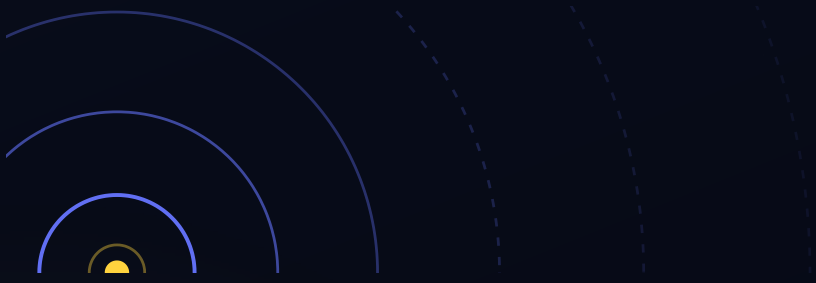
Cremit

크리밋 리서치 · 사고 대응 시리즈

Blast Radius

유출된 API 키와 클라우드 크리덴셜을 조사하는 실무 플레이북입니다. AWS, GCP, Azure와 그 배후의 시크릿 저장소를 대상으로 해당 키가 어디까지 접근할 수 있었는지를 증거로 입증하는 절차를 다룹니다.



01	02	03	04
키 1개	주체 1개	해당 주체가 수행 가능한 모든 역할	해당 역할의 권한이 미치는 전체 범위(영향 범위)

발행 2026년 9월 4일

대상 사고 대응 담당자, DevSecOps, 플랫폼 및 클라우드 보안팀

여기부터

유출 직후 첫 5분 대응 조치

API 키 유출 사고에 대응 중이라면 본 페이지만 우선 확인하면 됩니다. 나머지 내용은 사고 대응 중과 사후에 참조하는 자료입니다. **다음 세 가지 조치를 최초 5분 이내에 실행합니다.**

1. **차단 조치 전 증거 보존.**
   노출 지점 캡처/내보내기 수행 및 발견 시각(UTC) 정확한 기록. 사전 폐기 시 해당 키의 접근 범위 및 사용 주체 입증 증거가 소멸함.
2. **인증 절차 없는 소유 주체 식별.**
   접두사와 구조 분석만으로 발급처, 크리덴셜 유형, 소유 계정을 오프라인에서 파악 가능함.
3. **악용 가능성 전제.**
   인터넷에 노출된 키는 **1~5분** 이내에 악용될 수 있음. "사용 흔적 미발견"은 조사가 미완료되었음을 의미하며 안전을 보장하지 않음. 주요 3대 클라우드 중 2곳은 시크릿 조회 시 기본 설정에서 로그를 남기지 않음.

⚠ 증거 보존 시 자주 발생하는 실수

키 즉시 삭제, force-push를 통한 커밋 삭제, 영향 범위를 확보하기 전의 시크릿 로테이션입니다. 세 행위 모두 감사 기록을 훼손할 뿐 공격자가 이미 확보한 시간은 전혀 줄이지 못합니다.

ⓘ 인지 시점 기준 신고 기한

대부분의 규제는 인지 시점부터 정해진 시간 내 신고를 의무화합니다. 여기서 인지란 내용의 최종 이해 여부와 무관하게 정보를 수신한 시점을 의미합니다.
정보통신망법상 침해사고 신고는 24시간, 개인정보 보호법상 유출 신고는 72시간, EU NIS2 최초 신고는 24시간 이내입니다.

✓ 침해사고 증적 수집 방법

부록 A(28–29페이지)를 참조합니다. 휘발성이 높은 순서로 로그 소스를 정리했으며 수집 스크립트를 함께 제공합니다. 실행하면 결과가 해시 처리된 케이스 폴더 하나에 저장됩니다.

플레이북 6단계

각 장은 개별 단계에 대응하며 현재 단계는 페이지 상단에 표시됩니다.

01	02	03	04	05	06
식별	**노출 범위**	**영향 범위**	**사용 조사**	**차단 및 로테이션**	**사고 종료**
크리덴셜 유형 및 소유자 파악	유출 경로 및 노출 기간 확인	크리덴셜 도달 범위 및 수행 가능 역할 전수 분석	사용 주체 및 행위 이력 조사	장애 없는 안전한 폐기 절차	완전한 종료 상태 객관적 증명

목차

대응 속도보다 대응 순서가 중요한 이유

동시에 작동하는 세 개의 시계

공격자의 자동화 악용은 **분** 단위로 진행되는 반면 규제상 요구 시한은 **시간** 단위이며 탐지에 소요되는 시간은 실제로 **연** 단위에 달하는 경우가 많습니다. 세 시계가 동시에 작동하므로 폐기 속도만 최적화한 플레이북으로는 대응 목적을 달성할 수 없습니다.

T1 → T2 실제 진행 속도

1분
공개 리포지토리에 노출된 AWS 키가 최초 악용되기까지 관측된 최단 시간(Comparitech 허니팟).

4분
AWS의 격리 정책 적용 시점부터 공격자 정찰이 시작되기까지 걸린 시간(Unit 42).

400+
해당 공격자가 7분 동안 호출한 API 건수.

키가 GitHub에 푸시된 지 2분 만에 AWS 자동화가 `AWSCompromisedKeyQuarantine`을 적용했음에도 공격자의 침입을 막지 못했습니다. 자동화된 악용 속도가 자동화된 방어보다 빠르다는 뜻입니다. 노출이 확인되면 이미 악용된 것으로 전제하고 대응에 착수합니다.

법적 신고 기한

규제	기한
정보통신망법 제48조의3	인지 시점부터 **24시간** 이내 KISA 보호나라 또는 과기정통부에 침해사고 신고
개인정보 보호법 제34조	인지 시점부터 **72시간** 이내 정보주체 통지 및 신고
GDPR Art. 33	인지 시점부터 **72시간** 이내
EU NIS2 Art. 23	**24시간** 이내 조기 경보, 72시간 이내 통지, 1개월 이내 최종 보고
미국 SEC (상장사)	중대성 판단 후 영업일 기준 4일 이내 8-K Item 1.05 제출
PCI DSS	즉시 (카드 브랜드 규정에 따름)

국내 사업자에게 가장 촉박한 기한은 정보통신망법의 **인지 후 24시간**입니다. 유출 크리덴셜로 시스템에 접근한 사실이 확인되면 침해사고에 해당하며 개인정보가 포함되면 72시간 유출 신고가 별도로 따릅니다. 두 기한 모두 영향 범위 산정 전이므로 최초 신고는 조사가 끝나기 전에 이뤄집니다.

"인지"의 정확한 산정 시점

위 신고 기한의 산정 기준은 대부분 인지 시점이며 SEC는 중대성 판단 시점입니다. 어느 규제도 해당 기준을 "보안팀의 티켓 확인 시점"으로 인정하지 않습니다. 사전에 확정된 기준이 있어야 긴급 상황에서도 규제를 준수할 수 있습니다.

법적·실무적 해석

인지는 **조직 내 임의의 구성원**이 침해 가능성을 판단할 수 있는 정보를 수신한 시점에 개시됩니다. 개발자 메일함에 수신된 스캐너 알림 역시 조직이 수신한 것으로 간주됩니다.

대표 사고 사례

xAI 사례의 경우 알림 통보는 **3월 2일**에 수신되었으나 키는 **4월 30일**까지 유효했습니다. 상기 해석을 적용하면 보안팀 관여 전 이미 8주가 경과한 상태였습니다.

필수 조치 사항

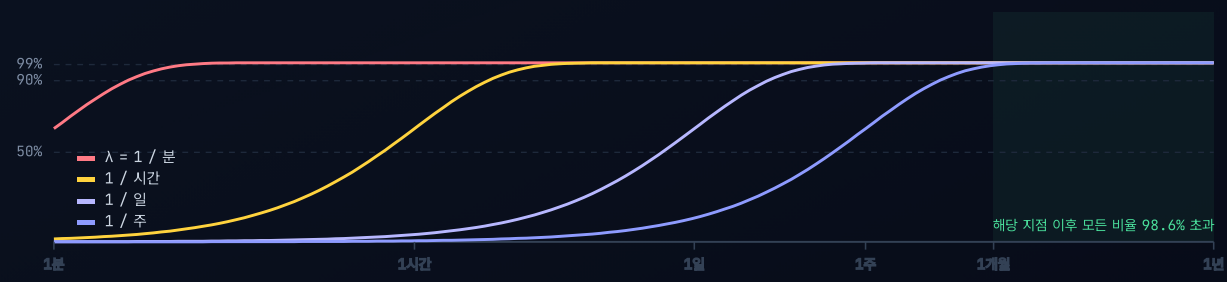
시크릿 탐지 결과는 즉각 조치 권한이 있는 전담 팀으로 에스컬레이션되도록 절차를 일원화합니다. 커밋 당사자에게만 전달되면 조치로 이어지지 않습니다. 이 절차 하나로 규제상 인지 시점과 보안팀의 탐지 시점이 일치합니다.

수치화된 직관

탐지를 전제해야 하는 이유

"로그에 기록이 없는데 노출된 크리덴셜을 왜 침해로 간주하는가"라는 질문에 대한 답은 짧습니다. 탐지가 비율 λ로 무작위 발생한다고 가정하면 구간 W 내 최소 1회 탐지될 확률은 $1 - e^{-\lambda W}$입니다. **핵심은 결과가 λ와 무관해지는 시점이 얼마나 신속히 도달하는가입니다.**

그림 2 · 노출 기간 대비 발견 확률



10^4배 범위에 걸친 4가지 비율입니다. 초기 1시간 동안은 격차가 크나 1개월이 경과하면 완전히 수렴합니다. 음영 구간은 λ 값과 무관하게 동일한 결론에 도달하는 영역입니다.

노출 기간	λ=1/분	1/시간	1/일	1/주
1분	63.2%	1.7%	0.1%	0.0%
1시간	>99.99%	63.2%	4.1%	0.6%
1일	>99.99%	>99.99%	63.2%	13.3%
1주	>99.99%	>99.99%	99.9%	63.2%
1개월	>99.99%	>99.99%	>99.99%	98.6%
1년	>99.99%	>99.99%	>99.99%	>99.99%

주 1회 스캔은 인터넷 외부 노출 자산 기준의 보수적 측정치입니다.

실제 사고 대입: Toyota 5년, Microsoft AI 3년, Mercedes-Benz 3.5개월, xAI 2개월 노출되었습니다. **4건 모두 λ와 무관하게 유출이 확정되는 영역에 위치합니다.** 로그 기록 유무와 상관없이 해당 키가 발견되지 않았을 가능성은 처음부터 없었습니다.

가장 보수적인 비율 기준 도요타 노출 기간 산출

```
# W = 2017-12 ~ 2022-09 = 1,749일.
# 표 내 가장 느린 비율인 주 1회 탐지 적용.
lam = 1/7            # 일 기준 산출
W   = 1749
P   = 1 - math.exp(-lam*W)
# P = 1 - e^-249.9 → 소수점 108자리까지 1.0.
#
# 키 미탐지를 입증할 만큼 느린 비율이나
# 이를 번복할 반올림 수치는 존재하지 않음.
```

구간 확정 불가 시 대응

입증 가능한 최단 구간과 최장 구간으로 각각 산출하여 두 결과를 모두 보고합니다. **하한선에서 이미 99%를 초과할 경우 구간 논쟁은 판단 결과를 변경시키지 못합니다.** 추가 논의는 결론을 바꾸지 못하는 수치에 불필요한 시간을 소모하는 일입니다.

✓ 의사결정 기준

노출 기간이 **분** 단위일 경우 결론이 확정되지 않았으므로 조사 자원을 투입합니다. **일 단위 이상**일 경우 탐지 여부는 이미 확정되었습니다. 남은 질문은 해당 키를 통한 행위와 도달 범위이므로 초기 1시간을 이에 집중 배분합니다.

ⓘ 수학적 모델의 범주

본 모델은 **탐지 여부**를 다루며 직접적인 피해 규모를 의미하지 않습니다. 탐지 주체가 모두 공격자이거나 악의적 행위를 수행하는 것은 아닙니다. **보고서에는 "탐지된 것으로 전제함"으로 명시하며 "실제 악용 확인" 표현은 사용하지 않습니다.**

λ 수치의 출처

실측 수치는 최상단 행뿐입니다. 허니팟 연구 기준 공개 AWS 키의 최초 사용 시간은 1~5분입니다. 이는 스캔이 가장 집중되는 영역의 수치입니다. 비공개 리포지토리, CI 로그, 채팅 스레드 내 키의 탐지 비율은 공식 발표된 바 없습니다. 단일 수치를 단정하는 대신 4가지 시나리오를 병렬 제시한 이유입니다.

λ 수치의 급변 시점

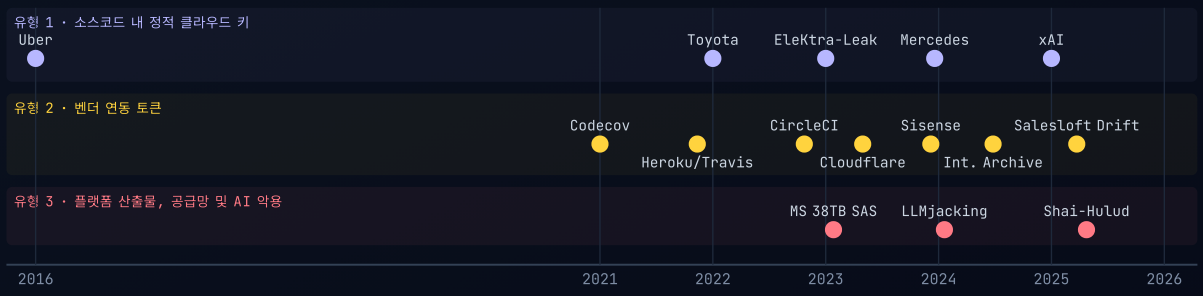
본 모델은 λ를 상수로 가정하지만 실제로는 계단식으로 급증합니다. 리포지토리 Star/Fork 추가, 검색엔진 색인, 패키지 배포, 이미지의 공개 레지스트리 게시 시점이 이에 해당합니다. **노출 기간이 며칠 수준이더라도 상기 이벤트가 발생했다면 하단 행의 비율을 적용합니다.**

실증적 근거 기반

본 플레이북의 근거가 된 침해사고 15건

15건 모두 공식 공개된 사고로 관련 기록이 남아 있습니다. 침해의 발단은 모두 크리덴셜이었으며 악성코드나 소프트웨어 결함에서 시작된 사고는 없습니다. 크리덴셜 유형은 변해 왔으나 **실패 양상은 동일합니다.**

그림 3 · 10년 간의 3대 유형 2016 → 2026 Cremit



수평 축은 사고 발생 연도입니다. 세 유형은 순차적으로 출현했으며 2025년 기준 세 유형 모두 발생했습니다. **사고의 핵심이 "개발자의 키 커밋"에서 "미발급 크리덴셜의 제3자 보유"로 전환되었습니다.** 자산 인벤토리는 양 영역을 모두 포함해야 합니다.

연도	침해사고	크리덴셜 및 노출 위치	침해 영향	시사점
2016	**Uber**	비공개 GitHub 리포지토리 내 AWS 키. 탈취된 GitHub 계정으로 접근	5,700만 건 유출, 합의금 1.48억 달러, CSO 형사 처벌	비공개 리포지토리는 경계가 아님. GitHub → AWS → S3 전체가 영향 범위
2021	**Codecov**	Docker 이미지 레이어에서 추출 가능한 GCS 크리덴셜	업로더 약 2개월간 변조, 고객사 23,000여 곳의 CI 환경변수 탈취	이미지 레이어 내부 시크릿 노출 위험. 단일 벤더 키가 고객 전체 시크릿 유출로 확산
2022	**Toyota**	협력업체가 공개 상태로 푸시한 T-Connect DB 접근 키	296,019건 유출, 2017-12 ~ 2022-09 간 공개 노출	노출 기간이 수년에 달함. 자사 시크릿이 협력사에 의해 커밋됨
2022	**Heroku / Travis**	탈취된 GitHub OAuth 연동 토큰	수십 개 조직의 비공개 리포지토리 복제	연동 토큰은 최초 부여된 권한을 지속 유지함
2023	**CircleCI**	악성코드를 통한 SSO 세션 탈취. 해당 계정은 운영 토큰 발급 권한 보유	고객 환경변수, 토큰, 키 탈취. 전체 고객 대상 로테이션 통보	저장소 침해 시 영향 범위는 보관된 시크릿 전체로 확대됨
2023	**Microsoft AI**	리포지토리 내 Azure SAS 토큰. 계정 전체 권한, 만료일 **2051년**	38TB 유출. 워크스테이션 백업, 서비스 비밀번호, Teams 메시지 3만 여건	SAS는 클라이언트에서 생성되어 플랫폼 측 생성/사용 추적 불가
2023	**EleKtra-Leak**	자동화 스캐너를 이용해 GitHub에서 수집된 IAM 키	2분 만에 격리되었으나 4분 뒤 공격자 활동 개시 (7분 간 400회 이상 API 호출)	자동화된 공격 속도가 자동화된 방어 메커니즘을 상회함
2023	**Cloudflare**	"미사용"으로 오판하여 로테이션에서 누락된 토큰 1개 및 서비스 계정 3개	27일 후 공격자가 해당 크리덴셜로 Atlassian 및 형상관리 시스템 재침입	로테이션의 핵심은 속도가 아닌 완전성(누락 방지)임
2024	**Mercedes-Benz**	공개 리포지토리 내 무제한 권한의 GitHub 토큰	사내 GitHub Enterprise 소스코드, 추가 키 및 접속 문자열 유출	소스코드 접근은 크리덴셜 접근과 동일함. 단일 토큰 유출이 다수 크리덴셜 노출로 확산
2024	**Sisense**	자체 호스팅 GitLab 인스턴스 내 크리덴셜 (S3 접근 권한 포함)	CISA 권고에 따라 전체 고객사의 Sisense 연동 크리덴셜 재설정	벤더의 Git 서버가 자사의 크리덴셜 공격 표면이 됨
2024	**Internet Archive**	2022-12부터 노출된 GitLab 설정 토큰, 및 로테이션되지 않은 Zendesk 토큰	계정 정보 3,100만 건, 문의 티켓 80만 여건 유출 (1개월 간 2회 침해)	불완전한 로테이션은 2차 침해 사고를 유발함
2024	**LLMjacking**	탈취한 클라우드 크리덴셜을 통한 관리형 AI 모델 엔드포인트 악용	피해 기업 추론 비용 발생 (일 수만 달러 규모)	피해 범위가 데이터 유출을 넘어 모델 추론 비용 자체로 확장됨
2025	**xAI**	임직원 개인 공개 리포지토리 내 x.ai API 키	미공개 모델 포함 약 60개 키 노출. 3/2 통보 후 4/30까지 유효 상태 유지	단순 탐지는 조치가 아니며 기한 내 조치를 전담할 책임자가 필요함
2025	**Salesloft Drift**	Salesforce 채팅 연동용 OAuth 토큰 탈취	10일 간 700개 이상 조직의 Salesforce 데이터 외부 반출	공격자는 티켓 내 AWS 키 및 Snowflake 토큰을 탐색하여 2차 침투를 시도함
2025	**Shai-Hulud**	npm 메인테이너 토큰 탈취. 웜 악성코드가 빌드 호스트에서 `TruffleHog` 실행	1차 500여 개 버전, 2차 796개 패키지 감염 (주당 다운로드 2,000만 회 이상)	공격자 역시 동일한 시크릿 스캐너를 사용하며 변경 통제 절차를 우회함

15개 사고 사례의 공통적 시사점

유출 크리덴셜 대응 6대 원칙

산업군, 조직 규모, 유출 크리덴셜 종류가 달라도 사고의 실패 양상은 6가지 패턴으로 수렴합니다.

원칙 01

노출은 자발적으로 만료되지 않습니다.

단 10초간 노출된 시크릿이라도 영구 노출된 것으로 간주하며 force-push 조치로도 완전히 삭제되지 않습니다. GitHub은 도달 불가능한 커밋 객체를 그대로 보관합니다. 해당 SHA를 포함한 푸시 이벤트는 공개 이벤트 아카이브에 영구 기록됩니다. 공격자와 연구자는 이 같은 커밋을 대규모로 스캔하여 여전히 유효한 크리덴셜을 추출합니다.

사고 사례 **Toyota**: 5년간 공개 방치. **Microsoft AI**: 만료일이 2051년인 SAS 토큰을 리포지토리에 3년간 방치.

대응 지침: 커밋 삭제에 의존하지 않고 즉시 로테이션을 수행합니다. 커밋 삭제는 보안 조치가 아닌 사후 정리 작업입니다.

원칙 02

탐지는 조치가 아닙니다.

대부분의 조직은 이미 탐지 알림을 수신하고 있습니다. 부재한 것은 담당자, 이행 기한, 검증 단계입니다. "도구가 알림을 발송함"과 "해당 크리덴셜의 동작이 중단됨" 사이의 시간적 공백에서 피해가 확산됩니다.

사고 사례 **xAI**: 개발자는 3월 2일 알림을 수신했으나 키는 4월 30일까지 유효했음. 약 **60일** 후 벤더가 보안팀으로 직접 에스컬레이션한 뒤에야 폐기됨.

대응 지침: 모든 탐지 건에 담당자를 지정하고 크리덴셜 기준 폐기 SLA를 수립·적용합니다.

원칙 03

크리덴셜은 단순 문자열이 아닌 관계 그래프입니다.

해당 키의 읽기/쓰기 권한, 수행 가능 역할, 사칭 및 발급 가능 대상을 전수 분석하고 연쇄 도달 범위를 파악해야 합니다. 역할 체이닝, 서비스 계정 사칭(Impersonation), 위임 앱 권한은 모두 최초 주체를 넘어 영향 범위를 확장시킵니다.

사고 사례 **Uber**: GitHub 크리덴셜 → 비공개 리포지토리 → AWS 키 → S3 → 5,700만 건 유출. **Mercedes-Benz**: 단일 토큰 → 사내 소스 전체 → 추가 키 및 접속 문자열 노출.

대응 지침: 영향 범위는 개발자의 기억이 아닌 클라우드별 정책 그래프(Policy Graph) 분석으로 산정합니다.

원칙 04

단 하나의 누락만으로도 로테이션은 실패합니다.

부분적인 로테이션은 미수행과 다름없습니다. 공격자에게는 누락된 단 하나의 크리덴셜로도 충분하기 때문입니다. "미사용 자산으로 판단함"은 침해 사고에서 가장 치명적인 오판 원인이며 여러 조직에서 반복 관측됩니다.

사고 사례 **Cloudflare**: 수천 개 자산 중 토큰 1개와 서비스 계정 3개를 미사용으로 오판함. 27일 후 공격자가 해당 4개 자산을 악용함. **Internet Archive**: 1차 사고 후 로테이션에서 누락된 토큰으로 인해 1개월 내 2차 침해 발생.

대응 지침: 자산 인벤토리 교차 검증을 거친 완전 로테이션을 수행합니다. 기억에 기반한 체크리스트에 의존해서는 안 됩니다.

원칙 05

시크릿의 실제 조회 여부는 기본 로그만으로 확인할 수 없습니다.

시크릿 저장소 침해의 핵심은 "실제 시크릿 값의 조회 여부"입니다. 그러나 주요 3대 클라우드 중 2곳은 기본 설정 상태에서 이를 기록하지 않습니다. GCP `AccessSecretVersion`은 데이터 접근 로그에 해당하여 기본적으로 비활성화되어 있습니다. Azure Key Vault는 진단 설정 없이는 기록을 남기지 않으며 Azure SAS 사용 이력은 플랫폼 차원에서 추적할 수 없습니다.

사고 사례 **Microsoft AI** 사례가 대표적임. 해당 토큰은 활성 상태였으나 플랫폼 로그상 어디에도 기록되지 않았음.

대응 지침: 로깅 정책은 사고 발생 이전에 확정·적용합니다. 사고 발생 후에는 필요한 로그가 기록되어 있지 않아 소급 확보가 불가능합니다.

원칙 06

폐기 명령과 실제 접근 차단 사이에는 시차가 존재합니다.

모든 클라우드 환경에서 폐기 조치 시점과 실제 접근 차단 시점 사이에 시차가 존재하며 일반적인 폐기 명령은 이 시차를 완전히 해소하지 못합니다. GCP 서비스 계정 키를 삭제하더라도 이미 발급된 액세스 토큰은 즉시 무효화되지 않습니다. AWS IAM 키를 비활성화하더라도 활성화된 `ASIA…` 세션은 유지됩니다. Azure SAS 토큰은 삭제 자체가 불가능합니다.

클라우드별 실제 동작 GCP: 차단 조치 후 최대 **60분**간 접근이 지속됨. AWS: `aws:TokenIssueTime` 기반 Deny 정책 적용이 필요함. Azure: 계정 키 로테이션 또는 저장된 액세스 정책(Stored Access Policy) 변경이 필요함.

대응 지침: 차단 조치 후 반드시 검증 및 세션 폐기 단계를 수행합니다.

6대 원칙 요약

원칙 1–2: 노출 기간은 예상보다 깁니다. 원칙 3: 영향 범위는 예상보다 넓습니다. 원칙 4–6: 차단 조치는 예상보다 불완전합니다. **대응 방안은 하나입니다. 자산 인벤토리와 클라우드 감사 로그에 근거하여 판단하며 추측성 직관에 의존하지 않습니다.**

3대 클라우드 메커니즘 차이점

동일한 질문, 세 가지 답

다음 10개 페이지는 AWS, GCP, Azure를 순서대로 다루며 각 클라우드별 증거 위치, 영향 범위, 차단 절차를 동일한 구조로 설명합니다. 본 표에는 세 클라우드의 답이 갈리는 주요 항목만 먼저 정리했습니다. 표의 세부 수치와 내용은 상세 페이지에서 근거와 함께 다시 제시됩니다.

조사 항목	AWS	GCP	AZURE
크리덴셜 자체 분석 가능 범위 오프라인 분석 (API 호출 없음)	소유 계정 번호까지 `AKIA` 디코딩	프로젝트와 서비스 계정 주소 `service_account` JSON	부여 내역 전체 `si` · `skoid` 파라미터
시크릿 조회 기본 기록 여부 원칙 05	✓ 기본 켜짐 CloudTrail 관리 이벤트 기록	✕ 기본 꺼짐 `DATA_READ` 미설정 시 미기록	✕ 기본 꺼짐 진단 설정 전 미기록
기본 로그로 확인 가능한 구간 원칙 01	90일 이벤트 히스토리 (리전별)	400일 / 30일 Admin Activity / Data Access	7일 / 30일 Entra Free / P1·P2
비인간 주체 사용 기록 위치	CloudTrail 한 곳 사용자 호출과 동일한 트레일	감사 로그 두 종 크리덴셜 종류별 기본값 상이	별도 테이블 `AADServicePrincipal…`
영향 범위 누락 빈발 지점 원칙 03	리소스 기반 정책 주체 측면에서 열거 불가	상속 · 가장 · 도메인 전체 위임 바인딩 정보만으로 확인 불가	RBAC와 Graph 이중 체계 단방향 열거 시 50% 누락 발생
폐기 즉시 적용 여부 원칙 06	◑ 키 비활성화만으로는 부족 활성화된 `ASIA…` 세션 유지	✕ 액세스 토큰 최대 60분 키 삭제 후에도 토큰 만료까지 유지	✕ SAS는 개별 폐기 불가 발급 목록 및 폐기 엔드포인트 미제공
세션·토큰 무효화 방안	`aws:TokenIssueTime` deny 발급 시각 기준 거부 정책 적용	계정 60분 비활성화 또는 바인딩 제거 인가 권한 요청 시마다 재평가	계정 키 또는 저장된 액세스 정책 로테이션 해당 키 기반 발급 토큰 전체 실효
조사 착수 명령어	`cloudtrail lookup-events`	`gcloud logging read`	KQL · `AADServicePrincipal…`

클라우드별 주요 오판 사례

AWS

격리 정책이 적용되었으니 차단되었다고 판단합니다.

AWSCompromisedKeyQuarantineV3는 과금 유발 액션을 중심으로 차단하며 데이터 접근 액션은 허용 상태로 남습니다.

GCP

키를 삭제했으니 종료되었다고 판단합니다.

서비스 계정 키 삭제는 이미 발급된 액세스 토큰을 무효화하지 않으며 해당 토큰은 **최대 60분간** 유효합니다.

AZURE

로그인 기록이 없으니 사용되지 않았다고 판단합니다.

비인간 주체의 로그인은 일반 사용자 로그인과 다른 테이블에 기록됩니다. 조회한 테이블이 애초에 다릅니다.

모든 클라우드 공통사항

원칙 01

노출은 만료되지 않습니다.

단 10초간 노출된 크리덴셜도 영구 노출 상태로 간주하며 노출을 취소하는 절차는 존재하지 않습니다.

원칙 02

탐지는 조치가 아닙니다.

알림부터 실제 폐기까지의 간격은 담당자와 SLA가 결정하며 이 시차가 사고 피해 규모를 결정합니다.

원칙 04

부분 로테이션은 미수행과 같습니다.

단 하나의 누락된 키로도 침해가 성립합니다. 인벤토리 교차 검증만이 유일한 해결책입니다.

ⓘ 비교 표 해석 시 유의사항

항목별 차이가 특정 클라우드의 우월성을 의미하지는 않습니다. **차이점은 사고 발생 후 무엇을 입증할 수 있는지에 있습니다.** 해당 차이는 대부분 사고 훨씬 전에 내린 로깅 설정 정책에서 비롯됩니다.

플레이북 전체 흐름도

초기 대응 의사결정 트리(Decision Tree)

각 단계는 본 플레이북의 개별 장에 대응하며 순차 진행됩니다. 두 갈래 분기는 불가역적 단계를 반영합니다. 크리덴셜이 노출된 이상 "안전함" 상태로 복귀할 수 없으며 시크릿 저장소 접근 권한이 노출되는 순간 사고는 다중 분기합니다.

인증 없이 크리덴셜에서 도출 가능한 정보

오프라인 크리덴셜 분석 식별표

크리덴셜은 포맷 자체로 종류가 식별됩니다. 접두사, 문자열 길이, 내부 구조만으로 발급처, 크리덴셜 종류, 소유 계정을 오프라인에서 식별할 수 있습니다. 전체 과정이 오프라인에서 수행되므로 API 호출, 로그 기록, 증거 훼손 위험이 없습니다.

AWS 식별자 접두사 목록

상위 4글자를 기준으로 구분하며 ASIA의 경우 차단 방식이 상이하므로 접두사를 최우선으로 확인해야 합니다.

접두사	식별 대상
AKIA	IAM 사용자 장기 액세스 키 (비활성화 후 삭제)
ASIA	임시 STS 키 (**삭제 불가**, 세션 Deny 필요)
AIDA	IAM 사용자 고유 ID
AROA	IAM 역할 (aws:userId 내 AROA…:session 형태)
AIPA	EC2 인스턴스 프로파일
ANPA / ANVA	관리형 정책 / 정책 버전
AGPA	사용자 그룹
ABIA / ACCA	STS 베어러 토큰 / 컨텍스트 전용 크리덴셜

발급처별 토큰 구조

토큰 형태를 기반으로 참조할 장을 결정합니다.

토큰 형태	크리덴셜 유형
ghp_ · github_pat_	GitHub PAT (Classic 또는 Fine-grained, 권한 조회 방식 상이)
gho_ · ghu_ · ghs_ · ghr_	GitHub OAuth / User-to-server / App 설치 / 갱신 토큰
xoxb- · xoxp- · xapp-	Slack Bot / User / App 레벨 토큰
AIza…	Google API 키 (소유자 정보 미포함, 제약 조건 설정이 유일한 통제 수단)
ya29.…	Google OAuth2 액세스 토큰 (단기 유효, 즉시 폐기 불가)
sk- · sk-ant-	AI 모델 벤더 API 키 (과금 내역이 유일한 징후인 경우 다수)
sk_live_ · rk_live_	Stripe 시크릿 / 제한 키
npm_ · pypi-	패키지 레지스트리 토큰 (공급망 사고 절차 적용)
eyJ…	JWT (헤더/클레임 디코딩 수행, exp 및 aud로 영향 범위 한정)

네트워크 접근 전 확정 가능 항목

오프라인 분석 기반의 초기 1시간

초기 분류 목적의 API 호출은 사이드 이펙트를 유발합니다(타임라인 오염, `last-used` 메타데이터 덮어쓰기, 외부 알림 유발 등). **오프라인 분석은 이러한 리스크 없이 예상보다 많은 정보를 제공합니다.**

AWS 액세스 키 내 계정 ID 추출

AWS 액세스 키 ID는 접두사 4자 뒤 16자의 Base32 구조로 구성됩니다. 디코딩 후 상위 6바이트를 7비트 우측 시프트하면 **소유 계정 번호**가 도출됩니다. 네트워크 호출 없이 유출 키의 소속 계정을 오프라인에서 식별할 수 있습니다.

흔적 없는 소유 계정 식별

```python
import base64

def account_from_key(key_id):
    body = key_id[4:].upper()    # 접두사 제거
    raw  = base64.b32decode(body)  # 16자 → 10바이트 디코딩
    n    = int.from_bytes(raw[:6], 'big')
    return (n & 0x7fffffffff80) >> 7

>>> account_from_key("ASIAY34FZKBOKMUTVV7A")
609629065308
```

오프라인 분석을 통해 확인 가능한 항목은 계정 식별에 한합니다. 키의 현재 유효 여부는 알 수 없으며 위조된 키 역시 동일하게 디코딩될 수 있습니다. 도출된 계정 번호로 사고 대상을 특정하되 필요한 경우에 한해 `get-access-key-info`를 실행합니다.

토큰 자체 검증 구조

GitHub 토큰 포맷의 하위 6자에는 **무작위 본문 값의 CRC32**를 Base62로 인코딩한 체크섬이 포함됩니다. 해당 값을 재계산하면 네트워크 요청 없이 토큰의 포맷 정합성을 확정할 수 있습니다.

초기 분류 단계에서 체크섬의 중요성

```
# 리포지토리 스캔 결과 추출된 후보 문자열 400개.
# 체크섬 미존재 시 유효성 검증을 위해 400회 인증 호출 필요.
# 감사 로그 400건 발생, 외부 알림 유발 리스크,
# 유효 키의 last-used 시각 오염.
#
# 체크섬 존재 시 오프라인 필터링을 거쳐
# 유효 후보에만 선택적 네트워크 호출을 수행함.
ghp_ gho_ ghu_ ghs_ ghr_ github_pat_
#  접두사      30자 본문       6자 체크섬
# \_______/ \_________________/ \____________/
```

`ghp_` 탐지가 대부분 정탐인 반면 `AIza…` 탐지에서 오탐이 빈번한 이유 역시 구조적 검증 가능 여부의 차이에서 비롯됩니다.

포맷 분석을 통해 확인 가능한 정보

포맷	오프라인 검증	문자열 기반 식별 가능 정보	오프라인상 식별 불가 정보
`AKIA · ASIA · ABIA`	◑ 부분 가능	**접두사 기반 크리덴셜 유형, 디코딩 기반 소유 계정 번호**	유효 여부, IAM 주체명, 부여 권한
`ghp_ · gho_ · ghs_ · ghr_ github_pat_`	✓ 가능	**토큰 유형, CRC32 기반 포맷 정합성**	Fine-grained 토큰의 부여 스코프 (오프라인 조회 불가)
`service_account JSON`	✓ 가능	**GCP 프로젝트 ID, 서비스 계정 이메일, 키 ID, 파싱 가능한 RSA 개인키**	부여된 IAM 역할, 키의 실제 삭제 여부
`eyJ… (JWT)`	◑ 구조 분석 가능	**헤더/페이로드 내 발급자(iss), 대상(aud), 주체(sub), 만료일(exp), 발급일(iat)**	발급자 공개키 미보유 시 서명 검증 불가
`SAS query string`	✓ 가능	**권한 범위 전체 (서비스, 리소스 타입, 접근 권한, 만료일, 폐기 가능 여부)**	실제 사용 여부 (SAS 생성/사용은 플랫폼 로그 미추적)
`xoxb- · sk- · sk_live_ npm_ · AIza…`	✕ 불가	발급처, 접두사로 구분 가능한 대상 환경 (그 외 정보 없음)	**소유자, 권한 스코프, 유효 여부 (네트워크 조회 필요)**

초기 대응 자원 배분 원칙

디코딩, 파싱, 체크섬 검증은 부작용 없는 최우선 수행 단계입니다. 자사 관리 계정을 통한 클라우드 메타데이터 조사는 안전하며 조사자 권한으로 기록됩니다. 유출된 크리덴셜을 사용한 직접 인증은 `last-used`를 오염시키고 외부 알림을 유발하므로 최후순위로 진행합니다. **본 순서를 준수해야 포렌식 증거가 온전히 보존됩니다.**

소유 주체 파악, 유효성 검증, 검증 과정의 증거 오염 주의점

유효성 검증 시 오염되는 메타데이터

⚠ 유효성 검증 전 필수 수행 사항

AWS `iam get-access-key-last-used`는 해당 키의 최종 사용 시각을 반환하며 공격자의 악용 여부를 판단하는 핵심 단서입니다. **유출 키로 직접 인증하면 해당 기록이 조사자의 호출 시각으로 덮어써져 원본 증거가 소실됩니다.** 반드시 조사자 관리 계정으로 사전 조회한 뒤 결과를 보존해야 합니다.

인증 없는 소유 계정 확인

AWS · 정보 조회 및 증거 보존

```
# 계정 번호는 오프라인 디코딩으로 확보. 본 명령은 조사자 계정 권한으로
# 수행되며 감사 로그에 조사자 주체로 기록됨.
aws sts get-access-key-info --access-key-id AKIA...
# { "Account": "111122223333" }
# 대상 계정 내에서 유효성 검증 실행 전 수행:
aws iam get-access-key-last-used --access-key-id
AKIA...
# LastUsedDate / ServiceName / Region 정보 보존
```

GCP 및 AZURE · 산출물 분석

```
# GCP: JSON 키 파일 내에 필요한 모든 정보가 평문으로 존재함.
jq '{project_id, client_email, private_key_id}'
key.json
# Azure: 클라이언트 시크릿 자체는 익명임. 소유 주체는 연동된
# appId 조회를 통해 파악함.
az ad sp show --id <appId> --query displayName
```

안전한 크리덴셜 유효성 검증 수칙

- **증거 사전 보존:** 크리덴셜 접근 전 `last-used` 메타데이터, 로그인 감사 로그, 사용량 대시보드 증거를 최우선 확보함.
- **비인가/변경 없는 읽기 전용 호출 제한:** `sts get-caller-identity`, `auth.test`, `GET /user` 등 조회용 명령만 허용하며 쓰기 작업 및 전체 리소스 전수 열거는 금지함.
- **통제된 단일 호스트 활용:** 고정된 이그레스(Egress) IP 및 식별 가능한 User-Agent가 설정된 점프 호스트를 사용하여 감사 로그상 자체 테스트 호출을 명확히 분리함.
- **검증 수행 이력 실시간 기록:** 실행 시각(UTC), 출발지 IP, User-Agent, 실행 명령어를 기록하여 단계 04 위협 헌팅 시 제외 대상 목록으로 활용함.
- **소유자/벤더 알림 발송 전제:** 서드파티 키의 경우 조사자의 검증 호출이 해당 벤더의 사고 인지 계기가 될 수 있음을 고려함.

ⓘ 무효화된 키의 조사 필요성

유효성 검증은 현재 차단이 필요한지 여부만을 판단할 뿐 과거 활성화 기간 동안의 악용 여부는 보장하지 않습니다. 원칙 1과 원칙 5가 동일하게 적용됩니다.

노출 기간 산정 기준표

최초 노출 시점(T1)은 증거 기반으로 확정하며 추정치는 인정되지 않습니다. 아래 출처 중 **가장 이른 시각**을 기준값으로 채택합니다.

증거 출처	확정 가능 항목	확보 방법	신뢰도
커밋 Author 시각	커밋 생성 기준 최선행 노출 추정 시점	`git log --format=%aI -S'<secret>' --all`	높음 (Push 시각과 교차 검증 필요)
커밋 Push 시각	외부 제3자 공개 시점	Events API 또는 감사 로그 (Author 시각 대비 수년 후일 수 있음)	Author 시각과 비교하여 이른 시각 채택
공개 이벤트 아카이브	삭제된 커밋의 과거 외부 공개 입증	공개 GitHub 이벤트 아카이브 내 리포지토리 및 Commit SHA 검색	높음 (리포지토리 삭제 후에도 기록 유지)
크리덴셜 생성 시각	T1 노출 시점의 절대적 하한선	`iam list-access-keys` · `gcloud iam service-accounts keys list`	높음
Last-used 메타데이터	최종 사용 이력 입증	`iam get-access-key-last-used`	정밀도 보통 (자체 유효성 검증 호출 시 덮어써짐)
빌드 산출물 및 CI 로그	배포 이미지 및 빌드 출력물을 통한 노출	`docker history` · 레이어 Diff 분석 · CI 작업 로그 검색	보존 기간에 종속되므로 즉시 확보 필요
벤더 사고 통보	제3자 외부 탐지 시점	벤더 통보 알림, AWS Health 이벤트, 시크릿 스캐닝 알림	높음 (단, 상한선 역할)

두 증거 출처의 결과가 상충할 경우 노출 기간은 교집합이 아닌 합집합 범위로 산정합니다.

Author 시각과 Push 시각 불일치 시

Author 시각은 로컬 커밋 생성 시점이며 Push 시각은 원격 저장소 노출 시점입니다. 지연 푸시 시 두 시각 사이에 수년의 격차가 발생할 수 있으며 Commit Squash를 수행하면 Author 시각이 뒤로 밀려 노출 기간이 축소 평가될 수 있습니다. **두 시각을 모두 확인하여 이른 시각을 T1으로 확정합니다.** 불명확할 경우 크리덴셜 생성 시각을 하한선으로 설정합니다.

유출 경로, 노출 기간, 불가역적 노출 영역 분석

단순 커밋을 초과하는 노출 표면

단일 파일 삭제로 시크릿이 완전히 제거되지는 않습니다. 노출 값은 Git 히스토리, Unreachable 객체, Fork 리포지토리, 공개 이벤트 아카이브, 외부 수집 주체의 데이터베이스에 잔존합니다. **노출 표면별 영구 삭제 가능 여부를 검증해야 하며 대부분 완전 삭제가 불가능합니다.**

그림 5 · 노출 표면 및 완전 삭제 가능 여부

점선 하단 항목은 통제 불가능 영역임. 시크릿이 해당 영역으로 유출된 경우 유일한 대응책은 로테이션 수행임.

Git 히스토리 포렌식 분석

히스토리 전체 대상 시크릿 노출 검색

```
# 파일명이 아닌 시크릿 문자열 기반으로 전체 ref 및 히스토리를 스캔함.
git log --all --format='%H %aI %cI %an' -S'AKIA...'
# %aI = Author 시각, %cI = Commit 시각. 둘 다 Push 시각이
# 아니므로 Push 시각은 감사 로그 또는 이벤트 API에서 확보함.

# Ref가 끊긴 Unreachable 객체 역시 SHA 직접 참조를 통해
# 호스팅 서버에서 조회가 가능하며 SHA는 공개 푸시 이벤트에 잔존함.
git fsck --unreachable --dangling --no-reflogs
git cat-file -p <dangling-sha>
```

Git 추적 제외 영역

```
# 컨테이너 이미지 레이어. 최종 파일시스템 미존재 건도
# 하위 레이어에 잔존할 수 있음.
docker history --no-trunc <image>
docker save <image> | tar -xO | grep -a 'AKIA'

# CI 빌드 로그. 평문 및 마스킹 형태를 모두 검색함.
# UI 마스킹 기능이 원시 로그 수집 단계에서 우회되는 사례 다수.
gh run view <run-id> --log | grep -nE 'AKIA|\*{3,}'

# IaC State 파일. Terraform apply 시 자동 기록되며 관리가 누락됨.
terraform state pull | jq -r '..|strings' | grep -i secret
```

Toyota 사례 2022 Cremit

노출 기간 분석

협력업체가 2017년 12월 고객 데이터베이스 접근 키가 포함된 T-Connect 소스를 **공개** 리포지토리에 푸시했습니다. 발견은 2022년 9월, 범위 내 레코드는 296,019건이었습니다.

TAKEAWAY

정확한 노출 기간에 대한 주관적 진술 대신 Git 커밋 메타데이터 조사를 통해 객관적 수치를 산출했습니다.

Codecov 사례 2021 Cremit

스캔 사각지대 노출 표면

해당 크리덴셜은 리포지토리에 존재한 적이 없습니다. **Docker 이미지 생성 과정**에서 추출할 수 있었습니다. 공격자는 해당 크리덴셜로 2개월간 Bash Uploader를 변조해 고객 **23,000곳 이상**의 CI 환경변수를 수집했습니다.

TAKEAWAY

이미지 레이어, 빌드 로그, 아티팩트는 소스코드와 동일한 수준의 노출 표면입니다.

⚠ Force-push의 한계

Git 호스팅 서비스는 Unreachable 커밋 객체를 일정 기간 보관합니다. 해당 객체는 Commit SHA를 알면 직접 조회할 수 있으며 SHA 값은 원본 Push 이벤트를 통해 공개 아카이브에 보존됩니다. 연구 및 공격 사례 분석 결과 force-push된 커밋 스캔을 통해 유효한 크리덴셜이 다수 복원되었습니다. **따라서 히스토리 재작성은 로테이션 완료 후 수행하는 정리 작업일 뿐 차단 조치가 될 수 없습니다.**

AWS · 1 / 4

AWS 로그 커버리지 기반 조사 한계 확정

모든 AWS 조사는 과거 시점에 설정된 로깅 범위에 맞춰 제한되므로 해당 경계 확정이 먼저입니다. 로깅이 설정되지 않은 구역을 전제한 조사는 오류를 도출합니다.

소스	기본 활성	보존	확인 가능 항목 및 유의점
CloudTrail Event history	✓ 예	90일	리전별 관리 플레인 호출. **데이터 이벤트 미포함.** (객체 읽기 여부 확인 불가)
CloudTrail trail → S3	× 아니오	직접 설정	유일한 영구 사본. 멀티 리전 및 조직 전체 적용 여부 확인 필요. 미설정 시 공격자 활용 리전 전체 누락 위험
CloudTrail data events	× 아니오	트레일에 종속	**S3 `GetObject`, Lambda `Invoke`, DynamoDB 아이템 접근.**미설정 시 "버킷 읽기 여부" 확인 불가
CloudTrail Lake	× 아니오	최대 7–10년	과거 이벤트 SQL 조회. 노출 기간이 긴 사고 시 사고 대응 중 생성 권장. 기존 트레일 데이터 적재 가능
GuardDuty	◑ 계정별	90일	해당 크리덴셜 대상 행위 기반 탐지. `CredentialAccess`, `Discovery`, `PrivilegeEscalation`, `Impact`, `InstanceCredentialExfiltration` 탐지
AWS Health	✓ 예	90일	`AWS_RISK_CREDENTIALS_EXPOSED`. AWS의 공개 영역 내 자사 키 발견 의미. 해당 타임스탬프는 T1의 상한선이며 실제 T1과 다름
IAM last-used / credential report	✓ 예	현재 상태	키별 최종 사용 이력. **자체 유효성 검증 시 덮어써짐.** 최우선 확보 필요
Access Advisor	✓ 예	365일	해당 주체의 실제 사용 서비스 확인. 부여 권한과의 차이가 곧 과다 권한에 해당

경계 확정과 기록 확보

1단계 · 실제 커버리지 확인

```
# 영구 보존되는 멀티 리전 및 조직 전체 트레일 존재 여부 확인
aws cloudtrail describe-trails --query 'trailList[].{n:Name,
  multiRegion:IsMultiRegionTrail, org:IsOrganizationTrail,
bucket:S3BucketName}'

# 데이터 이벤트 수집 여부 확인. 출력 결과 부재 시 객체 단위
# 가시성이 없으며 "데이터 읽기 여부" 확인 불가
aws cloudtrail get-event-selectors --trail-name <name>

# 공격자에 의한 로깅 비활성화 가능성 점검
aws cloudtrail get-trail-status --name <name> \
  --query '{logging:IsLogging, stopped:LatestDeliveryError}'
```

2단계 · 만료 전 로그 구간 확보

```
# 이벤트 히스토리는 90일 보존 및 리전별로 생성되므로 전 리전 순회 점검
# 수집된 데이터 사본만 증거로 인정 (콘솔 화면은 증거 능력 없음)
for r in $(aws ec2 describe-regions \
          --query 'Regions[].RegionName' --output text); do
  aws cloudtrail lookup-events --region "$r" \
    --start-time "$FROM" --max-results 50 \
    --lookup-attributes
AttributeKey=AccessKeyId,AttributeValue="$KEY" \
    > "ct-$r.json"
done

# 노출 기간 90일 초과 시 CloudTrail Lake 이벤트 데이터 저장소를
# 즉시 생성 후 S3 트레일 적재. 이미 소멸된 이벤트 히스토리 기록은
# 복구 불가
```

주요 필드

- `userIdentity.accessKeyId`: 1차 조회 조건. 역할 세션인 경우 키가 `ASIA…`이므로 `sessionContext`로 추가 조회
- `userIdentity.sessionContext.attributes.creationDate`: 세션 시작 시각. 폐기 시 사용하는 `aws:TokenIssueTime`과 동일한 값
- `sourceIPAddress`+ `userAgent`: 공격자와 조사자 구분 지표. 조사자 점프 호스트 IP 우선 제외
- `errorCode`: 다수 액션에서 `AccessDenied` 집중 발생 시 권한 탐색(열거) 흔적으로 판단
- `readOnly`: 타임라인을 정찰과 실행 단계로 구분

ⓘ 전 리전 확인

공격자는 모니터링이 미흡한 리전을 타깃으로 삼습니다. EleKtra-Leak 사례의 첫 호출 API는 `DescribeRegions`였습니다.

⚠ 보고서 기재 커버리지 공백

데이터 이벤트가 비활성화 상태였다면 올바른 결론 표현은 "객체 접근 여부 확인 불가(확인도 배제도 불가능)"입니다. "데이터 접근 흔적 없음"으로 표현하면 비기술자에게는 이상 없음으로 읽히지만 실제로는 로그 공백 상태를 뜻합니다.

AWS · 2 / 4

단일 키의 영향 범위 산출

영향 범위는 6개 권한 부여 메커니즘을 2개 제약 조건으로 필터링한 뒤, 해당 주체가 수임 가능한 모든 역할까지 연쇄 추적하여 산출합니다. 요소 하나만 누락해도 영향 범위는 축소 보고됩니다. 주요 누락 항목인 리소스 기반 정책은 주체가 아닌 리소스 측에 적용되므로 각별한 주의가 필요합니다.

그림 6 · AWS 액세스 키 하나의 권한 도달 범위

주황색 경로는 주체가 아닌 **리소스**에 적용된 권한입니다. 사용자 정책 열거만으로는 드러나지 않으므로 영향 범위 과소평가의 주요 원인이 됩니다.

영향 범위 계산 · 부여 권한 대비 실제 사용 분석

```
# 1. 해당 계정에 부여된 전체 권한 단일 문서로 확보
aws iam get-account-authorization-details --filter User Role \
  Group LocalManagedPolicy > auth.json

# 2. 해당 주체의 수임 가능 역할 확인 (권한 정책이 아닌
#  신뢰 정책 조회)
jq '.RoleDetailList[] | select(.AssumeRolePolicyDocument|tostring
   |contains("user/leaked-user")) | .Arn' auth.json

# 3. 정책 문서 단순 해석 대신 구체적 질의로 검증
aws iam simulate-principal-policy --policy-source-arn <arn> \
  --action-names secretsmanager:GetSecretValue kms:Decrypt \
  --resource-arns '*'

# 4. 리소스 정책을 통한 계정 간 노출 상태 확인
aws accessanalyzer list-findings --analyzer-arn <arn>
# 5. 최근 365일간 실제 사용 서비스 확인. 부여 권한 대비
#  차감한 값이 현재 보유한 과다 권한에 해당함
aws iam generate-service-last-accessed-details --arn <arn>
```

Uber 사례

2016 Cremit

단일 크리덴셜의 권한 연계

탈취된 GitHub 크리덴셜로 **비공개** 리포지토리에 접근한 뒤, 내부 AWS 키를 통해 승객 및 운전자 기록이 저장된 S3 버킷으로 연계 유출됐습니다.

레코드	5,700만 건
합의금	50개 주 및 워싱턴 DC 대상 1억 4,800만 달러
개인 책임	은폐 사유로 CSO 형사 유죄 판결

TAKEAWAY

GitHub 크리덴셜 단독 파악에 그쳐 해당 크리덴셜의 연계 권한 도달 범위를 식별하지 못한 사례입니다.

메커니즘 및 제약 조건별 열거 방법

	메커니즘	열거 방법	주요 누락 항목
G1	인라인 사용자 정책	`iam list-user-policies · get-user-policy`	콘솔 요약 화면 내 미표시
G2	연결된 관리형 정책	`iam list-attached-user-policies`	기본 정책 버전만 평가됨
G3	그룹 소속	`iam list-groups-for-user → 그룹 정책`	인지하지 못한 상속된 관리자 권한
G4	리소스 기반 정책	`accessanalyzer list-findings`	**주체 측 확인 불가. 누락 빈도 최다 항목**
G5	KMS 키 정책과 grant	`kms get-key-policy · kms list-grants`	grant는 생성 정책보다 오래 잔존함
G6	수임 가능 역할 (체이닝 포함)	`get-account-authorization-details 내 신뢰 정책`	깊이 2 이상의 체인은 미추적 위험 상존
C1	권한 경계	`iam get-user → PermissionsBoundary`	설정되어 있으나 범위가 과다하여 제한 효과 미흡
C2	서비스 제어 정책	`organizations describe-effective-policy`	적용 가정한 상태이나 해당 OU에 미적용된 사례 다수

실효 권한 = (G1U…UG6) ∩ C1 ∩ C2. 도달 범위 내 모든 역할을 대상으로 개별 계산이 필요합니다.

AWS · 3 / 4

공격자의 AWS 키 악용 정형 패턴

자동화된 크리덴셜 악용 패턴은 정형화되어 있으므로 아래 표를 타임라인 대조 체크리스트로 활용합니다. 항목에 해당하면 사고 단계가 '노출'에서 '실제 사용'으로 전환되며, 영향 단계 항목은 과금 내역으로 교차 검증합니다.

단계	조회 대상 CLOUDTRAIL EVENTNAME	의미
정찰	`GetCallerIdentity` · `DescribeRegions` · `DescribeInstances` · `ListBuckets` · `ListRoles` · `GetAccountAuthorizationDetails` · `DescribeDBInstances`	최초 공격 단계 (EleKtra-Leak 사례의 첫 API 호출은 `DescribeRegions`였음)
지속성 확보	`CreateAccessKey` · `CreateUser` · `CreateLoginProfile` · `UpdateLoginProfile` · `AttachUserPolicy` · `ImportKeyPair`	공격자가 생성한 2차 크리덴셜. 유출 키 로테이션 후에도 지속 잔존함
권한 상승	`PutUserPolicy` · `AttachRolePolicy` · `UpdateAssumeRolePolicy` · `PassRole` · `AssumeRole`	측정 완료 이후 영향 범위를 변경시킴 (최종 단계에서 재측정 필수)
방어 회피	`StopLogging` · `DeleteTrail` · `UpdateTrail` · `DeleteDetector` · `DeleteLogGroup` · `PutEventSelectors`	해당 항목 발견 시 이후 타임라인 분석에 구조적 한계가 발생하므로 보고서에 명시 필요
크리덴셜 접근	`GetSecretValue` · `GetParameter(s)` · `Decrypt` · `GetPasswordData`	사고 확산 분기점. AWS 기본 격리 정책으로 차단 불가
수집과 반출	`GetObject` DATA EVENT · `CreateDBSnapshot` + `ModifyDBSnapshotAttribute` · `CreateSnapshot` + `ModifySnapshotAttribute` · `PutBucketPolicy`	`Modify…Attribute`조작을 통해 스냅샷을 **외부 계정**과 공유 (데이터 이벤트 로그를 남기지 않는 반출 방식)
영향	`RunInstances` · `InvokeModel` · `InvokeModelWithResponseStream` · `CreateEndpointConfig` · `SendEmail` · `VerifyEmailIdentity` · `ScheduleKeyDeletion`	크립토마이닝, LLMjacking, 스팸, 리소스 파괴 등 (동일 구간 과금 내역과 교차 검증)

핵심 조사 쿼리 2종

ATHENA · 단일 키 전체 타임라인

```
SELECT eventtime, awsregion, eventsource, eventname,
       sourceipaddress, useragent, errorcode, readonly
FROM   cloudtrail_logs
WHERE  useridentity.accesskeyid = 'AKIA...'
  AND  eventtime BETWEEN '2026-06-01T00:00:00Z' AND '2026-08-
21T00:00:00Z'
ORDER BY eventtime;
-- 역할 세션의 경우 아래 필드 기준으로 조회
--  useridentity.sessioncontext.attributes.creationdate
```

ATHENA · 권한 열거 행위 징후

```
SELECT sourceipaddress, useragent,
       count(*)                          AS calls,
       count_if(errorcode IS NOT NULL)   AS denied,
       count(DISTINCT eventname)         AS distinct_actions
FROM   cloudtrail_logs
WHERE  useridentity.accesskeyid = 'AKIA...'
GROUP BY 1, 2 ORDER BY denied DESC;
-- 단일 IP에서 denied 및 distinct_actions 과다 발생 시 공격자의
-- 권한 탐색 행위로 판단 (정상 워크로드는 호출 액션 수가 적고
-- 거부 발생도 드묾)
```

EleKtra-Leak 사례

2023 Cremit

공격 진행 속도

공개 리포지토리에 키가 노출된 뒤 AWS 자동 격리 정책 적용까지 2분이 걸렸습니다. 공격자는 격리 4분 후 정찰을 개시하여 **7분간 400회 이상 API**를 호출했습니다. `DescribeRegions` 정찰 후 보안 그룹 변경과 채굴용 멀티 리전 `RunInstances` 인스턴스 생성으로 이어졌습니다.

TAKEAWAY

2분 이내 자동 방어 조치만으로는 한계가 있습니다. 정찰은 이미 완료되었다는 전제로 타임라인을 수립해야 합니다.

ⓘ 조사자 활동 우선 제외

최종 분석 전 수집 데이터에서 조사자의 출발지 IP, User-Agent, 검증용 `get-caller-identity` 호출 기록을 식별해 제외해야 합니다. 조사자 행위를 공격자 행위로 오인하면 타임라인 전체를 다시 작성해야 합니다.

AWS · 4 / 4

AWS 격리 정책의 범위와 한계

AWS는 공개 영역에서 자사 키 노출을 감지하면 AWSCompromisedKeyQuarantineV3 정책을 적용합니다. 명시된 목적은 "기존 리소스 영향 최소화 및 무단 과금성 활동의 피해 제한"으로 비용 악용 차단에 특화된 정책입니다. 이에 따라 데이터 유출 조사 시 핵심 액션 대부분은 차단되지 않고 허용 상태로 유지됩니다.

버전이 변경될 때마다 AWS가 상정하는 "탈취 키 용도"가 재정의됩니다. **2020년** 컴퓨팅 남용(EC2, Lightsail) 중심에서 **2021년** 파괴 및 재판매(S3 액션 15개 추가)로 전환되었으며, LLMjacking 등장 이후 `bedrock:InvokeModel`이 추가되었습니다. **2024–26년**에는 랜섬웨어 위협 대응으로 `kms:ScheduleKeyDeletion`과 `s3:PutEncryptionConfiguration`이 추가되었습니다. 전 버전을 비교한 결과 한번 추가된 차단 액션이 제거된 사례는 없습니다. S3 읽기를 제외하면 확인된 데이터 접근 액션 21개는 전 버전에서 거부 처리되지 않습니다.

허용 유지 액션 · 사용자 직접 차단 필요

데이터 탈취 공격에 필요한 주요 액션이 허용 상태로 유지됨.

- `sts:AssumeRole`: **역할 체이닝 정상 동작.** 격리는 사용자 계정에만 적용되며 수임 역할에는 미적용
- `secretsmanager:GetSecretValue`: 접근 가능한 시크릿 전체 조회 가능
- `ssm:GetParameter(s)`: `WithDecryption` 지정 시 평문 반환
- `kms:Decrypt`: 키 정책 변경 및 삭제 액션에 한해 거부 적용
- `rds:CreateDBSnapshot`+ `ModifyDBSnapshotAttribute` 및 ec2 대응 액션: 스냅샷 외부 계정 공유를 통해 S3 거부를 우회한 데이터 반출 가능
- DynamoDB 및 RDS 읽기 경로 허용 유지

차단 절차

1. **증거 확보 우선.** `get-access-key-last-used`, 트레일 상태, GuardDuty 탐지, 과금 내역 수집 (조치 개시 시 해당 값 변경됨)
2. AWS 정책 연결 상태 유지. 조치 완료 전 분리 시 AWS 비용 조정 대상 자격에 영향을 줄 수 있음
3. 허점 보완용 자체 Deny 정책 적용. 기본 격리는 최소 조치일 뿐 충분한 통제를 제공하지 못함
4. 키 삭제 대신 비활성화. `iam update-access-key --status Inactive` 적용. 삭제된 키 ID는 로그 대조 불가하며 `last-used` 정보 소멸. 삭제는 조사 종료 후 수행
5. **역할 세션 폐기.** `ASIA…`크리덴셜 또는 수임 역할 포함 시 필수. 사용자 키 비활성화는 활성 세션에 영향 없음
6. 차단 선언 전 지속성 확보 여부 점검. 구간 내 `CreateUser`, `CreateAccessKey`, `CreateLoginProfile`은 로테이션 후에도 잔존하는 2차 크리덴셜임

격리 정책 허점 보완

```
// 동일 사용자 대상 격리 정책과 병행 적용
{ "Version": "2012-10-17", "Statement": [{
  "Sid": "IRCloseQuarantineGaps",
  "Effect": "Deny", "Resource": "*",
  "Action": [
    "sts:AssumeRole",
    "kms:Decrypt",
    "secretsmanager:GetSecretValue",
    "ssm:GetParameter*",
    "rds:CreateDBSnapshot",
    "rds:ModifyDBSnapshotAttribute",
    "ec2:CreateSnapshot",
    "ec2:ModifySnapshotAttribute",
    "ec2:ModifyImageAttribute"
  ] }] }
```

활성 역할 세션 폐기 (ASIA…)

```
# 콘솔: 역할 → 세션 폐기 실행. AWSRevokeOlderSessions 정책 적용으로
# 현재 시각 이전 발급된 모든 세션 거부 (전파 소요 약 30초)
{ "Effect": "Deny",
  "Action": "*", "Resource": "*",
  "Condition": { "DateLessThan": {
    "aws:TokenIssueTime":
      "2026-08-25T09:00:00Z" }}}
# 해당 시각은 차단 시각(T4) 기준으로 변경
# 서비스 연결 역할은 해당 방식으로 폐기 불가
# Identity Center 세션은 IdC에서 폐기 수행
```

GCP · 1 / 3

"시크릿 조회 여부" 기본 답변: 확인 불가

GCP 감사 로그는 기본 설정값이 다른 두 유형으로 구분됩니다. Admin Activity는 기본 활성화(무료, 400일 보관) 상태이지만, Data Access는 기본 **비활성화**(유료, 30일 보관) 상태입니다. 시크릿 조회는 DATA_READ에 해당하므로 조직의 사전 로깅 설정 상태가 증명 가능 범위를 결정합니다.

로그 종류	기본값	보존	기록 내용
Admin Activity _Required bucket	✓ 활성화 (무료)	400일 변경 불가	설정 및 메타데이터 쓰기, IAM 바인딩, 키 생성/삭제. 지속성 확보 흔적 확인 위치
Data Access _Default bucket	✕ 비활성화	30일 1-3,650일 설정 가능	AccessSecretVersion, GetSecret, ListSecrets, GCS 객체 읽기. 미설정 시 시크릿을 읽어도 흔적 미남음
Security Command Center	◑ 요금제별 상이	가변	비정상 IAM 부여, 반출, 크립토마이닝 탐지. (탐지 부재를 사고 부재로 간주 불가)

계정 미지정, 키 기준 조회

감사 로그에 사용 키 정보가 정확히 기록되므로, 동일 서비스 계정의 다른 크리덴셜과 유출 키를 구분할 수 있습니다.

기록 항목 확인 후 위협 헌팅

```
# 1. Secret Manager 대상 Data Access 로깅 활성화 여부 확인
gcloud projects get-iam-policy PROJECT --format=json \
  | jq '.auditConfigs'
#  null → 시크릿 읽기 기록 부재 상태임을 명시 기재

# 2. 해당 키로 인증된 전체 호출 조회
gcloud logging read '
  protoPayload.authenticationInfo.serviceAccountKeyName:"KEY_ID"
  AND timestamp ≥ "2026-06-01T00:00:00Z"' \
  --project=PROJECT --format=json

# 3. 거부 기록 조회 (권한 탐색 행위 징후)
gcloud logging read '
  protoPayload.authenticationInfo.principalEmail="etl@
...gserviceaccount.com"
  AND protoPayload.status.code ≠ 0' --project=PROJECT
```

주요 필드

- authenticationInfo.serviceAccountKeyName: 요청 서명 키
- authenticationInfo.principalEmail: 주체
- authenticationInfo.serviceAccountDelegationInfo: 계정 사칭 증거 및 사칭 경유 체인
- requestMetadata.callerIp · callerSuppliedUserAgent
- authorizationInfo[].granted: 권한별 허용/거부 여부
- status.code: 0 이외 값은 거부 의미

⚠ 사고 대비 추천 설정

사고 대응 중 Data Access 로깅을 뒤늦게 활성화해도 과거 기록은 복구되지 않습니다. Secret Manager DATA_READ를 **조직** 레벨에서 활성화하여 프로젝트 누락을 방지해야 합니다. 비용이 발생하지만, 미설정 시 "확인 불가" 결론만 가능합니다.

Admin Activity만으로 증명 가능한 항목

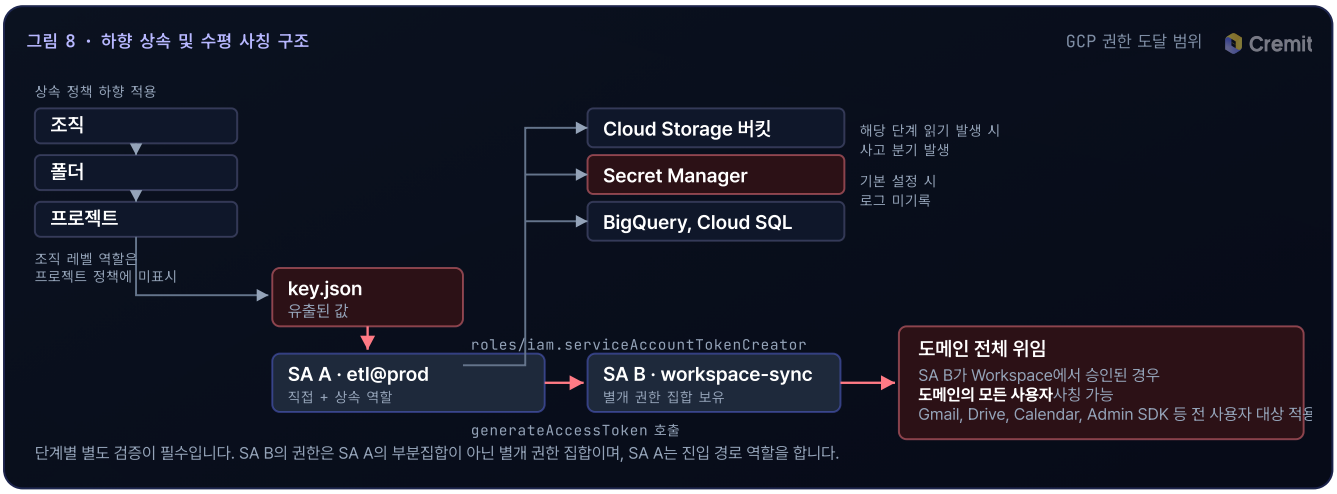
Data Access가 비활성화 상태여도 조사는 진행합니다. 지속성 확보와 권한 상승은 설정 변경에 해당하므로 400일간 무료로 기록됩니다.

공격자 목표	감사 로그 메서드명	해당 항목의 의미
지속성 확보	google.iam.admin.v1.CreateServiceAccountKey	공격자 측 2차 키 생성. 유출 키 삭제 후에도 잔존
지속성 확보	google.iam.admin.v1.CreateServiceAccount	인벤토리 외 신규 주체 생성
권한 상승	SetIamPolicy	policyDelta를 기준선과 비교. 신규 추가된 tokenCreator 바인딩은 계정 사칭 사전 작업임
권한 상승	google.iam.credentials.v1.GenerateAccessToken	계정 사칭 실제 발생 증거. 체인 추적 필수
회피	UpdateSink · DeleteSink	로그 라우팅 변경 확인. 타임라인 공백 발생 사실 기록
영향	compute.instances.insert · aiplatform…predict	크립토마이닝 및 모델 악용. 과금 내역과 교차 확인

GCP · 2 / 3

서비스 계정 키의 상속 및 사칭 경로

GCP 영향 범위는 두 방향으로 확장되며, 단일 주체의 바인딩 조회만으로는 확인할 수 없습니다. **하향**으로는 조직/폴더 상속 정책을 따라 적용되고, **수평**으로는 계정 사칭을 통해 확산됩니다. 체인 내 계정 중 도메인 전체 위임(Domain-wide Delegation)이 설정된 경우 조직 전체 메일함과 Drive 파일로 영향 범위가 확대됩니다.



권한 그래프 끝단 추적

```
# 1. 상속 포함 실효 정책 확인 (프로젝트 레벨 get-iam-policy
#  만으로는 과소평가됨)
gcloud asset analyze-iam-policy --organization=ORG_ID \
  --identity="serviceAccount:etl@prod.iam\
                      .gserviceaccount.com"

# 2. 해당 SA가 사칭 가능한 SA 목록 확인
#  반환된 SA 대상 연쇄 반복 추적하여 도달 범위 산출
gcloud asset analyze-iam-policy --organization=ORG_ID \
  --identity="serviceAccount:etl@..." \
  --permissions="iam.serviceAccounts.getAccessToken"
#  반환된 SA 마다 반복함. 여기까지가 도달 범위임.

# 3. 계정 사칭 실제 발생 여부 검증
gcloud logging read 'protoPayload.methodName=
  "GenerateAccessToken"' --project=PROJECT
```

⚠ 도메인 전체 위임 직접 확인

DWD는 GCP IAM 콘솔에서 확인할 수 없으며, **Workspace 관리 콘솔**(보안 → API 제어 → 도메인 전체 위임)에서 확인할 수 있습니다. `gcloud iam service-accounts describe SA --format='value(uniqueId)'` 명령으로 구한 고유 ID를 해당 목록과 대조합니다. 일치하면 사고 범위가 조직 전체로 확대됩니다.

독립 요소인 API 키

AIza…키는 주체 및 IAM 역할이 없으며 통제 수단은 API/애플리케이션 제한뿐임. 제한 없는 API 키는 프로젝트 내 활성화된 전체 API에 유효한 것으로 간주함.

단일 주체 확장 위험 바인딩

실효 정책에 아래 항목이 있으면 도달 범위가 표면 권한보다 대폭 확대됩니다.

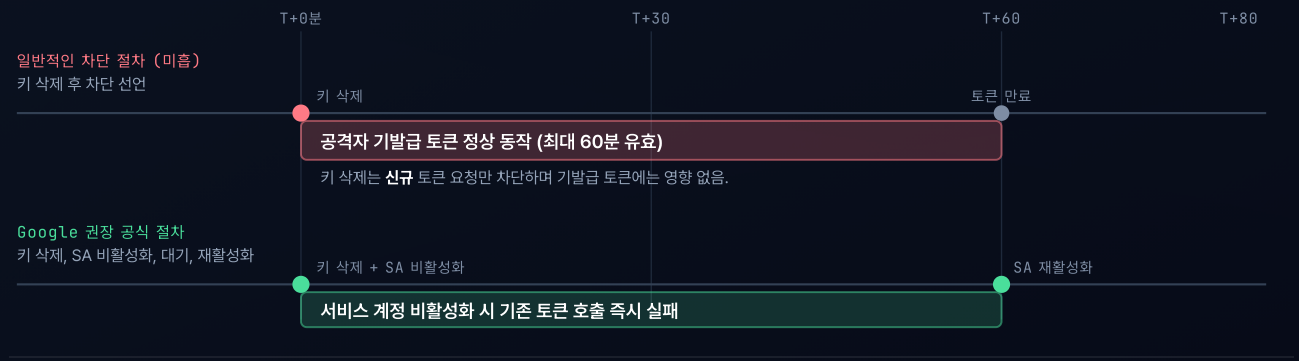
역할 또는 권한	부여 권한	분석 시 의미
`roles/iam.serviceAccountTokenCreator`	타 SA 명의 액세스 토큰 발급	**계정 사칭 핵심 수단. 연계 경로 전수 추적 필요**
`roles/iam.serviceAccountUser`	신규 리소스에 SA 할당	상위 권한 SA 기반 작업 배포 가능
`iam.serviceAccounts.signJwt / signBlob`	SA 명의 어서션 서명	`getAccessToken`로그 없이 토큰 발급 가능
`roles/iam.serviceAccountKeyAdmin`	신규 SA 키 생성	지속성 확보. 유출 키 삭제 후에도 잔존
`roles/resourcemanager.projectIamAdmin`	프로젝트 IAM 정책 수정	프로젝트 내 임의 권한으로 자체 상승 가능
`roles/owner · roles/editor`	프로젝트 리소스 전반 광범위 접근	폴더 상속 후 관리 누락 사례 다수

GCP · 3 / 3

키 삭제 후에도 1시간 유효한 액세스 토큰

GCP 서비스 계정의 기발급 액세스 토큰은 즉시 폐기가 불가능합니다. 키 삭제, 콘솔 조치, `gcloud auth revoke`를 수행해도 이미 발급된 토큰은 1시간 동안 유효 상태로 유지됩니다.

그림 9 · 1시간 시차에 따른 차단 절차 비교



Google 공식 가이드: “키 삭제 시 신규 토큰 발급은 차단되나 기발급 토큰은 무효화되지 않는다. 60분 수명 내 악용을 방지하려면 서비스 계정을 60분간 비활성화해야 한다.”

차단 절차

```
# 0. 증거 보존: 삭제 전 키 목록 확보 (미인지 키는 지속성 흔적임)
gcloud iam service-accounts keys list --iam-account=SA \
  --managed-by=user --format="table(name,validAfterTime,keyOrigin)"

# 1. 신규 토큰 발급 차단 (키 삭제/비활성화)
gcloud iam service-accounts keys delete KEY_ID --iam-account=SA

# 2. 기존 토큰 차단 (서비스 계정 비활성화)
gcloud iam service-accounts disable SA

# 3. 토큰 수명 만료 대기 (60분, 단축 불가)
#   ...60분 대기...

# 4. 서비스 재활성화 및 복구
gcloud iam service-accounts enable SA

# 사용자 gcloud 크리덴셜 유출 시: 관리 콘솔 내
# 연결된 애플리케이션 중 gcloud CLI를 제거하여 토큰 실시간 무효화
```

대기 조치 실효성 검증

```
# T+0 이후 기발급 키 성공 호출 존재 시 비활성화 미적용 상태임
# 해당 구간 전체 status.code ≠ 0 유지 확인 필요
gcloud logging read '
  protoPayload.authenticationInfo.serviceAccountKeyName:"KEY_ID"
  AND timestamp ≥ "T0"' --project=PROJECT \
  --format="value(protoPayload.status.code)" | sort -u
```

⚠ 무효한 대응 조치 3가지

서비스 계정에 `gcloud auth revoke`를 실행하면 로컬 크리덴셜 파일만 삭제되며 토큰은 서버 측에서 유효합니다. Google Cloud 세션 제어는 사용자 계정에만 적용됩니다. 계정 비활성화 없이 키만 로테이션하면 60분간 노출 위험이 유지됩니다.

ⓘ 운영 장애 미허용 시 대안

서비스 계정 1시간 비활성화가 불가능한 경우, 대안으로 해당 계정의 IAM 바인딩을 즉시 제거합니다. 단, IAM 전파 지연(최대 7분)을 고려해야 하며 모든 조치 내용과 시각을 보고서에 명확히 기록해야 합니다.

차단 후 검증 항목

- `keys list`내 해당 키 정상 제거 여부
- 구간 내 미인가 `CreateServiceAccountKey` 존재 여부
- 기존 `serviceAccountKeyName` 서명 호출 차단 여부
- 재활성화 후 계정 사칭 경로 재점검 완료 여부

AZURE · 1 / 3

비인간 주체 로그인 전용 로그 확인

Azure 조사에서 사용자 로그인 로그만 조회하고 사용 흔적이 없다고 판단하는 오류를 주의해야 합니다. 비인간 주체 로그인은 사용자 로그인과 별도 로그(Service Principal, Managed Identity)에 기록되므로 전용 테이블 조회가 필수입니다.

소스	기본값	보존	확인 가능 항목
Entra 서비스 주체 로그인 `AADServicePrincipalSignInLogs`	✓ 활성화	Free 7일 P1/P2 30일	유출 클라이언트 시크릿 분석 1차 소스. 호출 IP, 리소스, 결과 및 사용 크리덴셜 정보 포함
Entra 관리 ID 로그인 `AADManagedIdentitySignInLogs`	✓ 활성화	7 / 30일	별도 테이블. VM 또는 Function 주체 경유 침해 시 확인
Entra 감사 로그	✓ 활성화	7 / 30일	디렉터리 변경, 앱 추가 크리덴셜, 신규 소유자, 동의 부여 등 지속성 흔적 확인
Azure 활동 로그	✓ 활성화	90일	구독 및 리소스 제어 플레인 작업 내역
Key Vault `AuditEvent`	✗ 비활성화	싱크 종속	`SecretGet`, `SecretList` 등. 진단 설정 전 미기록
스토리지 로그 `StorageBlobLogs`	✗ 비활성화	싱크 종속	SAS 사용 확인의 유일한 소스. `AuthenticationType == "SAS"` 필터링 및 `AuthenticationHash` 식별
Defender for Cloud	◑ 요금제별 상이	가변	주체 및 리소스 악용 관련 행위 기반 경보

KQL · 최우선 실행 쿼리

```kql
// 1. 앱 로그인 및 사용 크리덴셜 확인
//   ServicePrincipalCredentialKeyId로 유출 시크릿 식별
AADServicePrincipalSignInLogs
| where AppId == "<appId>"
| project TimeGenerated, IPAddress, ResourceDisplayName,
          ResultType, ServicePrincipalCredentialKeyId
| order by TimeGenerated asc

// 2. Key Vault 내 조회 항목 확인
AzureDiagnostics
| where ResourceProvider == "MICROSOFT.KEYVAULT"
| where OperationName startswith "Secret"
| project TimeGenerated, OperationName, identity_claim_appid_g,
          CallerIPAddress, requestUri_s, httpStatusCode_d

// 3. SAS 토큰별 접근 리소스 확인
StorageBlobLogs
| where AuthenticationType == "SAS"
| summarize ops=count(), bytes=sum(ResponseBodySize),
          first=min(TimeGenerated), last=max(TimeGenerated)
          by AuthenticationHash, CallerIpAddress

// 4. 지속성 검증 (구간 내 추가된 크리덴셜/소유자 점검)
AuditLogs
| where OperationName has_any ("Add service principal credentials",
                               "Add owner to application",
                               "Update application")
| project TimeGenerated, OperationName, InitiatedBy,
          TargetResources
```

⚠ FREE 요금제 보존 기간 한계

Entra ID Free의 로그인 로그 보존 기간은 **7일**입니다. 외부 내보내기(진단 설정)에는 P1/P2 라이선스가 필요합니다. 사고 발생 중 업그레이드해도 과거 로그는 복구되지 않으므로, 7일 초과 건은 확인 불가 상태로 명시해 기재합니다.

ⓘ SAS 생성 로그 부재

계정 키 기반 SAS는 클라이언트 오프라인(HMAC)으로 생성되어 Azure에 생성 이력이 남지 않습니다. 로깅이 사전 활성화된 경우에 한해 SAS 사용 기록만 확인할 수 있습니다.

ResultType코드 해석

실패 코드 분석을 통해 공격 상태 파악 가능.

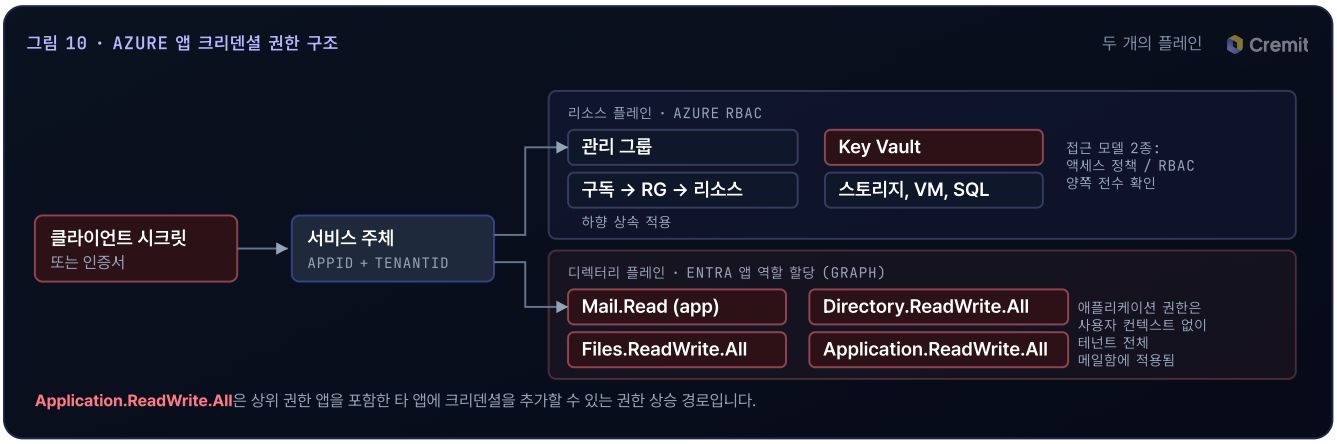
- `0`: 성공 (유효한 크리덴셜)
- `7000215`: 잘못된 클라이언트 시크릿 (구형 또는 추측값 시도)
- `700016`: 디렉터리 내 앱 부재 (유출 시크릿을 오타 테넌트에 재응용 중인 상태)
- `90002`: 테넌트 부재

AZURE

AZURE · 2 / 3

Azure RBAC와 Graph 권한 분리 구조

앱 주체 권한은 독립된 두 체계로 구분됩니다. **Azure RBAC**는 리소스 영역에 적용되고, **Entra 앱 역할 할당**은 Graph를 통해 디렉터리 및 M365 영역에 적용됩니다. 두 영역을 모두 검증하지 않으면 테넌트 전체 영향 범위를 잘못 판단할 위험이 있습니다.



두 플레인 전수 열거

```
# 리소스 플레인: 상속분 포함 전체 스코프 점검
az role assignment list --assignee <appId> --all \
  --include-inherited --include-groups \
  -o table

# 디렉터리 플레인: 부여된 Graph 앱 역할 확인
az ad app permission list-grants --id <appId>
az rest --method get --url "https://graph.microsoft.com/v1.0/
  servicePrincipals(appId='<appId>')/appRoleAssignments"

# 해당 앱에 크리덴셜 추가 가능한 타 주체 점검
az ad app owner list --id <appId>

# Key Vault 적용 접근 모델 확인 및 검증
az keyvault show --name <vault> \
  --query "{rbac:properties.enableRbacAuthorization,
            policies:properties.accessPolicies}"
```

Microsoft AI 사례

2023 Cremit

SAS URL 오설정 사례

GitHub에 공유된 SAS URL의 스코프가 **스토리지 계정 전체**와 **전체 제어** 권한으로 과다 설정되었던 사례입니다.

- 노출 규모: 38TB (직원 워크스테이션 백업 포함)
- 노출 내용: 내부 비밀번호, 시크릿 키, Teams 메시지 3만 여건
- 유효 기간: 2020년 7월 커밋 후 만료일을 **2051년 10월**로 연장 (2023년 6월 제보)

TAKEAWAY

쿼리 문자열에 과다 권한이 명시되어 있었으나 사전 검토가 누락되어 발생했습니다.

Graph 애플리케이션 권한 위험성

사용자 컨텍스트가 없는 권한이므로, 부여 시 영향 범위가 테넌트 전체로 확대됩니다.

권한	실효 도달 범위	보고서 기재 내용
`Application.ReadWrite.All`	모든 앱 등록에 크리덴셜 추가	최상위 권한 앱 포함 권한 상승 위험
`RoleManagement.ReadWrite.Directory`	디렉터리 역할 할당	전역 관리자 권한 상승 경로
`Directory.ReadWrite.All`	디렉터리 전체 읽기/쓰기	사용자, 그룹, 디바이스 전면 통제
`Mail.Read` · `Mail.ReadWrite`	테넌트 내 전체 메일함	조직 전체 메일 접근 (규제 신고 대상)
`Files.ReadWrite.All` · `Sites.FullControl.All`	전체 OneDrive 및 SharePoint	조직 전체 문서 접근 가능
`User.ReadWrite.All`	사용자 객체 수정	인증 관련 주요 속성 변경 가능

AZURE · 3 / 3

SAS 토큰 무효화 절차

앱 크리덴셜은 시크릿 삭제와 주체 비활성화로 즉시 차단되지만, SAS 토큰은 HMAC 오프라인 검증 방식이라 별도 폐기 엔드포인트가 없습니다. 따라서 사용된 3가지 서명 방식(si=, skoid= 존재 여부)에 따라 차단 절차를 다르게 적용합니다.

SAS 종류	식별 방법	무효화 방법	차단 영향 범위
임시 계정 / 서비스 SAS	si=, skoid= 모두 없음	**서명에 사용된 스토리지 계정 키 로테이션**	**해당 키 기반 타 SAS 및 연결 클라이언트 전체 중단**
저장 액세스 정책 연동 서비스 SAS	si=존재	**컨테이너 저장 액세스 정책 삭제 또는 만료일 수정**	**해당 정책 연결 토큰으로 차단 범위 한정**
사용자 위임 SAS	skoid=/ sktid= 존재	**az storage account revoke-delegation-keys**	**사용자 위임 SAS 전체 무효화 (계정 키는 유지)**

애플리케이션 크리덴셜 차단 절차

```
# 1. 등록 크리덴셜 목록 확보 및 보존 (미인지 항목은 지속성 흔적)
az ad app credential list --id <appId> -o table

# 2. keyId 기준 유출 항목 제거
az ad app credential delete --id <appId> \
  --key-id <keyId>

# 3. 사용 확인 시 주체 비활성화
az ad sp update --id <appId> --set accountEnabled=false

# 4. 기발급 토큰 유효기간(60~90분) 대기. CAE 지원 외
#   전체 API 포괄을 전제하지 않음
```

SAS 계정 키 로테이션 절차

```
# 사용처를 key2로 이관 후 key1 로테이션 수행
az storage account keys list -n <acct> -g <rg> -o table
# ...전체 사용처 key2 전환 검증...
az storage account keys renew -n <acct> -g <rg> \
  --key key1

# 저장 액세스 정책인 경우: 해당 정책만 무효화 (무중단)
az storage container policy delete -c <container> \
  -n <policy-id> --account-name <acct>

# 사용자 위임인 경우: 위임 키 철회 실행
az storage account revoke-delegation-keys \
  -n <acct> -g <rg>
```

⚠ 계정 키 로테이션 시 영향도

계정 키를 변경하면 연관된 모든 SAS와 접속 문자열이 즉시 무효화됩니다. SAS 사고 차단을 결정할 때는 운영 장애 발생 가능성을 고려하여 책임자 승인 하에 진행해야 합니다.

차단 검증 항목

- az ad app credential list내 해당 keyId 제거 여부
- T4 이후 해당 ServicePrincipalCredentialKeyId 로그인 차단 여부
- StorageBlobLogs내 AuthenticationHash의 403 수신 여부
- Entra 감사 로그 내 신규 앱 크리덴셜 부재 및 소유자 재확인 완료 여부

계정 키 로테이션 전 사용처 점검

SAS는 서버 측 목록 API가 제공되지 않으므로 아래 항목을 전수 점검하여 사용처를 사전에 확인합니다.

접속 문자열 위치	점검 방법	미조치 시 영향
앱 설정, Key Vault 참조	az webapp config appsettings list	스토리지 호출 시점 오류 발생
Data Factory / Synapse	워크스페이스별 연결된 서비스 정의	배치 파이프라인 실행 실패
쿠버네티스 시크릿, CSI 드라이버	kubectl get secrets -A	파드 재시작 시 크래시 루프 발생
CI/CD 변수, IaC 상태	변수 그룹, Terraform state/tfvars	재배포 시 구 키 재적용 위험
파트너사 및 도구	계약서, Storage Explorer 프로파일 등	외부 연동 장애 발생

공개 사고 보고서가 끝나는 지점 이후

크리덴셜에서 멈추지 않는 실제 침해 체인

공개된 사후 분석은 대개 "키 노출에 따른 로테이션" 수준에서 끝납니다. 본 플레이북의 기반이 된 실제 대응 사례는 그 지점에서 끝나는 경우가 드뭅니다. 아래는 홉(Hop) 단위 추적 경로입니다. IAM 이탈 지점이 핵심이며, 5번 홉부터 공격자는 권한 대신 실행 권한을 확보합니다. 기존 권한 모델로는 이러한 전환을 예측할 수 없습니다.

증폭 지점: 3번 홉

단일 키를 전체 키로 확장하는 호출 2회

```
# 이름 우선 조회. 저비용으로 전체 트리 확인 가능.
aws ssm describe-parameters --query 'Parameters[].Name'

# 증폭 지점: --recursive 옵션으로 접두사 하위 전체 조회
# --with-decryption 옵션으로 SecureString 평문 반환
# /prod/* 단일 권한으로 하위 전체 크리덴셜 확보 가능.
aws ssm get-parameters-by-path --path /prod \
  --recursive --with-decryption

# CloudTrail 관리 이벤트에 해당하여 GCP/Azure와 달리
# 기본 설정에서도 기록됨. 단일 주체의 1분 내 다수 파라미터
# 조회 패턴 탐색.
```

⚠ 권한 모델의 공백

권한 그래프는 크리덴셜의 허용 행위만 계산할 뿐, 유발 가능 행위는 계산하지 못합니다. 빌드 정의 쓰기 권한은 빌드 역할 기반 코드 실행 권한과 같습니다. `iam:*` 권한이 없는 키라도 CI 토큰을 보유하면 해당 파이프라인의 신뢰 수준만큼 실행되며, 이 신뢰 수준은 단일 키 권한을 크게 상회합니다. `simulate-principal-policy`로는 이 영역을 계산할 수 없습니다.

5번 및 6번 홉 해당 API 액션

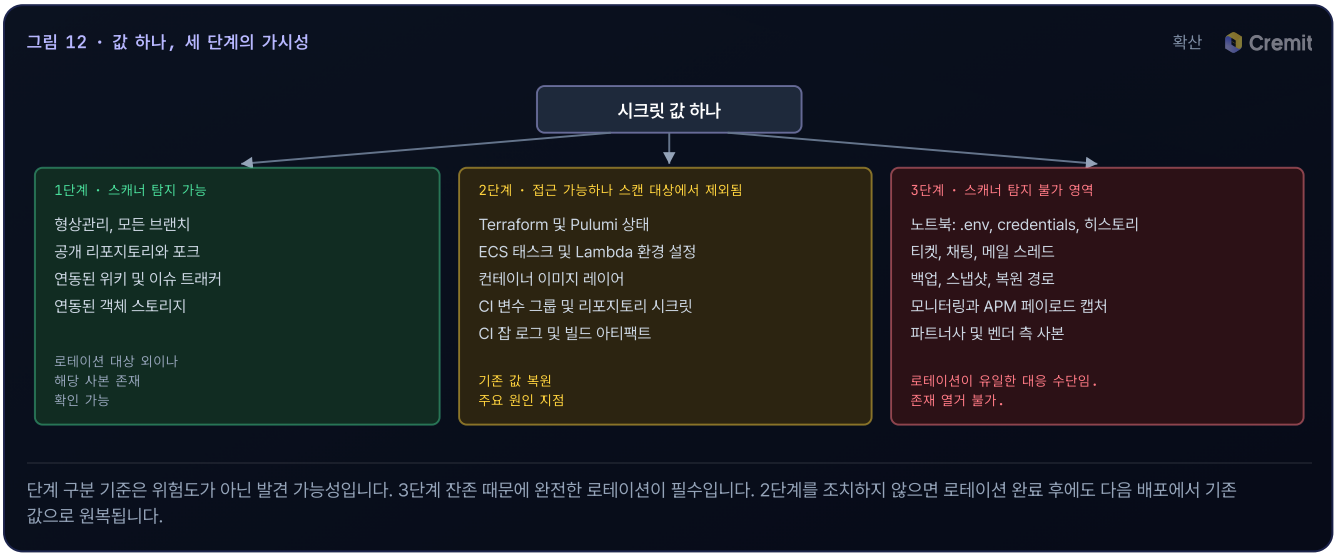
실행 표면	공격자 필요 권한	획득 결과 및 탐지 회피 사유
AWS CodeBuild	`iam:PassRole + codebuild:CreateProject / UpdateProject + StartBuild`	전달받은 역할 기반 임의 명령 실행. 빌드 계정 내 `StartBuild` 이벤트는 일반적 동작으로 분류되어 탐지 어려움
AWS Glue	`iam:PassRole + glue:CreateJob / UpdateJob + StartJobRun`	데이터 플랫폼 내부 코드 실행. 데이터 웨어하우스 크리덴셜 이미 연결됨
Lambda	`iam:PassRole + lambda:CreateFunction / UpdateFunctionCode`	지속적 실행 확보. `UpdateFunctionCode` 실행 시 기존 이름 및 ARN 유지
형상관리	쓰기 권한 토큰 또는 리포지토리 시크릿	차기 푸시 시 빌드 자동 실행. 빌드 개시 전까지 클라우드 API 호출 미발생
CI 서비스	3번 홉에서 탈취한 CI API 토큰	잡 정의, 러너 및 타 프로젝트 시크릿 확보

AWS 격리 정책은 중앙 열 항목 상당수를 거부하지만, 이는 격리 대상 자격 증명에만 해당하며 공격자가 타 주체 권한으로 실행을 전환하면 무력화됩니다.

침해 종료 후에도 잔존하는 것

홉마다 수평 복제되는 시크릿

단일 값의 전진 경로를 추적할 때 식별되지 않는 영역이 복사본입니다. 동일 시크릿 값이 관리 인벤토리에서 누락된 위치로 수평 복제됩니다. 공격자는 사본 1개로도 충분하지만, 조사자는 전수 식별이 필요합니다. 이 비대칭성 때문에 단 1건의 누락도 대응 실패로 이어지며, 로테이션 현황표에 **불명** 항목이 계속 발생합니다.



단계 구분 기준은 위험도가 아닌 발견 가능성입니다. 3단계 잔존 때문에 완전한 로테이션이 필수입니다. 2단계를 조치하지 않으면 로테이션 완료 후에도 다음 배포에서 기존 값으로 원복됩니다.

동일 값의 축적 위치 및 잔존 사유

잔존 위치	유입 경로	로테이션 후 잔존 사유	확인 방법
IaC 상태	수작업이 아닌 `apply` 명령어에 의한 자동 기록	**상태 파일은 소스코드로 분류되지 않아 검색 제외. 차기 `apply` 실행 시 기존 값 재기록 발생**	`terraform state pull`실행 및 State 버킷 ACL 점검
태스크 및 함수 설정	배포 시 환경변수로 주입	**값이 소스코드/이미지가 아닌 정의(Definition) 파일 내 잔존**	`ecs describe-task-definition` · `lambda get-function-configuration`
컨테이너 이미지 레이어	ARG, COPY 또는 빌드 시점 다운로드	**불변 레이어의 배포 완료 상태. 태그 삭제 시에도 이미지 레이어 잔존**	`docker history`조회, 레이어별 파일시스템 Differencing, 레지스트리 보존 정책 점검
CI 변수 및 리포지토리 시크릿	최초 설정 후 미관리 및 장기 방치	**관리자 지정, 만료일 설정, 정기 검토 부재. 프로젝트 간 중복 재사용 다수**	조직·프로젝트·환경 단위 클라우드 API 목록 전수 조회
CI 잡 로그	echo 출력 또는 미마스킹 도구의 로그 출력	**잡 및 파이프라인 삭제 후에도 로그 보존 기간 지속**	잡 로그 내 평문 및 마스킹 형태 전수 검색
개발자 장비	`.env`, `~/.aws/credentials`, 셸 히스토리	**모든 보안 스캐너 감지 범위 및 변경 통제 체계 이탈**	**직접 확인 불가 (해당 한계가 필수 로테이션 수행의 근거)**
티켓, 채팅, 위키	장애 대응 및 신규 온보딩 과정에서의 텍스트 공유	**일반 계정 권한으로 본문 검색 가능 및 만료 기한 부재**	내부 전수 키워드 검색 (Salesloft 침해 사례와 동일 패턴)
백업 및 스냅샷	백업 시점의 볼륨 내 전 데이터 포함	**데이터 복원 시 로테이션 이전의 유출 값이 미인지 상태로 재유입**	백업 자산 및 복원 경로의 인벤토리 등재, 복원 시 동시 복구 항목 사전 검증
모니터링 및 APM	요청 페이로드 또는 시스템 환경 덤프 수집 시 포함	**일반 로그 대비 수집 보존 기간이 장기화되는 경우 다수**	APM 에이전트별 페이로드 및 환경변수 캡처 설정 점검

⚠ 확산 규칙

크리덴셜 영향 범위에는 **해당 값의 모든 사본**이 포함되며, 미인지 사본 때문에 사고 피해가 확대됩니다. Cloudflare 사례는 수천 개 중 4개 누락, Internet Archive 사례는 Zendesk 토큰 1개 누락으로 1개월 내 재침해가 발생했습니다. 두 사례 모두 자산 목록에 등재되지 않은 사본이 원인이었습니다. 잔존 위치 목록 작성은 사고 발생 전에 완료되어야 하며, 장애 대응 중 식별할 수 있는 대상은 이미 파악된 자산으로 한정됩니다.

시크릿 저장소 · 1 / 2

시크릿 저장소 접근 시점에 분기하는 사고

침해 주체가 `GetSecretValue`를 호출한 시점부터 해당 주체가 접근 가능한 모든 시크릿은 개별 독립 사고로 전환됩니다. 이에 따라 영향 범위, 담당자, 로테이션 절차가 분리 적용됩니다. 단일 영향 범위를 확장하는 대신 개별 시크릿 단위로 사고를 분기하는 것이 타당합니다. 핵심 검증 항목인 "실제 조회 여부"는 플랫폼 구조에 따라 확인 가능 여부가 결정됩니다.

시크릿 조회의 기본 기록 여부

본 보고서의 핵심 기준표입니다. 자사 적용 플랫폼의 분류에 따라 "조회 여부 확인 가능성"이 사전에 결정됩니다.

저장소	조회 연산	기본 기록	기록 위치 및 필요 조치
AWS Secrets Manager	`GetSecretValue`	✓ 예	CloudTrail 관리 이벤트에 기록됨. 이벤트 히스토리 90일 보존, Trail 설정 시 장기 보존 (추적 용이)
AWS SSM Parameter Store	`GetParameter(s)`	✓ 예	관리 이벤트에 기록됨. `SecureString` 타입의 경우 `kms:Decrypt` 연계 확인 필요
GCP Secret Manager	`AccessSecretVersion`	× 아니오	`DATA_READ`이벤트에 해당함. 조직 차원의 Data Access 로깅 사전 활성화 필수 (미설정 시 조회 이력 미남음)
Azure Key Vault	`SecretGet`	× 아니오	Log Analytics 진단 설정을 통한 `AuditEvent` 전송 필요 (사전 설정 전 기록 불가)
HashiCorp Vault	`kv read`	× 아니오	Audit Device 활성화 필요. 대상 식별용 HMAC 값은 기록되나 실제 평문 값은 확인 불가
Kubernetes Secrets	`get secret`	× 아니오	API 서버 감사 정책의 `Metadata` 레벨 이상 설정 필수. etcd 백업 접근 권한자 병행 확인
CI/CD 변수	빌드 시 주입	× 아니오	조회 단위 개별 감사 로그 부재. 침해 구간 내 실행된 전체 잡을 영향 범위로 산정 (Codecov/CircleCI 피해 확산 원인)

Salesloft Drift 사례

2025 Cremit

2차 침투가 목표였던 사례

채팅 연동 OAuth 토큰 탈취를 통해 10일간 **700개 이상 조직**의 Salesforce 테넌트가 노출되었습니다. 공격자는 CRM 데이터 확보에 그치지 않고 지원 티켓(Case) 본문을 조회하여 AWS 액세스 키, Snowflake 토큰, VPN 크리덴셜, 계정 비밀번호를 수집했고, 해당 자격 증명으로 내부 환경 2차 침입을 수행했습니다.

TAKEAWAY

티켓 시스템, 위키, 채팅 채널 역시 실질적인 시크릿 저장소입니다. 감사 이력 검증용 `GetSecretValue` 기능만 없을 뿐이며, 공격자는 이를 악용합니다.

시크릿 저장소 · 2 / 2

로테이션 순서 및 완료 검증

로테이션 실패 유형은 크게 두 가지입니다. 첫째, 공격자의 진입 크리덴셜 통제가 미흡한 상태에서 하위 시크릿을 먼저 로테이션하여 신규 발급 값이 재탈취되는 경우입니다. 둘째, 현황 관리표 없이 임의로 완료를 선언하여 일부 잔존 자산이 누락되는 경우입니다.

로테이션 패턴 및 결과

패턴	적용 대상	결과
이중 슬롯	AWS IAM(키 2개), Azure 스토리지(key1/key2), GCP SA 키	기존 값 삭제 전 신규 값 이관을 통한 무중단 적용
유효기간 중첩	대부분의 SaaS. 복수 유효 토큰 허용	이중 슬롯 구조와 동일하며, 신·구 토큰의 동시 추적 관리 필요
단일 값	단일 활성 키만 허용하는 API, 일부 웹훅	전환 작업 일정 수립 및 사전 공지 필수
개별 폐기 불가	임시 Azure SAS, 공유 키로 서명된 모든 항목	마스터 서명 키 로테이션 시 연동된 전체 항목 무효화 처리됨. 연동 사용처 목록 선제 작성 필수

CircleCI 사례

2023 Cremit

"전체 로테이션"이 유일한 정답인 경우

엔지니어 단말의 악성코드를 통해 운영 토큰 발급 권한 계정의 SSO 세션이 탈취되었습니다. 이로 인해 고객 환경변수, 토큰, 키 자산이 외부로 반출되었습니다. 데이터는 암호화 상태였으나 메모리 프로세스에서 키가 추출되었습니다. 이에 따라 전체 고객에게 플랫폼 보관 시크릿 전수 로테이션을 통보하고 2주간의 로그 감사 가이드를 전달했습니다.

TAKEAWAY

저장소 자체가 침해되면 선별적 로테이션은 불가능합니다. 시크릿과 사용 주체 인벤토리를 미리 구축한 조직만이 신속한 복구를 달성합니다.

로테이션 현황표

식별된 시크릿 단위별로 1행씩 작성합니다. 모든 행이 최종 검증 단계에 도달해야 로테이션이 종료됩니다. "미사용 추정" 항목은 인정되지 않습니다.

시크릿	저장소	확인된 사용처	담당	신규 발급	사용처 이관	기존 값 삭제	기존 값 폐기 확인
`prod/db/app`	Secrets Manager	08	platform	✓	8 / 8	✓	401 확인
`stripe/live`	Secrets Manager	02	payments	✓	2 / 2	✓	401 확인
`partner/sftp`	Vault kv/	03	data	✓	2 / 3	–	× 미완
`legacy/etl`	CI 변수	불명	× 미지정	–	–	–	× 미완

하단 2개 행이 실질적인 사고이자 보고 대상입니다. 상단 2개 행은 완료된 관리 이력입니다.

기존 값 무효화 기록 생성

```
# 지정된 호스트에서 기존 값으로 1회 인증 호출 수행.
# 발생한 거부 로그를 검증 증적 자료로 사용.
aws sts get-caller-identity    # → InvalidClientTokenId
curl -so /dev/null -w '%{http_code}' \
  -H "Authorization: Bearer $OLD" "$URL"
# 401 ← 반환 응답 및 UTC 시각을 현황표에 기록
```

① 단일 거부 반응 확인의 한계

클라우드 내 캐시 토큰이 유효한 기간에는 원본 자격 증명이 무효화되더라도 세션 자체는 유지됩니다(AWS 역할 세션, GCP/Azure 액세스 토큰 등). 현황표 최종 단계 검증 시 기존 자격 증명과 발급된 세션 전체의 무효화 여부를 동시에 확인해야 합니다.

SaaS 키

SaaS 벤더 확인 질문

SaaS 자격 증명은 CloudTrail 같은 중앙 로깅 체계가 없어 영향 범위 산정이 벤더 제공 기능에 의존합니다. 주요 20개 연동 서비스를 대상으로 다음 4가지 검증 항목을 적용한 결과가 현실적인 조사 한계 범위입니다.

Q1

Q1: 미사용 상태에서의 스코프(Scope) 조회 가능 여부

미지원 시 해당 연동에 정의된 최대 권한을 전제로 영향 범위를 산정해야 합니다.

Q2

Q2: 출발지 IP가 포함된 요청 단위 로그 제공 여부

누적 사용량 지표는 이상 유무만 판별할 수 있으며, 구체적인 행위 주체와 위치는 요청 단위 로그(Request-level Log)로만 검증할 수 있습니다.

Q3

Q3: 단일 토큰의 개별 폐기 기능 지원 여부

동일 서명 키로 연동된 자격 증명이 연쇄 무효화되는 경우, 폐기 작업은 시스템 가용성에 영향을 주는 운영 이벤트로 분류됩니다.

Q4

Q4: 폐기 명령 적용 완료까지의 소요 시간

캐시 토큰 보존, 엣지 서버 전파 지연, 갱신 주기 등은 공격자의 서비스 내 체류 시간을 연장시키는 요인입니다.

크리덴셜	Q1: 스코프 확인	Q2: 요청 단위 로그	Q3/Q4: 폐기
GitHub classic PAT `ghp_…`	가능 (`X-OAuth-Scopes` 응답 헤더 내 명시)	조직 감사 로그 활용 (출발지 IP는 해당 요금제의 IP 로깅 옵션 활성화 시 제공)	토큰 삭제 시 즉시 무효화 (타 토큰 영향 없음)
GitHub fine-grained PAT `github_pat_…`	불가 (조회 엔드포인트 부재. `X-Accepted-GitHub-Permissions`는 해당 엔드포인트 요구 권한만 표시)	상동	토큰 삭제
GitHub App 설치 토큰 `ghs_…`	설치된 권한의 전수 열거 가능	조직 감사 로그	토큰 유효기간 약 1시간, 수동 폐기 지원
Slack 봇 / 사용자 토큰 `xoxb-` · `xoxp-`	`auth.test`를 통한 주체 확인, Web API를 통한 할당 스코프 조회	감사 로그 API는 Enterprise Grid 플랜 전용 (하위 요금제는 로그 조회 불가)	단일 토큰 폐기 또는 앱 자격 증명 로테이션 수행
Stripe `sk_live_` · `rk_live_`	Restricted Key 내 설정된 자체 범위 확인 가능	대시보드 상 출발지 IP 포함 요청 로그 전수 제공 (요청 단위 검증 가능)	키 로테이션 수행 (전환용 유예 기간 설정 옵션 지원)
모델 벤더 API 키 `sk-` · `sk-ant-`	프로젝트 또는 워크스페이스 단위 스코프 확인	누적 사용량 및 비용 지표만 수집됨 (비정상 과금 발생이 유일한 이상 탐지 지표)	키 삭제 시 즉시 적용
패키지 레지스트리 토큰 `npm_` · `pypi-`	세분화 토큰(Fine-grained)만 범위 확인 가능 (레거시 토큰은 미지원)	배포 이벤트 추적 가능, 단순 읽기(Read) 이력은 확인 불가	토큰 폐기 및 해당 기간 내 배포 건 전수를 공급망 침해 사고로 분류

xAI 사례

2025 Cremit

담당자 없는 탐지

내부 임직원의 공개 리포지토리에서 `x.ai` API 키 노출이 발생했습니다. **3월 2일** 보안 스캐닝 벤더가 개발자 개인에게 알렸으나 **4월 30일**까지 미조치 상태로 방치되었고, 보안팀 에스컬레이션 후에야 폐기가 완료되었습니다. 해당 키는 내부 데이터 학습 모델 등 약 60개 AI 모델에 대한 접근 권한을 보유하고 있었습니다.

TAKEAWAY

자동 탐지 체계는 정상 작동했으나 조치 프로세스가 없었습니다. 탐지 알림은 개별 작성자가 아닌 조치 SLA를 이행하는 전담 팀으로 전달되어야 합니다.

Shai-Hulud 사례

2025 Cremit

방어자의 스캐너가 곧 공격자의 도구

자기복제형 npm 웜이 메인테이너 토큰을 탈취한 뒤, 빌드 호스트에서 시크릿 스캐너인 `TruffleHog`를 실행하여 클라우드(AWS·GCP·Azure) 크리덴셜을 수집하고 타 패키지로 확산되었습니다. 1차로 500여 개 패키지 버전, 2차로 주간 다운로드 2,000만 회 이상의 패키지 796개에 전파되었습니다.

TAKEAWAY

보안 대응의 비대칭성은 점검 주기 격차에서 비롯됩니다. 공격자의 상시 스캔에 맞춰 방어자의 정기 스캔 간격을 단축하는 조치만으로도 다수의 침해 사고를 예방할 수 있습니다.

확보 우선순위

휘발성 높은 순서의 증거 확보

핵심 증적 자료일수록 휘발성이 높으며, 일부는 차단 조치 실행 과정에서 손실됩니다. 제시된 순서에 따라 선제 수집을 진행합니다. 특히 1단계 항목은 1시간 이내에 수집하지 못하면 다시 확보할 수 없습니다.

단계	산출물	확인 가능 항목	확보 방법 및 소멸 사유
1	**IAM 키 last-used** AWS	해당 키의 확인된 최종 사용 이력	`iam get-access-key-last-used` **검증 목적 직접 접속 시 기록 덮어쓰기 발생**
1	**노출 산출물**	유출 경로 및 시작 시점	리포지토리 클론 및 원본 페이지 저장 (URL/UTC 시각 포함 캡처). 소유자의 즉시 삭제 또는 비공개 전환 위험
1	**Unreachable Git 객체**	삭제된 커밋의 과거 공개 여부	`git fsck --unreachable --dangling` **리포지토리 삭제 또는 GC(Garbage Collection) 실행 시 소멸**
1	**Entra 로그인 로그** AZURE	서비스 주체 로그인 여부 및 출발지	`AADServicePrincipalSignInLogs` **Free 플랜 7일 보존 제한 (라이선스 업그레이드 시에도 소급 적용 불가)**
1	**CI 잡 로그**	출력된 시크릿 및 해당 시크릿을 사용한 잡	대상 구간 잡 로그 아카이브 추출. 보존 기간 30일 이하 기본 설정으로 인한 순차 소멸
1	**SaaS 사용량 및 과금 화면**	이상 사용량. 유일한 신호인 경우 다수	로테이션에 의한 컨텍스트 초기화 전 일별 사용량 그래프 캡처
1	**Data Access 로그** GCP	시크릿 조회 여부	`gcloud logging read` **_Default 버킷 30일 보존, 사전 설정 필수**
2	**권한 스냅샷** AWS GCP AZURE	당시의 영향 범위	`get-account-authorization-details · analyze-iam-policy · role assignment list` **차단 조치 수행 시 권한 상태 변경 발생**
2	**CloudTrail 구간** AWS	해당 키로 수행된 모든 호출	`cloudtrail lookup-events --lookup-attributes AttributeKey=AccessKeyId` **90일간 전 리전 순회 조회 필요**
2	**GuardDuty / SCC / Defender**	행위 기반 교차 확인	`guardduty list-findings` **약 90일**
2	**Key Vault 감사** AZURE	시크릿 조회 여부	`AzureDiagnostics … MICROSOFT.KEYVAULT` **사전 진단 설정 활성화 시에만 추출 가능**
2	**스토리지 / SAS 사용** AZURE	SAS 토큰별 접근 대상 및 데이터 유출량 식별	`StorageBlobLogs … AuthenticationType == "SAS"` **AuthenticationHash 교차 검증**
2	**크리덴셜 인벤토리**	공격자에 의한 지속성(Persistence) 백도어 생성 여부	`iam get-credential-report · sa keys list · az ad app credential list` **정상 변경 이력과의 대조 작업 수행**
2	**SaaS 감사 로그**	복제, 반출, 권한 변경	조직 감사 로그 API 활용. 플랫폼별 플랜 제약 존재 (Slack의 경우 Enterprise Grid 필수)
3	**CloudTrail S3 트레일** AWS	장기 구간 타임라인	Athena 쿼리 조회 또는 CloudTrail Lake 적재. 영구 보존 항목으로 확보 여유 있음
3	**Admin Activity** GCP	지속성 확보 및 권한 상승	`_Required` 버킷 **400일 기본 보존 (상시 활성화)**
3	**Azure 활동 로그** AZURE	컨트롤 플레인(Control Plane) 변경 이력	`az monitor activity-log list` **90일**
3	**레지스트리와 이미지 히스토리**	해당 시크릿의 산출물 포함 배포 여부	`docker history`조회, 레이어 diff 및 레지스트리 Pull 로그 분석

⚠ 차단 조치 전 권한 스냅샷 수집

다른 항목과 달리 본 항목은 대응 과정에서 조사자에 의해 삭제됩니다. 정책 분리, 주체 비활성화, 바인딩 삭제를 수행하면 해당 자격 증명의 접근 가능 범위 증적 자료가 함께 소멸합니다. 규제 기관 및 경영진 보고용 인가 상태 기록을 사전에 파일로 저장해야 합니다.

복사, 변수 입력, 실행

단일 케이스 폴더로 완료되는 수집 런북

수집 순서를 반드시 준수합니다. 선행 명령어로 차후 명령어가 삭제할 데이터를 먼저 확보합니다. 본 작업은 침해된 자격 증명이 아닌 전용 조사자 계정을 사용하여 로깅이 보장된 단말에서 수행해야 합니다.

1 · 준비, AWS 주체와 권한

```
#!/usr/bin/env bash. 유출 계정이 아닌 전용 분석 계정으로 실행.
set -euo pipefail
CASE="case-$(date -u +%Y%m%dT%H%M%SZ)"
FROM="2026-06-01T00:00:00Z"          # 노출 기간 시작
KEY="AKIA..."                         # 유출된 식별자
mkdir -p "$CASE"/{aws,gcp,azure,git,saas,notes}
exec > >(tee -a "$CASE/notes/collection.log") 2>&1
echo "op=$(whoami) host=$(hostname) egress=$(curl -s ifconfig.me)" | tee "$CASE/notes/chain-of-custody.txt"
# 1. 최우선 수집. 대상 키 기반 인증 발생 시 기존 기록 덮어쓰기 소멸.
aws iam get-access-key-last-used --access-key-id "$KEY" > "$CASE/aws/last-used.json"
aws sts get-access-key-info --access-key-id "$KEY" > "$CASE/aws/owner.json"
# 2. 권한 스냅샷 수집. 차단 조치 전 실행 필수.
aws iam get-account-authorization-details --filter User Role Group LocalManagedPolicy > "$CASE/aws/authz.json"
aws iam generate-credential-report >/dev/null
aws iam get-credential-report --query Content --output text | base64 -d > "$CASE/aws/cred-report.csv"
```

2 · 전 리전 트레일과 GCP

```
aws cloudtrail describe-trails > "$CASE/aws/trails.json"
for r in $(aws ec2 describe-regions --query 'Regions[].RegionName' --output text); do
  aws cloudtrail lookup-events --region "$r" --start-time "$FROM" \
    --lookup-attributes AttributeKey=AccessKeyId,AttributeValue="$KEY" > "$CASE/aws/ct-$r.json" || true
done
P="prod-analytics-01"; ORG="<org-id>"; SA="etl@$P.iam.gserviceaccount.com"; KEYID="a1b2c3..."
gcloud asset analyze-iam-policy --organization=$ORG --identity=serviceAccount:$SA --format=json > "$CASE/gcp/iam.json"
gcloud iam service-accounts keys list --iam-account="$SA" --managed-by=user --format=json > "$CASE/gcp/sa-keys.json"
# 해당 키로 서명된 전체 API 호출 조회.
Q="protoPayload.authenticationInfo.serviceAccountKeyName:\"$KEYID\""
gcloud logging read "$Q AND timestamp≥\"$FROM\"" --project="$P" --format=json > "$CASE/gcp/key-usage.json"
```

3 · AZURE와 GIT, 마지막으로 폴더 봉인

```
APPID="0000...0000"; WS="<workspace-id>"
az ad app credential list --id "$APPID" -o json > "$CASE/azure/app-credentials.json"
az role assignment list --assignee "$APPID" --all --include-inherited \
  --include-groups -o json > "$CASE/azure/rbac.json"
G="https://graph.microsoft.com/v1.0"; SP="servicePrincipals(appId='$APPID')"
az rest --method get --url "$G/$SP/appRoleAssignments" > "$CASE/azure/graph-roles.json"
# 로그인 이력: Azure KQL 실행 결과를 지정 디렉토리로 내보내기.
# Git 자산: 전체 히스토리 및 Unreachable 객체 추출.
git log --all --format='%H %aI %cI %ae' -S"$KEY" > "$CASE/git/hits.txt"
git fsck --unreachable --dangling --no-reflogs > "$CASE/git/dangling.txt"
# 데이터 봉인: 개별 파일 해시 생성 후 최종 아카이브 해시 검증.
find "$CASE" -type f -exec shasum -a 256 {} \; | sort -k2 > "$CASE/MANIFEST.sha256"
tar -czf "$CASE.tar.gz" "$CASE" && shasum -a 256 "$CASE.tar.gz" | tee "$CASE.tar.gz.sha256"
```

별도 기록 항목

- 발견자, 발견 방법 및 정확한 UTC 시각
- 조사자 실행 검증 호출 내역: 명령어, 실행 시각, 출발지 IP, User-Agent
- 의사결정 분기별 선택된 차단 방안 및 해당 승인자
- 미확보 소스 및 해당 사유 (기록된 누락은 분석 결과로 인정되나, 미기록 누락은 관리 책임 사유로 분류됨)

ⓘ 사전 스크립트 최적화 필수

본 수집 스크립트는 템플릿 형태로 제공됩니다. 리전 순회 시 API 호출 비용이 발생하고 `lookup-events` 실행 시 속도 제한(Rate Limit)이 적용될 수 있으므로 평시에 사전 모의 테스트가 필요합니다. 클라우드 제어 플레인 외의 KQL 결과, GuardDuty, CI/SaaS 로그, 레지스트리 이력 등은 수동 수집 항목입니다.

개념이 아닌 정확한 문자열

답이 실제로 담긴 로그 필드 경로

CloudTrail이 전체 호출을 기록한다는 사실과 JSON 객체 내 공격자 계정 식별자(Account ID)의 위치를 파악하는 것은 구분되어야 합니다.

AWS CloudTrail

필드 경로	확정 가능 항목
`userIdentity.accessKeyId`	해당 API 호출 서명에 사용된 AKIA / ASIA 키
`…sessionContext.attributes.creationDate`	임시 세션 생성 시각 (`aws:TokenIssueTime` 대조 기준값)
`userIdentity.invokedBy`	AWS 내부 서비스에 의한 호출 시에만 생성 (외부 공격자 트래픽에는 미포함)
`requestParameters.roleArn` `responseElements.assumedRoleUser.arn`	`AssumeRole`수행 시: 차기 이동 홉 및 생성 세션 식별
`requestParameters.secretId`	`GetSecretValue`수행 시: 실제 조회 대상 시크릿 식별
`requestParameters.withDecryption`	`GetParameter`수행 시: 평문 복호화 반환 여부
`…createVolumePermission.add.items[].userId · .group`	**`ModifySnapshotAttribute`수행 시: 스냅샷 공유 대상 AWS 계정 ID (전체 공개 시 `all`)**
`SharedSnapshotCopyInitiated` `SharedSnapshotVolumeCreated`	**외부 공격자의 스냅샷 사본 반출 시 자사 계정에 기록 (데이터 유출 확정 증적)**
`errorCode`	접근 거부 이력 (권한 탐색 행위의 징후)

GCP Cloud Audit Logs

필드 경로 (`PROTOPAYLOAD` 하위)	확정 가능 항목
`authenticationInfo.serviceAccountKeyName`	인증 서명에 사용된 서비스 계정 키 ID
`….serviceAccountDelegationInfo[]` `….principalEmail`	**계정 가장(Impersonation) 체인 순서 및 최종 인증 주체**
`authorizationInfo[].permission · .granted`	개별 권한 단위 승인/거부 여부
`status.code`	0 이외 값은 요청 거부 상태 의미
`methodName`	예: `…SecretManagerService.AccessSecretVersion`
`requestMetadata.callerIp` `.callerSuppliedUserAgent`	요청 출발지 (명령행 CLI 및 라이브러리 간 기록 형식 차이 존재)

자주 누락되는 명령 2종

VAULT · 보이지 않는 시크릿 확인

```
# Audit Device의 전체 값 HMAC 처리로 평문 검색 불가.
# 유출 값의 HMAC을 Vault에서 조회 후 해당 해시값으로 로그 검색.
vault write sys/audit-hash/<device> \
  input="<leaked value>"
# hash: hmac-sha256:9f2c... → 해당 해시값 기반 감사 로그 검색
```

쿠버네티스 · 한 줄로 확인하는 영향 범위

```
kubectl auth can-i --list \
  --as=system:serviceaccount:<ns>:<sa>
```

AZURE SAS 쿼리 문자열 해독

파라미터별 구성을 바탕으로 접근 범위, 만료 시점, 그리고 마스터 키 로테이션 없는 개별 폐기 가능 여부를 판단합니다.

파라미터	의미	도출 결론
`ss`	서명된 서비스	계정 SAS 전용 (`b q t f` = Blob, Queue, Table, File)
`srt`	서명된 리소스 타입	**Service, Container, object. sco 설정 시 계정 전체 접근 허용 (서비스 SAS는 `sr` 사용: blob, container, directory)**
`sp`	권한	**`rwdlacupx` 설정 시 데이터 삭제 포함 전체 권한 부여**
`st · se`	시작 및 만료	**임시 SAS의 유일한 통제 수단 (만료 여부 판별 시 연도 단위 점검 필수)**
`sip · spr`	IP 범위 및 프로토콜	`sip`미설정 시 임의 위치 접속 허용 (설정 시 조사 대상 IP 축소)
`si`	저장된 액세스 정책 id	**해당 정책 삭제만으로 즉시 폐기 가능 (마스터 키 로테이션 불필요)**
`sk*`	사용자 위임 키 필드 (`skoid sktid skt ske sks skv`)	**az storage account revoke-delegation-keys 적용으로 즉시 무효화 (계정 키 유지)**
`sig`	HMAC 값 자체	`StorageBlobLogs`내 `AuthenticationHash` 대조용 사용

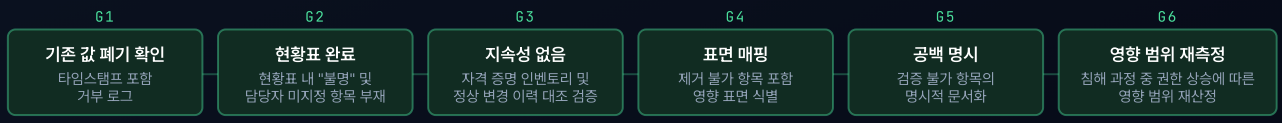
`si`와 `skoid`가 모두 없으면 계정 마스터 키 로테이션이 유일한 폐기 수단이며, 해당 키로 발급된 전체 토큰이 연쇄 무효화됩니다.

종료 판정 기준

사고 종료를 증명하는 6개 게이트

침해사고의 재발은 신규 공격이 아닌 최초 사고의 불완전한 조기 종료에서 기인합니다. 아래 6개 게이트별로 감사 제출 수준의 증적 산출물 확보가 필수이며, 증적이 미비한 게이트는 통과할 수 없습니다.

그림 15 · 6개 게이트, 전수 통과 필요 — 종료 Cremit



게이트 1개만 실패해도 의사결정 트리 04단계로 복귀합니다(부분 종료 불인정).

Cloudflare 사례

2023 Cremit

게이트 2, 4개 항목 검증 실패

Okta 침해 사고 이후 수천 개의 자격 증명을 로테이션했으나 4개 항목(서비스 토큰 1개, 서비스 계정 3개)이 누락되었습니다. 사유는 "미사용 자산으로 오판"이었습니다.

IdP 침해 발생일은 **10월 18일**입니다. 공격자는 **11월 14일** 잔존 크리덴셜 4개로 탐색을 시작했고, **11월 23일** 최초 탐지되었으나 이미 Atlassian, Jira, 형상관리 시스템까지 권한이 확대된 상태였습니다.

TAKEAWAY

불완전한 사고 종료와 2차 침입 사이의 경과 시간은 27일입니다. "미사용 추정"은 가설에 불과하며, 현황표에는 실제 관측 결과만 기록해야 합니다.

Internet Archive 사례

2024 Cremit

동일 게이트, 1개월 내 2회 연속 실패

인증 토큰이 포함된 GitLab 설정 파일이 방치되어 **3,100만 건**의 계정 정보가 유출되었습니다. 수주 후 1차 사고 당시 노출되었으나 로테이션에서 누락된 Zendesk 토큰을 통해 2차 침입이 발생했고, 고객 지원 티켓 80만 건 이상이 노출되었습니다.

TAKEAWAY

2차 침입은 예견된 결과이며, 최초 사고의 게이트 2 종결 지연이 주요 원인입니다.

보고 필수 지표 5개

지표	정의	측정 사유
노출 기간	T1 → T4	침해 발생 가정 구간 한정
탐지까지의 시간	T1 → T3	실질적 모니터링 개선 대상 구간 (기타 구간은 프로세스 개선 영역)
차단까지의 시간	T3 → T4	장애 대응 런북 완성도 측정 지표
공격자 체류	T2 → T4	공격자 내부 체류 시간 측정 (T2 미확정 시 추정 구간 표기, 0 입력 불가)
로테이션 완료율	로테이션 / 범위	**100% 미달 시 사고 조치 진행 중 상태 유지 (티켓 마감 여부 무관)**

⚠ 보고서 작성 표준 지침

"악용 흔적 미발견"과 "미악용 증명"은 완전히 별개의 결론입니다. 실제 수집되는 증적은 전자에 한합니다. Data Access 로깅 미설정, 로그 보존 기간 경과, 스토리지 로깅 부재 등의 한계 사항은 부록이 아닌 경영진 요약(Executive Summary)에 명시해야 합니다. 기록하지 않으면 비기술 경영진은 이를 '악용이 없었음'으로 오해할 위험이 큽니다.

경영진 요약 · 기재 순서

경영진 요약은 다음 6개 핵심 요약문으로 작성하여, 단독으로 읽어도 정확한 상황 판단이 가능하도록 구성합니다.

1. 노출 대상: 크리덴셜 유형, 대상 주체, 노출 일자 및 시간 구간 명시
2. 접근 가능 범위: IAM 정책명이 아닌 실제 영향 받는 데이터 자산 및 시스템 명시
3. 실제 오남용 검증: 로그 분석을 통해 확정된 행위 및 기술적 한계로 확인 불가한 공백 구분 기재
4. 대응 조치 이력: 계정 차단/로테이션 결과, 가용성 보장을 위해 유예된 항목 및 최종 승인자 명시
5. 법적/규제 신고 검증: 관련 법령 검토 결과, 신고 의무 판단 결론 및 이행 기한 기재
6. 재발 방지 개선안: 식별된 로깅/관제 공백 개선 과제, 지정 담당자 및 조치 예정일 적시

자명하지 않은 모든 기재 내용의 근거

본문 기재 내용의 1차 출처 목록

2026년 8월 25일기준 정보입니다. 이후 변경 가능성이 가장 높은 항목은 벤더 측 동작이므로, 차단 명령 수행 직전 각 벤더의 최신 공식 문서를 재확인하시기 바랍니다. 본 부록에 명시되지 않은 내용은 본문에도 포함되지 않습니다.

AWS

- 격리 정책 deny 목록 및 명시 목적 · AWS 관리형 정책 레퍼런스
- 조치 완료 시까지 정책 연결 상태 유지 · AWS re:Post
- IAM 고유 ID 접두사 · IAM 식별자 레퍼런스
- 세션 폐기, `aws:TokenIssueTime`, 전파, 서비스 연결 역할 · IAM 역할 임시 크리덴셜 폐기
- 이벤트 히스토리 90일, 관리 이벤트, 리전별 · CloudTrail 이벤트 히스토리
- `GetSecretValue` 기본 기록 · Secrets Manager 이벤트 로깅
- `GetParameter`는 읽기 전용 관리 이벤트 · 관리 이벤트 로깅
- `get-parameters-by-path` 플래그 · AWS CLI 레퍼런스
- 스냅샷 공유를 이용한 반출 및 피해자 측 이벤트 · Datadog Security Labs, Stratus Red Team, Elastic

GCP, Azure 및 기타

- SA 액세스 토큰 폐기 불가에 따른 60분간 계정 비활성화 · Google Cloud, 침해된 gcloud OAuth 토큰 대응
- `AccessSecretVersion`은 `DATA_READ` 항목, 기본값 비활성화 · Secret Manager 감사 로깅
- 보존 기간 400일 고정, 최소 30일 설정 가능 · Cloud Logging 할당량
- Entra 로그인 보존 정책 및 업그레이드 비소급 적용 · Microsoft Entra 데이터 보존
- 진단 미설정 시 Key Vault 이력 미기록 · Key Vault 로깅 활성화
- SAS 개별 폐기 불가 및 생성 이력 미추적 · MSRC 권고, Wiz 리서치
- SAS 파라미터별 보안적 함의 · 서비스 SAS 생성
- fine-grained PAT 조회 수단 부재 · GitHub REST 문서, 커뮤니티 논의 156115
- GitHub 토큰 말단의 CRC32 체크섬 구조 · GitHub Engineering
- force-push된 커밋의 지속적 조회 가능성 · Truffle Security, Neodyme
- `sys/audit-hash`를 통한 로그 내 HMAC 반환 · HashiCorp Vault
- `kubectl auth can-i --list --as=` · 쿠버네티스 인가

침해사고 사례

- **Uber** 2016 · Breaches.cloud, NBC News
- **Codecov** 2021 · Codecov 사후 분석, GitGuardian
- **Heroku / Travis** 2022 · GitHub 보안 블로그
- **Toyota** 2022 · BleepingComputer, The Register
- **CircleCI** 2023 · CircleCI 사고 보고서, TechCrunch
- **Microsoft AI** 2023 · Wiz 리서치, MSRC
- **EleKtra-Leak** 2023 · Unit 42
- **Cloudflare** 2023 · Cloudflare 사고 보고서
- **Mercedes-Benz** 2024 · RedHunt Labs, BleepingComputer
- **Sisense** 2024 · Krebs on Security, TechCrunch
- **Internet Archive** 2024 · BleepingComputer, Security Affairs
- **LLMjacking** 2024 · Sysdig
- **xAI** 2025 · Krebs on Security
- **Salesloft Drift** 2025 · AppOmni, Anomali
- **Shai-Hulud** 2025 · Unit 42, Datadog Security Labs

벤더 발표 제외 자체 분석 내용

- **발견 확률 표 및 곡선.** 일정 비율로 무작위 발생하는 사건이 특정 구간 내 1회 이상 발생할 확률로, 본문에 명시한 공식을 적용하여 산출했습니다. 4개 비율은 실제 측정치가 아닌 선정값이며 본문에도 동일하게 명시했습니다.
- deny 목록 28 → 89 → 99 (제거된 액션 없음, S3 읽기를 제외한 데이터 접근 액션은 전 버전에서 미거부). 공개된 3개 정책 문서를 직접 비교 분석한 결과입니다. AWS의 문서 갱신 주기에 따라 버전별 재확인이 필요합니다.
- 격리 정책 허용 액션. V3 JSON 데이터를 직접 확인한 결과이며, AWS 공식 발표 내용이 아닙니다.
- 액세스 키 ID 내 계정 ID 추출. 분석 방법은 Aidan Steele, Tal Be'ery, Truffle Security 사례를 인용했습니다. 발행 전 직접 구현·실행하여 검증했으므로 본문 예시는 실제 출력값입니다.
- **7홉 체인 및 확산 단계.** 실제 관측 패턴을 바탕으로 구성했으며, 개별 수단은 공개 문서에 근거하고 연계 순서 자체가 1차 분석 자료입니다.

대응 시간 및 신고 기한

- 노출 키 최초 악용까지 약 1분 소요 · Comparitech 허니팟
- 2분 내 격리, 4분 후 공격자 활동 개시, 7분간 400회 이상 접근 · Unit 42
- GDPR 제33조 · Regulation (EU) 2016/679 · NIS2 제23조 · Directive (EU) 2022/2555 · SEC Item 1.05 · 2023 공시 규정 · 정보통신망법 제48조의3 · 침해사고 신고 · 개인정보 보호법 제34조 · 유출 통지·신고 규정

ⓘ 본 문서에서 제외한 주장

본 문서에 수록된 특정 조직의 부주의를 단정하지 않습니다. 각 사례는 업계 전반의 구조적 실패 양상을 보여주는 예시로만 인용했습니다. **"리포지토리의 X%에 시크릿 존재" 등 시장 집계 통계는 사용하지 않습니다.** 벤더 텔레메트리는 재현이 어렵고 시의성이 낮으므로, 실명 사고와 1차 문서에 근거하여 논증합니다. **특정 제품이 본 문서의 사고를 예방한다고 주장하지 않습니다.** LLMjacking 피해 금액 역시 "일 수만 달러 규모" 이상으로 확대 기재하지 않았으며, 이는 널리 인용되는 수치가 실제 과금 내역이 아닌 추정치이기 때문입니다.

본 문서에는 크리밋 고객사명이 포함되어 있지 않으며, 유출된 실제 키 또한 수록되지 않았습니다. 본문의 키 2개는 공개된 예시용 데이터입니다.

한국어판 안내

본 보고서는 영문판 **Blast Radius**의 한국어 번역/개정판입니다. 국내 독자에게는 GDPR보다 침해사고 신고 및 유출 통지 기한이 우선 적용되므로, 영문판에 없는 **정보통신망법 제48조의3**과 **개인정보 보호법 제34조**를 병기 반영했습니다.

상시 관리로 넘어가는 지점

플랫폼 담당 영역과 조직 담당 영역의 구분

본 문서에 수록된 침해사고는 대부분 단순 인지 부족이 아닌 **유지 관리 실패**에서 기인했습니다. 도구 도입에 앞서 자산 인벤토리 구축, 담당자 지정, 클라우드별 차단 절차 수립, 로테이션 현황 관리가 선행되어야 합니다. **크리밋 플랫폼**은 이 네 가지 핵심 요소 중 자동화 가능한 영역을 전담합니다. 아래 표는 플랫폼 담당 영역과 조직 내부 담당 영역을 구분하여 정리한 내용입니다.

단계	플랫폼 담당 영역	조직 잔존 영역
02 · 노출 범위	GitHub, GitHub Packages, GitLab, Bitbucket, AWS S3, Confluence, Jira, Notion, Slack, Google Drive 상시 스캔 수행 (커밋 히스토리 전체 및 티켓, 채팅 내역 포함)	스캔 대상 조직 및 리포지토리 선정, 커넥터 범위 외 표면 관리
01/04 · 유효성 검증	탐지 건별 유효성 검증을 통한 상태 부여 및 주기적 재확인 (단순 패턴 일치와 실제 인증 여부 구분)	해당 계정의 자사 소유 여부 확인 및 안전한 검증 절차 수립
03 · 영향 범위	AWS 및 GCP 역할/서비스 계정 키 연동 시 실효 권한 집합 자동 산출 (Permission Analyzer)	Azure, 리소스 정책 예외 상황 분석 및 해당 권한의 자사 데이터 영향도 평가
05 · 차단	탐지 건을 담당자·상태·이력 기반 사고(Incident) 단위로 관리 및 알림 연동	실제 시크릿 폐기 수행 (크리밋 플랫폼은 고객 시크릿 로테이션용 크리덴셜을 보관하지 않음)
06 · 사고 종료	크리덴셜 무효화 시 검증 상태 자동 전환 및 외부 접근 가능 유효 시크릿 추적·관리	커넥터 미포착 사용처 식별 및 수동 현황 관리
탐지 품질	Git 기반 탐지 룰 원본 관리 및 코드 리뷰를 통한 커밋 반영 (오탐 벤치마킹 수행)	자사 환경 내 탐지 대상 정의 및 관리 기준 수립

ⓘ 솔루션으로 해결 불가한 영역

원칙 5와 원칙 6은 사고 발생 전 선제적으로 결정해야 하는 보안 설정 항목입니다. GCP Data Access 로깅 활성화, Key Vault 진단 설정, CloudTrail 데이터 이벤트 기록, 서비스 계정 60분 비활성화 시 운영 가능 여부 등이 이에 해당합니다. 어떤 도구도 사후에 증거를 소급 생성할 수 없으며, 장애 감수 판단을 대신할 수 없습니다. 본 문서에서 단 하나의 항목만 재검토한다면 '시크릿 저장소 비교표' 확인을 적극 권장합니다.

사내 배포 안내

본 문서는 사내 런북 및 교육 자료로 별도 수정 없이 활용 가능합니다. 표 및 그림 발췌 시 출처 명시가 필요하며, 수정 활용을 원하실 경우 별도 요청을 통해 편집 가능한 원본 파일이 제공됩니다.

단기 실행 권장 항목

- 자사 플랫폼 기준 시크릿 저장소 비교표 작성 및 기록
- 시크릿 탐지 건별 담당자 지정 및 폐기 SLA 수립 (커밋 수행자가 아닌 전담 관리자 지정)
- 수집 스크립트 정기 사전 시운전 실시
- 최장기 사용 크리덴셜 10개 선정 및 식별·사용 주체 전수 점검

크리밋은 비인간 크리덴셜(Non-human Credentials) 탐지·검증·관리 플랫폼입니다. 본 문서는 크리밋 리서치 사고 대응 보고서 시리즈의 일환입니다.

영문판 **Blast Radius**와 본 한국어판은 동일한 내용을 담고 있으며, 국내 관련 규제 항목만 추가 반영되었습니다. 인용된 1차 출처 정보는 전 페이지(부록 C)에 기술되어 있습니다.

cremit.io · ben@cremit.io

세 구간

T1 → T3

탐지 소요 시간. 시크릿 유출 가능 표면 전체가 상시 스캔 범위에 포함되어야 소요 시간을 단축할 수 있습니다. (예: Toyota 5년 소모, xAI 탐지 후 60일 소모)

T3 → T4

차단 소요 시간. 담당자 사전 지정 및 클라우드별 런북 확보가 필요합니다. 60분 대기, 세션 Deny, SAS 정책 판단 절차가 사전 수립되어야 합니다.

T4 → T6

사고 종결 소요 시간. 자산 인벤토리 구축을 통해 단축 가능합니다. Cloudflare와 Internet Archive 사례는 모두 "미사용 판단"의 불확실성으로 인해 해당 구간이 장기화되었습니다.

Cremit

크리밋 리서치 · 사고 대응 시리즈

단 하나의 키.
단 하나의 주체.
권한이 미치는
모든 영역.

크리덴셜 노출은 단순 문자열 유출 사고가 아닙니다. 해당 문자열을 통해 도달 가능한 범위와 현재 시점에서 증명 가능한 영향도를 규명하는 침해사고입니다.

발행 2026년 9월 4일
출처 부록 C(32페이지) 수록
문의 ben@cremit.io