토콜을 사용하여 당사자 간 신뢰를 맺을 수 있습니다. 이것이 프로토콜에서 사용되는 암호화에 추가되어 통신이 변조되거나 가로채질 위험을 줄입니다. 인증을 구현하는 일반적인 프로토콜에는 많은 AWS 서비스에서 사용되는 TLS(Transport Layer Security)와 AWS Virtual Private Network(AWS VPN)에서 사용되는 IPsec가 포함됩니다.

무단 데이터 침입 탐지 자동화: Amazon GuardDuty와 같은 도구를 사용하여 데이터 분류 수준에 따라 정의된 경계 밖으로 데이터를 이동하려는 시도를 자동으로 탐지합니다. 예를 들어 DNS 프로토콜을 사용하여 알 수 없거나 신뢰할 수 없는 네트워크로 데이터를 복사하는 트로이 목마를 감지할 수 있습니다. Amazon 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. S3 Access Analyzer는 S3 버킷에서 누가 어떤 데이터에 액세스할 수 있는지를 평가하는 데 도움이 될 수 있습니다.

리소스

전송 중 데이터 보호와 관련된 AWS 모범 사례에 대해 자세히 알아보려면 다음 리소스를 참조하십시오.

동영상

- How can I add certificates for websites to the ELB using AWS Certificate Manager
- Deep Dive on AWS Certificate Manager Private CA

설명서

- AWS Certificate Manager
- Application Load Balancer의 HTTPS 리스너
- AWS VPN
- API Gateway 엣지 최적화

인시던트 대응

고도의 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 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.

클라우드 응답의 설계 목표

NIST SP 800-61 Computer Security Incident Handling Guide, 에 정의된 일반적인 사고 대응 프로세스 및 메커니즘도 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 목표로는 문제 억제 및 완화, 영향을 받은 리소스 복구, 포렌식을 위한 데이터 보존, 귀속 등이 있습니다.
- **계획 문서화:** 인시던트에 대응하고, 인시던트 중에 통신하고, 인시던트로부터 복구하는 데 도움이 되는 계획을 수립합니다.
- **클라우드를 사용하여 대응:** 이벤트와 데이터가 발생하는 곳에 응답 패턴을 구현합니다.
- **무엇을 가지고 있고 무엇이 필요한지 파악:** 로그, 스냅샷 및 기타 증거를 중앙 집중식 보안 클라우드 계정에 복사하여 보존합니다. 보존 정책을 적용하는 태그, 메타데이터, 메커니즘을 사용합니다. 예를 들어 Linux dd 명령이나 이에 해당하는 Windows 명령을 사용하여 조사 목적을 위한 데이터의 전체 복사본을 만들 수 있습니다.
- **재배포 메커니즘 사용:** 보안 문제의 원인이 잘못된 구성일 수 있는 경우, 적절한 구성의 리소스를 재배포하여 변형을 제거하는 것만으로 간단하게 해결할 수 있습니다. 가능하면 알 수 없는 상태의 환경에서 응답 메커니즘을 두 번 이상 실행할 수 있도록 합니다.

- **가능한 경우 자동화:** 반복되는 문제나 인시던트의 경우, 프로그래밍 방식으로 분류하고 일반적인 상황에 대응하는 메커니즘을 구축합니다. 사람은 특별하고, 새롭고, 중요한 인시던트에 대응합니다.
- **확장 가능한 솔루션 선택:** 조직에서 클라우드 컴퓨팅을 확장하는 방식에 맞추고 탐지와 대응 사이의 시간을 단축하기 위해 노력합니다.
- **프로세스 교육 및 개선:** 프로세스, 도구, 인력의 격차가 발견되면 이를 해결하기 위한 계획을 구현합니다. 시뮬레이션은 격차를 찾고 프로세스를 개선할 수 있는 안전한 방법입니다.

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

- 클라우드 기술과 조직에서 이러한 기술을 어떻게 사용하고자 하는지에 대해 보안 운영 및 인시던트 대응에 참여하는 직원을 **교육**합니다.
- 인시던트 대응팀이 클라우드에서 인시던트를 탐지 및 대응하고, 탐지 기능을 활성화하고, 필요한 도구 및 클라우드 서비스에 대한 적절한 액세스를 보장할 수 있도록 **준비**시킵니다. 또한 안정적이고 일관된 응답을 보장하는 데 필요한 수동 및 자동 런북을 준비합니다. 다른 팀과 협력하여 예상되는 기준 작업을 설정하고, 해당 지식을 사용하여 정상 운영과의 차이를 파악합니다.
- 클라우드 환경 내에서 예상되는 보안 이벤트와 예기치 않은 보안 이벤트를 모두 **시뮬레이션**하여 얼마나 효과적으로 준비했는지 확인합니다.
- 시뮬레이션 결과를 **반복**하여 대응 태세를 강화하고, 가치 창출 시간을 단축하고, 위험을 더 줄입니다.

교육

자동화된 프로세스 덕분에 조직은 워크로드 보안을 강화하는 조치에 더 많은 시간을 할애할 수 있습니다. 또한 자동화된 인시던트 대응을 통해 인시던트 상관관계 파악, 시뮬레이션 연습, 새로운 대응 절차 개발, 연구 수행, 새로운 기술 개발, 새로운 도구 테스트 또는 구축 업무에

인력을 활용할 수 있습니다. 자동화 기능이 향상되어도, 보안 조직 내의 팀, 전문가, 대응 인력에게는 여전히 지속적인 교육이 필요합니다.

일반적인 클라우드 경험을 넘어 더 큰 성공을 거두려면 인력에 크게 투자해야 합니다. 직원에게 프로그래밍 기술, 개발 프로세스(버전 관리 시스템 및 배포 방식 포함), 인프라 자동화를 학습할 수 있는 추가 교육을 제공하는 것이 조직에도 이익이 될 수 있습니다. 가장 좋은 방법은 인시던트 대응 게임 데이를 실행하면서 실습하는 것입니다. 그러면 팀 내 전문가들이 다른 사람들을 교육하면서 도구와 기법을 연마할 수 있습니다.

준비

인시던트 중에 인시던트 대응팀은 인시던트와 관련된 다양한 도구 및 워크로드 리소스에 액세스할 수 있어야 합니다. 이벤트가 발생하기 전에 업무 수행에 필요한 적절한 권한을 팀에 미리 프로비저닝해야 합니다. 이들이 적시에 대응할 수 있도록 이벤트가 발생하기 전에 모든 도구, 액세스, 계획을 문서화하고 테스트해야 합니다.

주요 직원과 외부 리소스 파악: 클라우드에서 인시던트 대응 방식을 다른 팀(예: 법률 자문, 리더십, 비즈니스 이해관계자, AWS Support Services 등)과 함께 정의할 때는 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.

인시던트 관리 계획 개발: 인시던트에 대응하고, 인시던트 중에 통신하고, 인시던트로부터 복구하는 데 도움이 되는 계획을 수립합니다. 예를 들어 워크로드와 조직에서 발생할 가능성이 가장 큰 시나리오부터 인시던트 대응 계획을 시작할 수 있습니다. 사내외에서 커뮤니케이션 및 에스컬레이션할 방법도 포함해야 합니다. 워크로드와 조직에 발생할 가능성이 가장 큰 시나리오부터 시작하여 플레이북 형태의 인시던트 대응 계획을 생성합니다. 이는 현재 생성된 이벤트여도 됩니다. 출발점이 필요한 경우, AWS Trusted Advisor 및 Amazon GuardDuty 결과를 참조하는 것을 권장합니다. 쉽게 유지 관리할 수 있도록 마크다운과 같은 간단한 형식을

사용하되, 다른 설명서를 조회하지 않고도 실행할 수 있도록 중요한 명령이나 코드 조각을 포함해야 합니다.

간단하게 시작하여 반복합니다. 보안 전문가 및 파트너와 긴밀하게 협력하여 프로세스를 가능하게 만드는 데 필요한 작업이 무엇인지 식별합니다. 수행하는 프로세스에 대한 수동 설명을 정의합니다. 그런 다음, 프로세스를 테스트하고 런북 패턴을 반복하여 대응의 핵심 로직을 개선합니다. 어떠한 예외가 있는지 그리고 그러한 시나리오에 대한 대체 해결 방법을 파악합니다. 예를 들어 개발 환경에서 잘못 구성된 Amazon EC2 인스턴스를 종료하고자 할 수 있습니다. 그런데 같은 이벤트가 프로덕션 환경에서 발생한 경우에는 인스턴스를 종료하는 대신 인스턴스를 중지한 후에 중요한 데이터가 손실되지 않으며 종료해도 괜찮은지를 이해관계자와 함께 확인하고자 할 수도 있습니다. 사내외에서 커뮤니케이션 및 에스컬레이션할 방법도 포함해야 합니다. 프로세스에 수동으로 대응하는 것이 익숙할 경우에도 이를 자동화하면 해결 시간이 단축됩니다.

액세스 사전 프로비저닝: AWS 및 기타 관련 시스템에 사전 프로비저닝된 올바른 액세스를 인시던트 대응 인력에게 제공함으로써 조사부터 복구까지 걸리는 시간을 단축할 수 있도록 합니다. 인시던트 상황에서 적절한 담당자의 액세스 권한을 받는 방법을 확인하는 경우 인시던트에 대응하는 시간이 지연되며, 액세스 권한을 공유하거나 긴급 상황에서 적절하게 프로비저닝하지 않는 경우에는 다른 보안 약점도 발생할 수 있습니다. 팀원에게 필요한 액세스 수준(예: 수행할 가능성이 높은 작업 종류)을 알고 있어야 하며, 액세스를 미리 프로비저닝해야 합니다. 보안 인시던트에 대응하기 위해 특별히 생성된 역할 또는 사용자는 액세스가 충분히 부여된 경우가 많습니다. 따라서 이러한 사용자 계정의 사용은 제한적이어야 합니다. 일상적인 활동에 사용해서는 안 되며, 사용되는 경우 알림을 통해 경고해야 합니다.

도구 사전 배포: AWS에 사전 배포된 올바른 도구를 보안 담당자에게 제공함으로써 조사부터 복구까지 걸리는 시간을 단축할 수 있도록 합니다.

보안 엔지니어링 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.

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.

EC2 인스턴스의 운영 체제 내에서 실행되는 도구의 경우, Amazon EC2 인스턴스 운영 체제에 설치한 에이전트를 사용하여 원격으로 안전하게 인스턴스를 관리할 수 있도록 해주는 AWS Systems Manager Run Command를 사용하여 평가해야 합니다. 많은 AMI(Amazon Machine Image)에 기본적으로 설치된 SSM Agent(AWS Systems Manager Agent)가 필요합니다. 하지만 일단 인스턴스가 손상되었으면 해당 인스턴스에서 실행 중인 도구 또는 에이전트의 응답은 신뢰할 수 있는 것으로 간주해서는 안 됩니다.

포렌식 기능 준비: 외부 전문가, 도구, 자동화 등 적합한 포렌식 조사 기능을 식별하고 준비합니다. 일부 인시던트 대응 활동에는 인시던트와 관련된 디스크 이미지, 파일 시스템, RAM 덤프 또는 기타 아티팩트를 분석하는 것이 포함될 수 있습니다. 영향을 받는 데이터 볼륨의 복사본을 마운트하는 데 사용할 수 있는 사용자 지정 포렌식 워크스테이션을 구축합니다. 포렌식 조사 기법에는 전문가 교육이 필요하므로 외부 전문가의 참여가 필요할 수도 있습니다.

시뮬레이션

게임 데이 실행: 시뮬레이션 또는 연습이라고도 하는 게임 데이는 실제 시나리오에서 인시던트 관리 계획 및 절차를 연습할 수 있는 체계적인 기회를 제공하는 내부 이벤트입니다. 게임 데이는 기본적으로 대응 능력을 준비하고 반복적으로 개선하기 위한 것입니다. 게임 데이 활동을 수행하는 것이 중요한 몇 가지 이유는 다음과 같습니다.

- 준비 상태 검증

- 자신감 향상 – 시뮬레이션 및 교육 담당자를 통한 학습
- 규정 준수 또는 계약 의무 준수
- 인증을 위한 아티팩트 생성
- 민첩성 – 점진적 개선
- 속도 향상 및 도구 개선
- 커뮤니케이션 및 에스컬레이션 다듬기
- 드문 상황이나 예기치 않은 상황에 대한 대처 능력 개발

이러한 이유로 SIRS 활동에 참여하는 동안 파생된 가치는 스트레스 이벤트 발생 시 조직의 실효성을 높입니다. 현실적이고 유리한 SIRS 활동을 개발하는 것은 어려운 연습이 될 수 있습니다. 제대로 파악한 이벤트를 처리하는 절차 또는 자동화를 테스트하는 것도 몇 가지 장점이 있지만, 창의적인 SIRS 활동에 참여하여 예기치 않은 상황을 테스트하면서 지속해서 개선하는 경우도 그만한 가치가 있습니다.

반복

억제 및 복구 기능 자동화: 인시던트의 억제 및 복구를 자동화하여 대응 시간과 조직에 미치는 영향을 줄입니다.

플레이북에서 프로세스와 도구를 생성하고 연습한 후에는 로직을 코드 기반 솔루션으로 해체할 수 있습니다. 그리고 이것은 많은 대응 인력들이 조치를 자동화하고 대응 인력의 편차 또는 추측을 없애기 위한 도구로 사용할 수 있습니다. 이렇게 하면 대응 수명 주기를 가속화할 수 있습니다. 다음 목표는 사람 응답자에 의해서가 아니라 알림 또는 이벤트 자체에서 이 코드가 호출되도록 함으로써 완벽히 자동으로 이루어지는 이벤트 중심의 대응을 생성하는 것입니다.

이벤트 중심의 대응 시스템을 사용하면 탐지 메커니즘이 대응 메커니즘을 트리거하여 이벤트를 자동으로 해결합니다. 이벤트 중심의 대응 기능을 사용하여 탐지 메커니즘과 대응 메커니즘 간의 시간을 단축할 수 있습니다. 이러한 이벤트 중심의 아키텍처를 생성하기 위해 이벤트에 대한 응답으로 코드를 실행하고 기본 컴퓨팅 리소스를 자동으로 관리하는 서버리스 컴퓨팅 서비스인 AWS Lambda를 사용할 수 있습니다. 예를 들어 AWS CloudTrail 서비스가 활성화된 AWS 계정이

있다고 가정해 보겠습니다. AWS CloudTrail이 비활성화된 경우(`cloudtrail:StopLogging` API 호출을 통해) Amazon EventBridge를 사용하여 특정 `cloudtrail:StopLogging` 이벤트를 모니터링하고 AWS Lambda 함수를 호출하여 `cloudtrail:StartLogging`을 호출함으로써 로깅을 다시 시작할 수 있습니다.

리소스

인시던트 대응과 관련된 최신 AWS 모범 사례에 대해 자세히 알아보려면 다음 리소스를 참조하십시오.

동영상

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

설명서

- AWS 인시던트 대응 안내서
- AWS Step Functions
- Amazon EventBridge
- CloudEndure Disaster Recovery

실습

- 실습: AWS 콘솔 및 CLI를 사용한 인시던트 대응
- 실습: Jupyter를 사용한 인시던트 대응 플레이북 - AWS IAM
- 블로그: AWS Step Functions로 보안 인시던트 대응 조율

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.

기고자

다음은 본 문서 작성에 도움을 준 개인 및 조직입니다.

- Ben Potter, Well-Architected 보안 부문 수석 담당자, Amazon Web Services
- Bill Shinn, CISO 오피스 선임 수석, Amazon Web Services
- Brigid Johnson, 선임 소프트웨어 개발 관리자, AWS Identity, Amazon Web Services
- Byron Pogson, 선임 솔루션 아키텍트, Amazon Web Services
- 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
- Paul Hawkins, 선임 보안 전략가, Amazon Web Services
- Sam Elmalak, 선임 기술 책임자, Amazon Web Services

추가 자료

자세한 내용은 다음 출처를 참조하십시오.

- AWS Well-Architected 프레임워크 백서

문서 개정

날짜	설명
2020년 7월	계정, 자격 증명 및 권한 관리에 관한 지침을 업데이트하였습니다.
2020년 4월	모든 영역에서 더 많은 조언을 제공하고 새로운 모범 사례, 서비스 및 기능을 추가하기 위해 업데이트하였습니다.
2018년 7월	새 AWS 서비스 및 기능과 업데이트된 참조를 반영하여 업데이트하였습니다.
2017년 5월	새로운 AWS 서비스 및 기능을 반영하기 위해 시스템 보안 구성 및 유지 관리 섹션을 업데이트하였습니다.
2016년 11월	최초 게시