Hochschule Wismar
University of Applied Sciences Technology, Business and Design
Fakultät für Ingenieurwissenschaften

Master-Thesis

Einsatz von Cloud Services in Incident Response Prozessen

Eingereicht am: 18. Dezember 2023

von: Christian G. Winnen

Betreuer: Herr Prof. Dr. Nils Gruschka
Zweitbetreuer: Frau Prof. Dr. Antje Raab-Düsterhöft

Kurzreferat

Der zunehmende Einfluss von Cloud-Technologien verändert das Feld der Informationssicherheit grundlegend. Besonders im Bereich der Incident Response offenbaren sich durch Cloud Services neue Potenziale und Herausforderungen. Diese Master-Thesis erforscht, wie Cloud Services in Incident Response Prozesse integriert werden können, um die Effektivität und Effizienz der Reaktion auf Sicherheitsvorfälle zu steigern. Ziel ist es, die Unterschiede von Incident Response Ansätzen in on-premise und Cloud Umgebungen aufzuzeigen sowie ein tieferes Verständnis für Synergien zwischen Cloud-Technologien und Incident Response Strategien zu entwickeln. Die Integration der Cloud Services wird anhand von drei Incident Response Szenarien analysiert. Dabei zeigt sich, dass der Einsatz bestimmter Cloud Services vor allem in der Vorbereitungsphase des Incident Response Prozesses allgemeingültig anwendbar ist, während der Einsatz in den Phasen Erkennung, Analyse, Eindämmung, Behebung und Wiederherstellung abhängig von den jeweils betrachteten Incident Response Szenarien ist.

Abstract

The increasing influence of cloud technologies is fundamentally changing the information security landscape. Cloud services reveal new potential and challenges, particularly in the area of incident response. This master's thesis explores how cloud services can be integrated into incident response processes to increase the effectiveness and efficiency of responding to security incidents. The goal is to demonstrate the differences between incident response approaches in on-premise and cloud environments and to develop a deeper understanding of the synergies between cloud technologies and incident response strategies. The integration of cloud services is analyzed using three different incident response scenarios. It was identified that the use of certain cloud services is generally applicable in the preparation phase of incident response, whilst the use of cloud services in the detection, analysis, containment, eradication and recovery after security incidents depends on the incident response scenario under consideration.

Inhaltsverzeichnis

1 Einleitung

Immer mehr Organisationen verlagern ihre IT-Systeme in Public Clouds und nutzen dabei Services von Cloud-Anbietern Amazon Web Services (AWS), Microsoft Azure oder Google Cloud Platform (GCP).

Doch ähnlich wie in traditionellen on-premise Umgebungen, sind IT-Systeme in der Cloud anfällig für Sicherheitsvorfälle und erfordern daher geplante, strukturierte und funktionierende Incident Response Prozesse. Dies stellt Unternehmen vor neue Herausforderungen.

1.1 Zielsetzung

Das Ziel dieser Master-Thesis ist es, relevante Cloud Services vorzustellen und zu untersuchen, wie diese erfolgreich in Incident Response Prozesse integriert werden. Dabei werden die Unterschiede der Behandlung von Sicherheitsvorfällen in on-premise Landschaften zu Cloud Umgebungen berücksichtigt und sowohl Herausforderungen als auch Potentiale von Cloud Services untersucht.

Im Rahmen dieser Thesis werden drei Incident Response Szenarien konzipiert, anhand derer die Integration von Cloud Services betrachtet wird. Dabei wird der Einsatz von Cloud Services auf Allgemeingültigkeit geprüft und szenariospezifisch betrachtet.

1.2 Aufbau

Diese Master-Thesis beginnt mit der Vorstellung der theoretischen Grundlagen der Incident Response und des Cloud Computings in Kapitel 2.

Im dritten Kapitel wird auf die Gemeinsamkeiten und Unterschiede der Incident Response in on-premise und Cloud Umgebungen eingegangen sowie die Potentiale von Cloud Services hervorgehoben.

In Kapitel 4 und 5 werden drei verschiedene Incident Response Szenarien konzipiert und deren Ursachen, Auswirkungen und empirische Relevanz beschrieben, sowie der Aufbau von Incident Response in der Cloud dargestellt.

Kapitel 6 befasst sich mit der allgemeingültigen Integration von Cloud Services bei der Behandlung von Sicherheitsvorfällen.

Die Kapitel 7, 8 und 9 zielen auf die Integration von Cloud Services in den zuvor konzipierten Incident Response Szenarien ab.

Die Master-Thesis schließt mit dem Fazit in Kapitel 10 ab, welches neben der Zusammenfassung auch einen Ausblick auf zukünftige Forschung sowie eine kritische Betrachtung beinhaltet.

2 Theoretische Grundlagen

Dieses Kapitel befasst sich mit den theoretischen Grundlagen dieser Master-Thesis. Sie dienen als Fundament für die folgenden Betrachtungen. Der erste Abschnitt umfasst die Grundlagen rund um Incident Response im Allgemeinen. Im zweiten Abschnitt wird der Begriff, sowie die Hauptarten und Service Modelle des Cloud Computings vorgestellt.

2.1 Incident Response Grundlagen

2.1.1 Definition und Anwendungsgebiete

Der englische Begriff *Incident* wird im Deutschen mit *Vorfall*, *Ereignis* oder *Vorkommnis* übersetzt. In der IT und Informationssicherheit beziehen sich Incidents auf *sicherheitsrelevante Ereignisse*.

Events (Ereignisse) Bevor sicherheitsrelevante Ereignisse näher definiert werden können, muss zuerst der Begriff *Ereignis* bzw. der englische Begriff *Event* erläutert werden. Im Standard *ISO/IEC 27000:2018* definiert die International Organization for Standardization (ISO) ein Event als den Eintritt oder Änderung einer bestimmten Reihe von Umständen [1, S. 3]. Das US-amerikanische National Institute of Standards and Technology (NIST) definiert ein Event als jedes beobachtbare Ereignis in einem System oder Netzwerk [2, S. 6].

Nachfolgend sind einige Beispiele für Ereignisse in IT-Systemen bzw. Netzwerken aufgeführt:

- Ein Login-Versuch in einer Anwendung,
- das Versenden einer E-Mail,
- eine Web Application Firewall (WAF), die einen Request blockiert,
- ein Shell-Command, der auf einem Server ausgeführt wird,

- die Provisionierung einer neuen Serverinstanz,
- das Löschen von Dateien auf einem Filesystem.

Incident (Sicherheitsvorfall) Wie eingangs beschrieben, handelt es sich bei einem Incident um ein *sicherheitsrelevantes* Ereignis bzw. Event.

Das Bundesamt für Sicherheit in der Informationstechnik (BSI) definiert im IT-Grundschutzkompendium (GSK) Baustein Detektion und Reaktion (DER) ein solches Ereignis, das sich *„auf die Informationssicherheit auswirkt und die Vertraulichkeit, Integrität oder Verfügbarkeit beeinträchtigen kann“* [3, DER.1].

Das NIST definiert einen *computer security incident* folgendermaßen: *„A computer security incident is a violation or imminent threat of violation of computer security policies, acceptable use policies, or standard security practices“* [2, S. 6].

Die ISO verwendet in der *ISO/IEC 27000:2018* die folgende Definition für einen *information security incident*: *„single or a series of unwanted or unexpected information security events that have a significant probability of compromising business operations and threatening information security“* [1, S. 4].

Unter dem Begriff *incident response* versteht man laut IT-Grundschutzkompendium des BSI ein *„vorgegebenes und erprobtes Verfahren zur Behandlung von Sicherheitsvorfällen“* [3, DER.2.1]. Sprich ein systematisches, koordiniertes und einheitliches Vorgehen als Reaktion auf eingetretene Sicherheitsvorfälle innerhalb einer Organisation.

Nachfolgend sind einige Beispiele für Sicherheitsvorfälle in IT-Systemen bzw. Netzwerken aufgeführt:

- Ein Secure Shell (SSH) Brute-Force Angriff, bei dem automatisiert verschiedene Kombinationen von Username und Passwort ausprobiert werden.
- Ein Mitarbeiter öffnet einen Anhang einer Phising-Mail, welcher Schadsoftware enthält.
- Ein SQL-Injection Angriff wird gegen eine Webapplikation gestartet.
- In einer Cloud Umgebung werden mehrere Cryptominer gestartet.
- Dateien auf einem zentralen Fileserver werden exilftriert und anschließend verschlüsselt.

2.1.2 Notwendigkeit und Motivation

Incident Response in der IT ist mittlerweile ein akzeptiertes und weit verbreitetes Konzept in Unternehmen. Das liegt unter anderem daran, dass sich für Unternehmen eine Reihe von Vorteilen aus einem strukturierten Vorgehen zur Behandlung von Sicherheitsvorfällen ergeben:

- **Systematik:** Eine systematische Reaktion auf Vorfälle stellt sicher, dass die angemessenen Schritte als Folge eingeleitet werden.
- **Erfüllung der Schutzziele:** Durch Incident Response werden der Verlust oder Diebstahl von Informationen, Systemausfälle oder Manipulation von Daten minimiert, wodurch die Schutzziele Vertraulichkeit, Verfügbarkeit und Integrität eingehalten werden.
- **Lessons learned:** Außerdem besteht die Möglichkeit Informationen einer durchgeführten Vorfallbehandlung zu nutzen, um sich auf zukünftige Incidents vorzubereiten und weitere Systemhärtungsmaßnahmen abzuleiten.
- **Koordination rechtlicher Schritte:** Des Weiteren können Organisation mit funktionsübergreifenden Incident Response Teams schneller die ordnungsgemäße Bewältigung rechtlicher Schritte einleiten und koordinieren [2, S. 6–7].

Neben diesen Vorteilen aus Unternehmenssicht gibt es allerdings noch weitere regulatorische Anforderungen an Organisationen aufgrund von Gesetzen, Vorschriften und Richtlinien. Dadurch können diese teilweise verpflichtet sein Incident Response Prozesse zu etablieren, welche wiederum bestimmten Anforderungen entsprechen.

Für deutsche Bundesbehörden, sowie Kritische Infrastrukturen (KRITIS) Betreiber gelten Vorgaben zur Umsetzung nach *Stand der Technik*. Um den Stand der Technik umzusetzen, können Organisationen den *BSI-Standard 200-2* zur IT-Grundschutz-Methodik verwenden. Je nach resultierendem Schutzbedarf, sind Organisationen verpflichtet eine *Basis-Absicherung, Standard-Absicherung* oder *Kern-Absicherung* umzusetzen.

Konkrete Vorgaben an die Incident Response von Organisation ergeben sich aus den GSK Bausteinen der Kategorie Detektion und Reaktion (DER). Die konkreten Anforderungen des jeweiligen Bausteins sind abhängig von dem gewählten Absicherungslevel der IT-Grundschutz-Methodik. Abbildung 1 verdeutlicht diesen Zusammenhang bildlich.

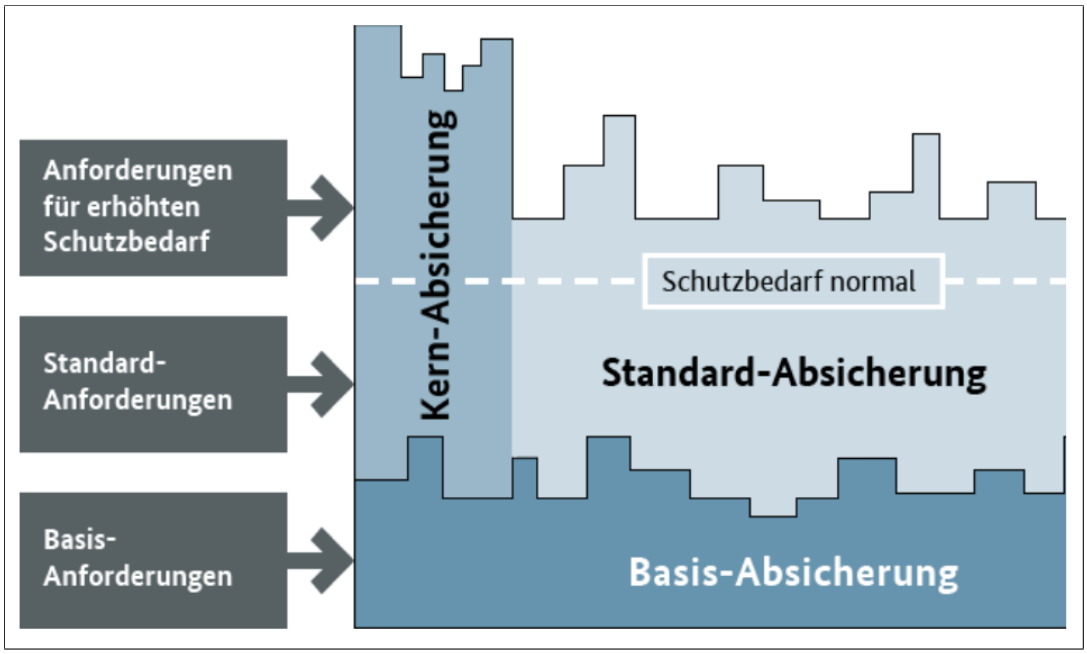


Abbildung 1: Anforderungen der IT-Grundschutz-Bausteine [4, Lektion 6.1]

Für US-amerikanische Organisationen, leiten sich rechtliche verbindliche Vorgaben bei der Umsetzung von Incident Response aus der *NIST Special Publication 800-61 Rev. 2* ab.

Die Zielgruppe, für die diese Anforderungen rechtlich bindend sind, sind hauptsächlich Bundesbehörden der US-Regierung, staatliche und lokale Regierungsbehörden, KRITIS-Betreiber, sowie Auftragnehmer und Dienstleister solcher. Analog zu den unterschiedlichen Anforderungen der GSK Bausteine je nach Art der Absicherung, sind die Anforderungen in der NIST Richtlinie nach *MUST* und *SHOULD* unterteilt.

Des Weiteren gelten für Organisationen, die sich nach ISO/IEC 27035 zertifizieren lassen, die Anforderungen der ISO-Standard Reihe. Diese ist wiederum in vier Teile untergliedert:

- *Part 1: Principles and process:* Hierbei handelt es sich um die Grundlage der 27035-Reihe. In diesem Teil werden grundlegende Konzepte, Prinzipien und das Phasenmodell für Incident Response Prozesse vorgestellt.
- *Part 2: Guidelines to plan and prepare for incident response:* Dieser Teil enthält Richtlinien zur Planung und Vorbereitung der Reaktion auf Incidents, sowie dem Anwenden von Lessons Learned nach einem Vorfall. Die Richtlinien

basieren auf den Phasen *plan and prepare* und *learn lessons* des Phasenmodells für das Management von Informationssicherheitsvorfällen.

- *Part 3: Guidelines for ICT incident response operations:* In diesem Dokument werden Richtlinien für die Reaktion auf Sicherheitsvorfälle im Bereich Information and Communication Technology (ICT) dargestellt. Dabei werden operationelle Aspekte aus der Perspektive von Menschen, Prozessen und Technologie behandelt. Zudem umfasst es die Prozessschritte Detektion, Berichterstattung, Triage, Analyse, Reaktion, Eindämmung, Beseitigung, Wiederherstellung und Schlussfolgerung.
- *Part 4: Coordination:* Der letzte Teil der 27035-Reihe befindet sich noch im Draft Status und ist bisher noch nicht veröffentlicht.

Bisher wurden Beispiele genannt welche Anforderungen an das Incident Response von Organisationen sich durch gesetzliche Vorgaben oder zur Erfüllung von Standards gelten können. Es gibt allerdings noch weitere Frameworks und Best Practice Ansätze, die von Organisationen auf freiwilliger Basis genutzt werden können.

Als Beispiel sei hier auf die *Top 10 Considerations For Incident Response* verwiesen. Dieses Dokument ist analog zu anderen von Veröffentlichungen von OWASP in eine *Top 10* aufgebaut und enthält zehn Faktoren die es im Bereich Incident Response zu betrachten gilt. Das Open Worldwide Application Security Project (OWASP) ist eine gemeinnützige Stiftung, die sich für die Verbesserung von Softwaresicherheit einsetzt. [6]

2.1.3 Phasen und Aufgaben des Incident Response Prozesses

Im vorherigen Kapitel wurden bereits die verschiedenen Phasen des Incident Response Prozesses erwähnt. Nachfolgend werden die Phasenmodelle des *NIST Special Publication 800-61 Rev. 2* und *ISO/IEC 27035:2023* vorgestellt.

Das NIST bezeichnet die Phasen als *Incident Response Life Cycle*, welcher sich aus den folgenden vier Phasen zusammensetzt:

1. Preparation (*Vorbereitung*)
2. Detection & Analysis (*Erkennung & Analyse*)
3. Containment, Eradication & Recovery (*Eindämmung, Beseitigung & Wiederherstellung*)

4. Post-Incident Recovery (*Aktivitäten nach Vorfällen*) [2, S. 21]

Dabei ist es wichtig zu betonen, dass es sich hierbei um einen Zyklus, der sich stetig wiederholt und von vorne beginnt, und nicht um einen statischen, linearen Prozess mit einmaligem Start und Ende handelt. Dieser kontinuierliche Kreislauf ist in Abbildung 2 klar erkennbar.

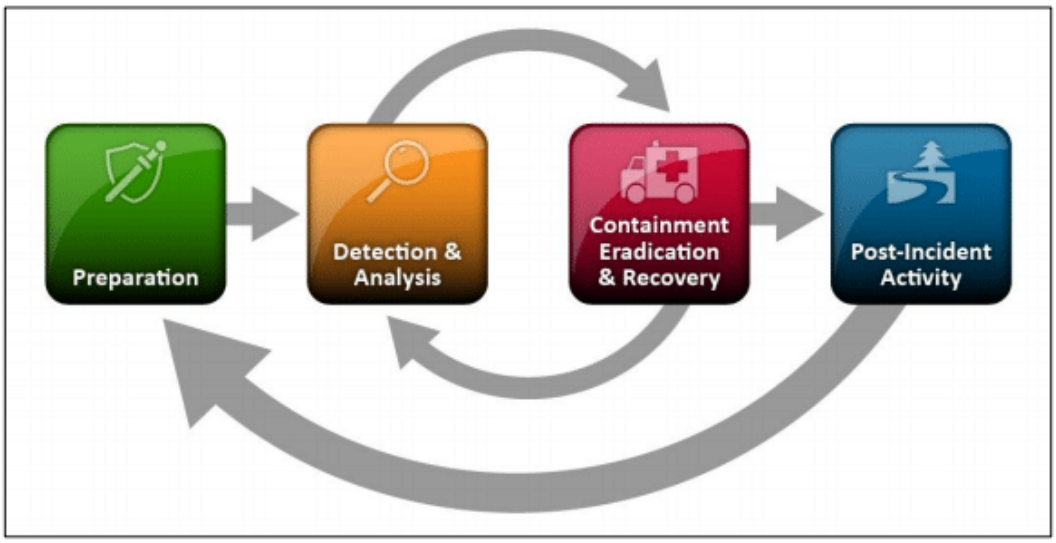


Abbildung 2: Incident Response Life Cycle nach *NIST Special Publication 800-61 Rev. 2* [2, S. 21]

In der anfänglichen Phase des Incident Response Life Cycles werden ein Incident Response Team aufgebaut, geschult und die erforderlichen Werkzeuge und Ressourcen beschafft. Während der Vorbereitung versucht die Organisation außerdem, die Anzahl der Vorfälle durch die Auswahl und Implementierung von Sicherheitsmaßnahmen auf der Grundlage von Risikobewertungen zu begrenzen. Nach der Implementierung der Maßnahmen bleibt jedoch zwangsläufig ein Restrisiko bestehen.

Die Erkennung von Sicherheitslecks ist daher notwendig, um die Organisation zu benachrichtigen, wenn Vorfälle auftreten. Abhängig von der Schwere des Vorfalls kann die Organisation die Auswirkungen des Vorfalls durch Eindämmung begrenzen und schließlich zur Ausgangslage wiederherstellen. Während dieser Phase kehrt die Aktivität oft zur Erkennung und Analyse zurück, beispielsweise um festzustellen, ob zusätzliche Hosts von Malware befallen sind, während ein Malware-Vorfall beseitigt wird.

Nach angemessener Behandlung des Vorfalls erstellt die Organisation einen Bericht, der die Ursache und die Kosten des Vorfalls sowie die Maßnahmen beschreibt, die

die Organisation ergreifen sollte, um zukünftige Vorfälle zu verhindern. Durch den Übergang zwischen der letzten Phase (Post-Incident Activity) in die erste Phase (Preparation) stellt sich ein kontinuierlicher Kreislauf [2, S. 21].

Die ISO verwendet in *ISO/IEC 27035-1:2023* ebenfalls ein zyklisches Phasenmodell für den Incident Response Prozess. Wie im vorherigen Kapitel erwähnt handelt es sich dabei um die folgenden fünf Aktivitäten:

1. Plan and Prepare (*Planung und Vorbereitung*)
2. Detect and Report (*Erkennung und Berichterstattung*)
3. Assess and Decide (*Beurteilung und Entscheidung*)
4. Respond (*Reagieren*)
5. Learn Lessons (*Erkenntnisse ziehen*)

In der Phase *Planung und Vorbereitung* etablieren Organisationen ein Incident Response Team, schulen Mitarbeiter, und stellen die erforderlichen Ressourcen und Notfallpläne bereit, um auf mögliche Vorfälle vorbereitet zu sein.

Bei der *Erkennung und Berichterstattung* liegt der Fokus auf der Identifizierung von Sicherheitslecks und der rechtzeitigen Benachrichtigung der zuständigen Stellen, wobei Frühwarnsysteme und Überwachungsmechanismen eingesetzt werden, um Anomalien zu erkennen.

Während der Phase *Beurteilung und Entscheidung* erfolgt die Analyse des Vorfalls, die Bestimmung seiner Schwere und die Festlegung von Maßnahmen zur Eindämmung und Bewältigung, wobei die Vorgehensweise und die Koordination der Incident Response Maßnahmen festgelegt werden.

In der Phase *Reagieren* setzen Incident Response Teams die geplanten Maßnahmen zur Eindämmung und Wiederherstellung in die Tat um, indem sie aktiv daran arbeiten, den Vorfall zu bewältigen und weitere Schäden zu verhindern.

Zum Schluss, in der Phase *Learn Lessons*, erfolgt nach Abschluss des Vorfalls eine umfassende Analyse und Evaluierung. Die gewonnenen Erkenntnisse dienen dazu, zukünftige Vorfälle zu verhindern und die Incident Response Strategie kontinuierlich zu verbessern. Auch hier geht die letzte Phase in die erste Phase über, weshalb dieses Phasenmodell ebenfalls einen kontinuierlichen Zyklus darstellt. Abbildung 3 verdeutlicht dies nochmals.

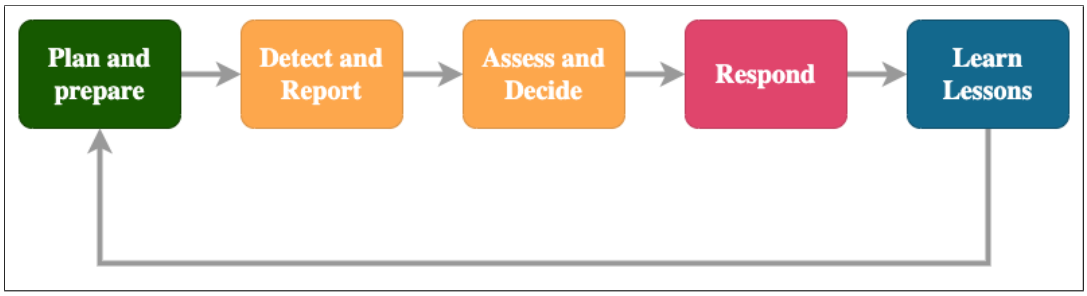


Abbildung 3: Incident Response Lifecycle nach *ISO/IEC 27035-1:2023*

Auch wenn beide Leitfäden auf den ersten Blick unterschiedliche Phasenmodelle - verschiedene Anzahl als auch Bezeichnungen der Phasen - für Incident Response Prozesse aufweisen, so gibt es doch eine Reihe von Gemeinsamkeiten und Überlappungen.

Beide Phasenmodelle teilen eine gemeinsame Struktur und ähnliche Phasen, die die Aktivitäten und Entscheidungen während des Managements von Informationssicherheitsvorfällen beschreiben. In Abbildung 3 sind deshalb die Farben bewusst so gewählt, dass sie den korrespondierenden, vergleichbaren Phasen des Modells der *NIST Special Publication 800-61 Rev. 2* in Abbildung 2 passen:

1. Preparation <-> Plan and Prepare
2. Detection & Analysis <-> Detect and Report, Assess and Decide
3. Containment, Eradication & Recovery <-> Respond
4. Post-Incident Activity <-> Learn Lessons

Beide Modelle betonen einen strukturierten Ansatz, um auf Vorfälle zu reagieren und aus ihnen zu lernen. Beide Modelle erkennen die Bedeutung der Kommunikation während eines Vorfalls an und dass Informationen an die richtigen Stakeholder, wie das Management und betroffene Parteien, weitergegeben werden sollten. Außerdem werden die Phasen des Incident Response als ein kontinuierlicher Zyklus angesehen.

Zusammenfassend konzentrieren sich beide Leitfäden auf einen proaktiven und strukturierten Ansatz zum Incident Management, um Organisationen zu helfen, auf Sicherheitsvorfälle effektiv zu reagieren und die Auswirkungen von Sicherheitsvorfällen zu minimieren.

Für die nachfolgenden Betrachtungen dieser Arbeit wird das Phasenmodell der *NIST Special Publication 800-61 Rev. 2* gemäß Abbildung 2 herangezogen, da es das

weit verbreitetste Modell ist und in den meisten Organisationen zur Anwendung kommt.

2.1.4 Incident Response Teams

Neben dem Phasenmodell des Incident Response Zyklus werden sowohl in der *NIST Special Publication 800-61 Rev. 2* als auch in der *ISO/IEC 27035-1:2023* der Aufbau und die Struktur von Incident Response Teams erläutert, auch wenn in der ISO-Norm dies nur in einem Paragraphen aufgeführt ist. Die ISO-Norm definiert ein Incident Response Team als „*Team aus entsprechend qualifizierten und vertrauenswürdigen Mitgliedern einer Organisation, das auf Vorfälle koordiniert reagiert und diese löst*“ [8, S. 1].

Das Kapitel *Incident Response Team Structure* der *NIST Special Publication 800-61 Rev. 2* behandelt die Bildung und Organisation von Incident Response Teams, die bei Sicherheitsvorfällen in Organisationen eingreifen. Es legt dar, dass die Effektivität des Incident Response Teams von der Mitwirkung und Kooperation verschiedener Organisationsmitglieder abhängt.

Für die Teamstruktur werden unterschiedliche Modelle vorgeschlagen:

1. **Zentrales Incident Response Team**: Ein einziges Team ist für die gesamte Organisation zuständig, geeignet für kleinere Organisationen oder solche mit geringer geografischer Streuung.
2. **Verteilte Incident Response Teams**: Mehrere Teams sind jeweils für bestimmte logische oder physische Segmente der Organisation verantwortlich. Dieses Modell empfiehlt sich für größere Organisationen oder solche mit wichtigen Ressourcen an entfernten Standorten.
3. **Koordinierendes Team**: Ein Team bietet Beratung ohne direkte Autorität über andere Teams [2, S. 13].

Bei der Personalbesetzung werden die folgenden drei Ansätze unterschieden:

1. **Vollständig interne Teams**: Die Organisation übernimmt die gesamte Incident Response-Arbeit selbst.
2. **Teilweise ausgelagerte Teams**: Bestimmte Aufgabenbereiche werden an externe Dienstleister ausgelagert.

3. **Vollständig ausgelagerte Teams**: Die gesamte Incident Response-Arbeit wird an externe Dienstleister vergeben [vgl. 2, S. 14].

Das Kapitel betont zudem, dass die Teammitglieder über ausgezeichnete technische Fähigkeiten in den Bereichen System- und Netzwerkadministration, Programmierung, technischer Support oder Threat Detection verfügen sollten. Zusätzlich sind Problemlösungsfähigkeiten und kritisches Denken erforderlich. Die Weiterentwicklung und der Erhalt von Kompetenzen werden als wesentlich für die Verhinderung von Mitarbeiter Burnout angesehen [vgl. 2, S. 16–17].

2.2 Einführung in Cloud Computing

Cloud Computing repräsentiert einen fundamentalen Wandel in der Bereitstellung und Verwaltung von IT-Ressourcen. Es bietet Flexibilität, Skalierbarkeit und Kosteneffizienz und hat sich zu einem unverzichtbaren Bestandteil moderner IT- Strategien entwickelt. In diesem Kapitel werden die theoretischen Grundlagen des Cloud Computings vertieft, mit besonderem Fokus auf den verschiedenen Arten und Service Modellen des Cloud Computings.

2.2.1 Definition und Charakteristika

Zunächst muss der Begriff Cloud Computing definiert werden.

Bislang existiert keine allgemein anerkannte Definition von Cloud Computing, da in wissenschaftlichen Arbeiten und Präsentationen oft leicht unterschiedliche Erklärungen verwendet werden. Eine häufig zitierte Beschreibung stammt vom NIST in den USA, welche auch von der European Network and Information Security Agency (ENISA) herangezogen wird [vgl. 9].

In der *NIST SP 800-145* wird der Begriff Cloud Computing folgendermaßen definiert: Cloud Computing ist ein Modell, das es erlaubt bei Bedarf, jederzeit und überall bequem über ein Netz auf einen geteilten Pool von konfigurierbaren Rechnerressourcen (z. B. Netze, Server, Speichersysteme, Anwendungen und Dienste) zuzugreifen, die schnell und mit minimalem Managementaufwand oder geringer Serviceprovider-Interaktion zur Verfügung gestellt werden können [10, S. 2].

Zudem zeichnen sich Cloud Services nach NIST durch die folgenden fünf Eigenschaften aus:

1. **On-demand Self Service**: Die Provisionierung der Ressourcen (z. B. Rechenleistung, Storage) läuft automatisch ohne Interaktion mit dem Service Provider ab.
2. **Broad Network Access**: Die Services sind mit Standard-Mechanismen über das Netz verfügbar und nicht an einen bestimmten Client gebunden.
3. **Resource Pooling**: Die Ressourcen des Anbieters liegen in einem Pool vor, aus dem sich viele Anwender bedienen können (Multi-Tenant Modell). Dabei wissen die Anwender nicht, wo die Ressourcen sich befinden, sie können aber vertraglich den Speicherort, also z. B. Region, Land oder Rechenzentrum, festlegen.
4. **Rapid Elasticity**: Die Services können schnell und elastisch zur Verfügung gestellt werden, in manchen Fällen auch automatisch. Aus Anwendersicht scheinen die Ressourcen daher unendlich zu sein.
5. **Measured Services**: Die Ressourcennutzung kann gemessen und überwacht werden und entsprechend bemessen auch den Cloud-Anwendern zur Verfügung gestellt werden [9].

2.2.2 Deployment Models

Des Weiteren gibt es im Cloud Computing verschiedene *Deployment Models (Bereitstellungsmodelle)*. Die NIST führt hierbei die folgenden vier Varianten auf:

1. **Private Cloud**: Eine private Cloud-Infrastruktur wird speziell für den exklusiven Gebrauch einer einzelnen Organisation, die mehrere Nutzer umfasst (beispielsweise verschiedene Geschäftseinheiten), bereitgestellt. Sie kann im Besitz der Organisation sein und entweder von dieser selbst, einem Drittanbieter oder einer Kombination aus beiden betrieben werden, wobei sie sich sowohl vor Ort als auch extern befinden kann.
2. **Community Cloud**: Bei einer Community Cloud wird die Infrastruktur ausschließlich für eine spezifische Gemeinschaft von Nutzern aus Organisationen mit gemeinsamen Anliegen (wie Mission, Sicherheitsanforderungen, Richtlinien und Compliance-Überlegungen) bereitgestellt. Diese kann von einer oder mehreren Organisationen innerhalb der Gemeinschaft, einem Drittanbieter oder einer Kombination daraus betrieben werden und sowohl intern als auch extern angesiedelt sein.

3. **Public Cloud**: Eine Public Cloud hingegen wird für die allgemeine Nutzung durch die Öffentlichkeit zur Verfügung gestellt. Sie kann von einem Unternehmen, einer akademischen Einrichtung oder einer Regierungsorganisation, oder einer Kombination davon, betrieben werden und befindet sich auf dem Gelände des Cloud-Anbieters.

4. **Hybrid Cloud**: Die Hybrid-Cloud stellt eine Kombination aus zwei oder mehreren unterschiedlichen Cloud- Infrastrukturen (privat, Community oder öffentlich) dar, die zwar eigenständige Einheiten bleiben, jedoch durch standardisierte oder eigens entwickelte Technologien miteinander verbunden sind. Diese ermöglichen den Transfer von Daten und Anwendungen, zum Beispiel das sogenannte Cloud Bursting zur Lastenverteilung zwischen verschiedenen Clouds [vgl. 10, S. 3].

Im Rahmen dieser Arbeit werden hauptsächlich Umgebungen betrachtet, denen ein Public oder Hybrid-Cloud Bereitstellungsmodell zugrunde liegt.

2.2.3 Service Models

Cloud Computing lässt sich in drei *Service Models (Dienstleistungsmodelle)* einteilen. Diese werden nachfolgend erläutert und mit Beispielen der größten drei größten Cloud Service Provider (CSP) belegt.

1. **Software as a service (SaaS)**: Software as a service (SaaS) ermöglicht es Nutzern, Anwendungen des Anbieters zu verwenden, die auf einer Cloud-Infrastruktur betrieben werden. Diese Anwendungen sind über verschiedene Endgeräte zugänglich, entweder durch eine schlanke Client-Schnittstelle wie einen Webbrowser (beispielsweise webbasierte E-Mail-Dienste) oder über eine Programmschnittstelle. Der Nutzer hat dabei keine Kontrolle über die zugrundeliegende Cloud-Infrastruktur, einschließlich Netzwerk, Server, Betriebssysteme, Speicher oder sogar individuelle Anwendungsfunktionen, mit Ausnahme von möglicherweise begrenzten, benutzerspezifischen Anwendungskonfigurationseinstellungen [vgl. 10, S. 2]. Beispiele hierfür sind Google Workspace oder Microsoft Office 365, welche umfassende Produktanwendungen für E-Mail, Kalender, Textverarbeitung, Präsentationserstellung oder Tabellenkalkulation beinhalten.

2. **Platform as a service (PaaS)**: Platform as a service (PaaS) stellt dem Nutzer die Möglichkeit zur Verfügung, auf der Cloud-Infrastruktur selbst erstellte

oder erworbene Anwendungen zu implementieren, die unter Verwendung von Programmiersprachen, Bibliotheken, Diensten und Werkzeugen des Anbieters erstellt wurden. Der Nutzer verwaltet oder kontrolliert nicht die darunterliegende Cloud-Infrastruktur, einschließlich Netzwerk, Server, Betriebssysteme oder Speicher, hat jedoch Kontrolle über die eingesetzten Anwendungen und möglicherweise über Konfigurationseinstellungen für die Anwendungshosting-Umgebung [vgl. 10, S. 2–3]. Als Beispiele hierfür können die vollständig verwalteten Datenbankdienste Amazon Relational Database Service (Amazon RDS) oder Google Cloud SQL aufgeführt werden.

3. **Infrastructure as a service (IaaS)**: Infrastructure as a service (IaaS) bietet dem Nutzer die Fähigkeit, Verarbeitungsleistung, Speicher, Netzwerke und andere grundlegende Rechenressourcen zu provisionieren, wobei der Nutzer in der Lage ist, beliebige Software zu implementieren und auszuführen, einschließlich Betriebssystemen und Anwendungen. Der Nutzer hat keine Kontrolle über die darunterliegende Cloud Infrastruktur, jedoch Kontrolle über Betriebssysteme, Speicher und eingesetzte Anwendungen; und möglicherweise begrenzte Kontrolle über ausgewählte Netzwerkkomponenten wie beispielsweise Host-Firewalls [10, S. 3]. Hierzu gehören beispielsweise Amazon Elastic Compute Cloud (EC2), Azure Virtual Machines oder Google Compute Engine (GCE).

Abbildung 4 zeigt die Kontrolle als auch die Verantwortung über verschiedene Systemkomponenten je nach gewähltem Service Modell.

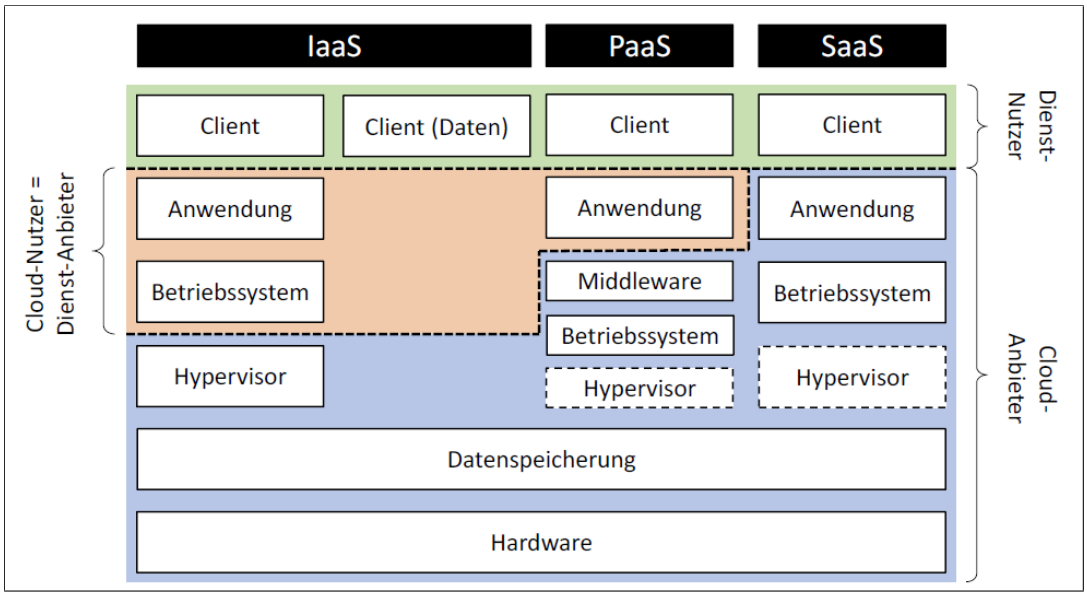


Abbildung 4: Systemkomponenten in verschiedenen Dienstmodellen [11, S. 12]

In der betrachteten Abbildung wird ersichtlich, dass mit einer Bewegung von links nach rechts eine Verringerung der Flexibilität in Bezug auf die Nutzungsmöglichkeiten von Cloud-Diensten einhergeht. Parallel dazu reduziert sich jedoch auch der Aufwand für Wartung und Administration. Zudem entfallen für den Cloud-Nutzer Lizenzkosten für jene Komponenten, die direkt vom Cloud-Anbieter zur Verfügung gestellt werden. Gleichzeitig verringert sich der Anteil des Gesamtsystems, der unter der direkten Kontrolle des Cloud-Nutzers steht.

Es muss jedoch berücksichtigt werden, dass bei jedem Modell die ultimative Kontrolle beim Cloud-Anbieter verbleibt. So hat der Nutzer im Infrastructure as a service (IaaS)-Modell zwar die Möglichkeit, seine eigene virtuelle Maschine und Anwendungen zum Cloud-Anbieter hochzuladen und zu betreiben, doch diese Systeme laufen auf der Hardware des Anbieters. Dieser hat demnach die Fähigkeit, jedes virtuelle System zu überwachen und zu modifizieren.[vgl. 11, S. 13–14] Diese Eigenschaft hat vor allem in Bezug auf Datenschutz enorme rechtliche als auch technische Implikationen, welche im Rahmen dieser Arbeit jedoch nicht näher beleuchtet werden.

3 Incident Response in on-premise vs. Cloud-Umgebungen

3.1 Besonderheiten und Unterschiede

In Kapitel 2.1 wurden die theoretischen Grundlagen der Incident Response erläutert. Dabei lag der Fokus auf der Behandlung von Sicherheitsvorfällen in traditionellen on-premise Umgebungen. Doch durch die Etablierung von Cloud Computing ging ein Paradigmenwechsel einher, der auch die Art und Weise von Incident Response betrifft. Deshalb werden in dieser Sektion die Unterschiede bei der Behandlung von Sicherheitsvorfällen in der Cloud im Vergleich zu on-premise IT-Landschaften erarbeitet.

3.1.1 Geteilte Verantwortung für Sicherheit

Im Kontext des Cloud-Computings ist das *Shared Responsibility Model* ein zentrales Konzept, das die Verteilung der Sicherheitsverantwortung zwischen Cloud-Anbietern und Cloud-Nutzern definiert. Während Cloud-Anbieter für die *Sicherheit der Cloud*-Infrastruktur verantwortlich sind, wie beispielsweise Rechenzentren, Netzwerke und Hardware, liegt die Verantwortung für die *Sicherheit in der Cloud* – also Daten, Anwendungen und Zugriffsmanagement – bei Cloud-Nutzern. Diese geteilte Verantwortung beeinflusst maßgeblich die Incident Response Prozesse. In einer Cloud-Umgebung müssen die Cloud-Nutzer ihre eigenen Incident Response Strategien entwickeln, die mit den Tools und Diensten des Cloud-Anbieters harmonieren, um eine umfassende Reaktionsfähigkeit auf Sicherheitsvorfälle zu gewährleisten.

Im Gegensatz dazu sind in on-premise Umgebungen die Organisationen vollständig für die gesamte Bandbreite der Sicherheitsmaßnahmen verantwortlich, von der physischen bis zur Anwendungssicherheit. Dies erfordert in der Regel eine größere Investition in Sicherheitsinfrastruktur und -personal. Der wesentliche Unterschied in Cloud-Umgebungen besteht darin, dass Organisationen auf die fortgeschrittenen Sicherheitsmaßnahmen des Cloud-Anbieters zurückgreifen können, was jedoch eine

klare Abstimmung und ein Verständnis der jeweiligen Verantwortlichkeiten voraussetzt. Dadurch, dass bestimmte Aspekte der Sicherheit an den Cloud-Anbieter ausgelagert sind, können sich die Organisationen stärker auf die Sicherung ihrer Daten und Anwendungen konzentrieren, was eine effizientere Ressourcennutzung ermöglicht.

Wie in Kapitel 2.2 beschrieben, ändern sich je nach gewähltem Dienstmodel die Systemkomponenten, die unter der Kontrolle der Cloud-Nutzer stehen. Dadurch verschieben sich zwangsläufig auch die Verantwortlichkeiten für Aufgaben und Tätigkeiten rund um diese Systemkomponenten. AWS verwendet häufig die Begriffe *security* ***of*** *the cloud*, um die Verantwortung der Cloud-Anbieter zu beschreiben, und *security* ***in*** *the cloud*, um die Verantwortung der Cloud-Nutzer zu erläutern [12, Shared Responsibility Model]. Abbildung 5 ist aus dem Whitepaper *Amazon Web Services: Risk and Compliance* und zeigt die Verantwortlichkeiten der Cloud-Nutzer (*Customer*) und der Cloud-Anbieter (AWS).

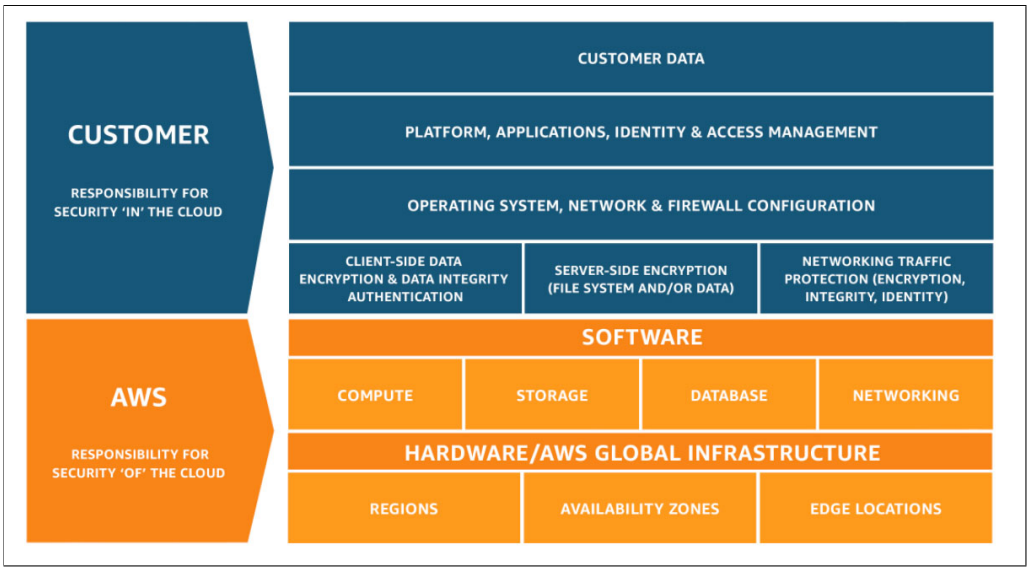


Abbildung 5: Shared Responsibility Model [12, Shared Responsibility Model]

3.1.2 Detaillierungsgrad und Verfügbarkeit von Logs

Ein weiterer signifikanter Unterschied von Incident Response in Public Cloud Umgebungen im Vergleich zu on-premise Umgebungen, bezieht sich auf die Aktivitäten in der *Detection & Analysis* sowie in der *Post-Incident Activity*-Phase. Die primären forensischen Artefakten in Public Cloud Umgebungen sind Logs anstelle von Disk Images. Eine weitere Besonderheit stellt zudem die Verfügbarkeit, sowie der De-

taillierungsgrad von Logs in Bezug auf das gewählte Service Modell. Abbildung 6 verdeutlicht diesen Zusammenhang:

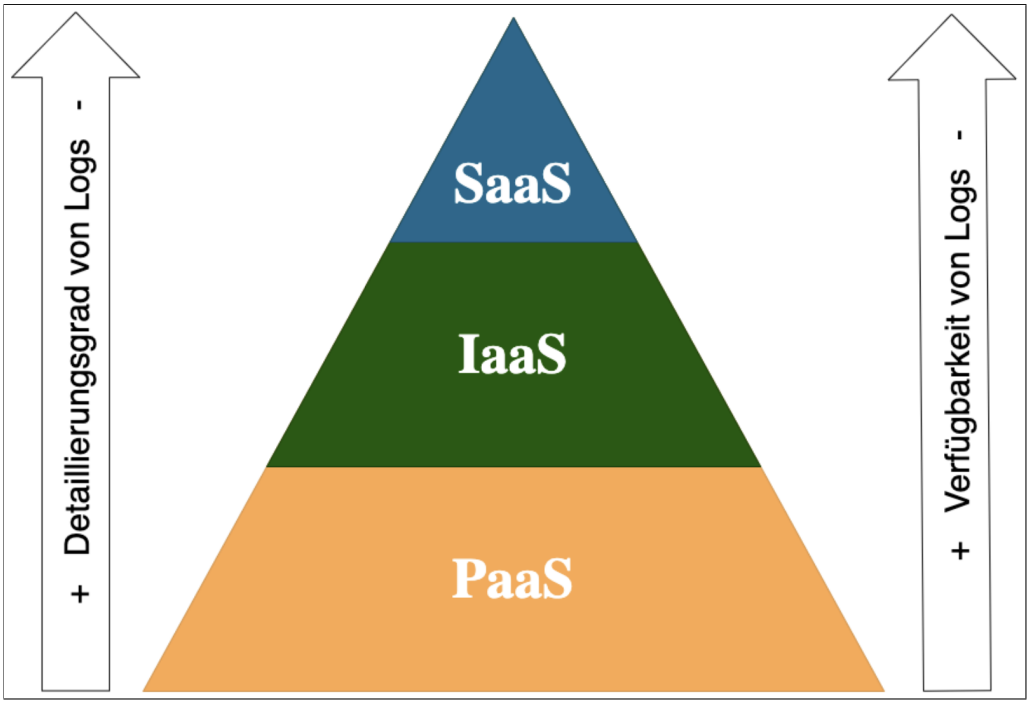


Abbildung 6: Detaillierungsgrad und Verfügbarkeit von Logs nach Service Model [eigene Darstellung in Anlehnung an 13, Folie 5]

Bewegt man sich anhand der Servicemodell Pyramide nach oben (von IaaS zu Platform as a service (PaaS) zu Software as a service (SaaS)), sinkt durch die Abstrahierung und das Shared Responsibility Modell der operationelle Wartungs- und Betriebsaufwand. Gleichzeitig sinkt damit auch die Kontrolle über das Gesamt-system, welches unter der direkten Kontrolle der Cloud-Nutzer steht [vgl. 11, S. 13]. Dies hat zur Folge, dass die verfügbaren Logs während als auch nach einem Sicherheitsvorfall je nach gewähltem Service Modell nicht ausreichend für die Behandlung eines Sicherheitsvorfalls sein können.

Eine weitere Besonderheit in Bezug auf die Verfügbarkeit von Logs sind die unterschiedlichen Preismodelle je nach Log Art. Hierbei unterscheiden Cloud-Anbieter in Logs von *data plane operations (Data Events)* und *control plane operations (Management Events)*.

Data Events Data Events zeigen die auf bzw. innerhalb einer Ressource durchgeführten Ressourcenoperationen an. Dazu gehört z.B. das Aufrufen, Bearbeiten oder

Löschen eines Datenobjekts in einem Amazon Simple Storage Service (S3) Bucket.

Management Events Management Events zeigen Verwaltungsoperationen an, welche für Ressourcen ausgeführt werden. Dazu gehört z.B, das Starten, Stoppen oder die Terminierung einer EC2 Instanz. Im Vergleich zu Management Events, welche in der Regel ein geringeres Volumen aufweisen, sind Data Events standardmäßig nicht verfügbar und müssen in Verbindung mit höheren Kosten expliziert aktiviert werden. Da Logs für Data Events häufig für die Bestimmung von Indicators of Compromise (IoC) erforderlich sind, kann diese Kostenhürde zu enormen Risiken für Unternehmen führen.

In der wissenschaftlichen Terminologie wird der Begriff IoC als eine Reihe von technischen Attributen definiert, die zur Identifikation und Analyse von Cyberangriffen beitragen. Diese Attribute umfassen diverse Elemente, wie Reste von Programmcode, sowie IP-Adressen und Domänennamen, die es ermöglichen, den Netzwerkverkehr zu analysieren und verdächtige Aktivitäten zu identifizieren [14].

Ein Beispiel dafür ist die Bestimmung von IoCs im Zusammenhang mit dem Signing Key von Microsoft Azure, welcher im Juli 2023 als gestohlen gemeldet wurde: So konnte eine Menschenrechtsorganisation, die von Microsoft alarmiert wurde, in ihren Logs keine Nachweise einer Kompromittierung finden, da die Organisation keinen Aufpreis für die dafür notwendige E5-Lizenz zahlte [15].

Aufgrund der Medienpräsenz, heftiger Reaktionen von Cloud Security Forschenden sowie Druck der amerikanischen Cybersecurity and Infrastructure Security Agency (CISA) auf diese sogenannte *"Logging Tax"* gab Microsoft kurze Zeit später bekannt, dass diese Logs für alle Kunden kostenlos zugänglich gemacht, sowie die Standard Aufbewahrungsdauer der Logs von 90 auf 180 Tage angepasst werde [16].

3.1.3 API Abhängigkeit

Im Rahmen der Analyse von Incident Response in Cloud-Umgebungen ist die Abhängigkeit von Application Programming Interfaces (APIs) ein wesentliches Element, das sich deutlich von traditionellen on-premise Umgebungen unterscheidet. APIs spielen in Cloud-Umgebungen eine zentrale Rolle, da sie die Interaktion zwischen verschiedenen Cloud-Diensten und dem Nutzer ermöglichen. Sie fungieren als Schnittstellen für die Automatisierung von Sicherheitsprozessen, die Bereitstellung von

Echtzeit-Sicherheitsdaten und die Integration von Drittanbieter-Sicherheitstools. Diese Abhängigkeit von APIs erweitert das Spektrum der Incident Response Möglichkeiten, indem sie schnelle und flexible Reaktionen auf Sicherheitsvorfälle ermöglicht. Beispielsweise können APIs verwendet werden, um automatisch Sicherheitsbedrohungen zu identifizieren und entsprechende Abwehrmaßnahmen zu initiieren.

Im Gegensatz dazu basieren Incident Response-Prozesse in on-premise Umgebungen häufig auf manuellen Interventionen und weniger integrierten Sicherheitssystemen. Die fehlende Flexibilität und Automatisierung kann zu längeren Reaktionszeiten auf Sicherheitsvorfälle führen. In on-premise Umgebungen sind die Organisationen komplett für die Entwicklung, Implementierung und Wartung der Incident Response-Infrastruktur verantwortlich, was oft eine größere Herausforderung darstellt. Die eingeschränkte Skalierbarkeit und Anpassungsfähigkeit in on-premise Systemen kann zudem das Risiko erhöhen, dass Sicherheitsvorfälle nicht effizient erkannt oder behoben werden.

Allerdings wird die Abhängigkeiten an die Verfügbarkeit von APIs zu einem weiteren Risiko. Incident Response Tätigkeiten in der Cloud sind meist auf sogenannte *Control Plane* APIs angewiesen. Die Control Plane im Kontext von Cloud-Computing bezieht sich auf eine Schlüsselkomponente, die für die Verwaltung und Steuerung der gesamten Cloud-Infrastruktur verantwortlich ist. Beispiele für relevante Control Plane Aktivitäten während der Reaktion auf Sicherheitsvorfälle in Cloud Umgebungen sind:

- Anpassen von Firewall Regeln (z.B. Blocken bestimmter IP-Adressen) als Reaktion auf identifizierten Angreifer.
- Deaktivieren geleakter Access Keys, um unautorisierten Zugriff zu entziehen.
- Erstellen von Disk Snapshots kompromittierter Server, um nachgelagerte forensische Aktivitäten zu ermöglichen.
- Skalieren weiterer Ressourcen als Reaktion auf einen Distributed Denial of Service (DDoS) Angriff.
- Isolieren von betroffenen Systemen, um die weitere Verbreitung zu verhindern.

3.1.4 Einfluss von SLAs

Ein Service Level Agreement (SLA) im Kontext von Cloud Computing ist ein vertraglich festgelegtes Dokument, das die vereinbarten Dienstleistungsstandards zwi-

schen einem Cloud-Anbieter und dem Cloud-Nutzer definiert. Es spezifiziert in der Regel die Qualität und den Umfang der zu erbringenden Services, einschließlich Aspekte wie Systemverfügbarkeit (Uptime), Leistungsparameter, Datenmanagement, Sicherheitsstandards und Support-Details.

Hauptmerkmale eines Cloud-Computing SLAs umfassen:

- **Verfügbarkeit und Zuverlässigkeit**: Hier wird festgelegt, wie verfügbar der Dienst sein soll (oft als Prozentsatz der Zeit, z.B. 99,9% Verfügbarkeit) und welche Maßnahmen bei Ausfällen ergriffen werden.
- **Leistungsbenchmarks**: Diese beinhalten spezifische Leistungsindikatoren wie Antwortzeiten, Bandbreite und Verarbeitungsgeschwindigkeit.
- **Datensicherheit und Datenschutz**: Hier werden die Maßnahmen und Standards beschrieben, die der Anbieter ergreifen muss, um die Sicherheit und den Schutz der Kundendaten zu gewährleisten.
- **Notfallwiederherstellung und Business Continuity**: Dieser Teil des SLA beschreibt, wie Daten im Falle eines Ausfalls oder einer Katastrophe wiederhergestellt werden und wie die Geschäftskontinuität aufrechterhalten wird.
- **Kundensupport und Service-Management**: Hier werden die Art und Weise des Supports, die Reaktionszeiten und das Verfahren zur Handhabung von Serviceanfragen oder Beschwerden festgelegt.
- **Strafen bei Nichteinhaltung**: Dieser Abschnitt beschreibt die Kompensationen oder Strafen, die im Falle der Nichteinhaltung der vereinbarten Service Levels durch den Anbieter gelten

Die definierten SLAs der Cloud-Anbieter können die Schnelligkeit und den Umfang von Supportleistungen während eines Vorfalls beeinflussen. Somit besteht hier das Risiko einer negativen Beeinflussung hinsichtlich der Geschwindigkeit als auch der Effektivität von Incident Response Maßnahmen. In Kombination mit der Abhängigkeit von APIs kann dies kritisch werden, wenn die in Kapitel 3.1.3 genannten Beispiele für Control Plane Actions aufgrund mangelnder Verfügbarkeit nicht ausgeführt werden können.

3.1.5 Dynamik der Cloud

Eines der prägendsten Merkmale des Cloud-Computings ist dessen dynamische Natur, welche auch der vierten Charakteristik *Rapid Elasticity* der NIST Definition des

Cloud Computings (vgl. Kapitel 2.2.1). Im Gegensatz zu traditionellen on-premise Umgebungen, in denen Ressourcen relativ statisch sind und Änderungen seltener vorkommen, zeichnet sich die Cloud durch ihre Fähigkeit aus, schnell auf unterschiedliche Anforderungen zu reagieren. Diese Dynamik zeigt sich auf mehrere Arten, welche die Behandlung von Sicherheitsvorfällen beeinflussen:

Schnelles Bereitstellen und Entfernen von Ressourcen In der Cloud können Ressourcen innerhalb kürzester Zeit erstellt und deprovisioniert werden. Diese Fähigkeit ermöglicht ein hohes Maß an Flexibilität und Skalierbarkeit. Beispielsweise können kompromittierte Serverinstanzen isoliert und stattdessen durch neu provisionierte Instanzen ersetzt werden. Ein weiteres Beispiel sind vorkonfigurierte Umgebungen für forensische Analysen, welche auf Abruf gestartet oder je nach benötigter Rechenleistung erweitert werden können.

Autoscaling-Funktionen Ein Schlüsselmerkmal von Cloud Umgebungen ist das Autoscaling, bei dem Ressourcen automatisch basierend auf vordefinierten Metriken wie CPU-Nutzung oder Netzwerkverkehr skaliert werden. Dies stellt sicher, dass die Infrastruktur unterschiedliche Arbeitslasten ohne manuellen Eingriff bewältigen kann, wodurch Leistung aufrechterhalten und das Risiko von Ausfallzeiten oder Überprovisionierung reduziert wird. Diese Funktionalität kann beispielsweise als Abwehrmaßnahme gegen DDoS Angriffe verwendet werden. Autoscaling Funktionen stehen zwar auch Nutzern traditioneller on-premise Systemen zur Verfügung; allerdings haben vor allem kleinere Unternehmen nicht die nahezu unbegrenzten Skalierungsmöglichkeiten, welche die großen Cloud Service Provider (CSP) anbieten können.

Kurzlebige Infrastruktur Die verwendete Infrastruktur der Cloud ist oft kurzlebig und flüchtig. Dieser Trend ist vor allem mit dem Aufstieg von *Serverless* Design Prinzipien verstärkt worden. *Serverless* bezeichnet ein Architekturmodell, bei dem die Verwaltung von Serverinfrastrukturen und Betriebssystemen vollständig vom Cloud-Anbieter übernommen wird. In einer serverlosen Architektur können Entwickler Anwendungen und Dienste erstellen und bereitstellen, ohne sich um die zugrunde liegende Infrastruktur kümmern zu müssen. Dadurch werden beispielsweise virtuelle Maschinen oder Container nur für kurze Zeit eingesetzt, um bestimmte Aufgaben auszuführen, bevor sie beendet werden. Dies steht im starken Kontrast zu traditio-

nellen Umgebungen, in denen Server und Infrastrukturelemente typischerweise eine längere Lebensdauer haben.

Die oben genannten Punkte verdeutlichen, dass Sicherheitsstrategien in Cloud Umgebungen diese Dynamik berücksichtigen müssen. Dies erfordert eine Abkehr von traditionellen, perimeterbasierten Sicherheitsmodellen hin zu flexibleren, datenorientierten Ansätzen.

Zusammenfassend führt die dynamische Natur der Cloud sowohl zu Chancen als auch zu Herausforderungen. Während sie Flexibilität, Skalierbarkeit und Effizienz bietet, erfordert sie auch einen proaktiveren und innovativeren Ansatz für die Reaktion auf Sicherheitsvorfälle und das Sicherheitsmanagement. Das Verständnis und die Nutzung der von Cloud-Anbietern bereitgestellten Tools und Dienste zur Überwachung und Verwaltung dieser dynamischen Ressourcen sind entscheidend für die Effektivität und Geschwindigkeit bei der Reaktion auf Sicherheitsvorfälle. In Kapitel 6.2.1 werden essentielle AWS Services (wie zum Beispiel *AWS Config*) vorgestellt, die in der Lage sind die Dynamik der Cloud in Incident Response Prozessen zu berücksichtigen.

3.1.6 Ständige und schnelllebige Änderungen

Cloud-Umgebung unterliegt schnellen und hochfrequenten Änderungen. Ressourcen werden nicht nur schnell erstellt und gelöscht, sondern können auch häufige Modifikationen wie Konfigurationsänderungen, Software-Updates oder Änderungen von Netzwerkzugriffsregeln [vgl. 17, Introduction]. Dies wird durch neue Softwareentwicklungsmethoden wie zum Beispiel die DevOps Philosophie oder das *Scrum* Framework verstärkt, da diese auf iterative Produktentwicklung mit hochfrequenten Software- und Feature-Releases setzen, um eine kürze *Time-to-Market (Zeit bis zur Markteinführung)* zu gewährleisten.

3.1.7 Neue Incident Response Domänen

Eine effektive Vorbereitung und Reaktion auf Sicherheitsvorfälle erfordert ein Verständnis der gängigen Arten von Sicherheitsvorfällen in der Cloud. Durch das Cloud Computing ergeben sich drei Domänen, die unter der Verantwortung der Cloud-Nutzer stehen und in den Sicherheitsvorfälle auftreten können: *Service Domain (Service Domäne)*, *Infrastructure Domain (Infrastruktur Domäne)* und *Application Domain (Applikations Domäne)*. Diese unterschiedlichen Domänen erfordern

unterschiedliche Expertise, Werkzeuge als auch Prozesse zur Reaktion auf Sicherheitsvorfälle. In den nachfolgenden Absätzen werden diese neuen Incident Response Domänen detailliert beschrieben und anhand von Beispielen erklärt.

- **Service Domain**: Sicherheitsvorfälle innerhalb dieser Domäne können beispielsweise eine Cloud Subscription (z.B. AWS-Account oder Azure Subscription), Berechtigungen oder Ressourcenmetadaten betreffen. Ein Sicherheitsvorfall in der Service Domain zeichnet sich dadurch aus, dass darauf ausschließlich mit API-Mechanismen reagiert werden kann. Zudem liegt bei Incidents innerhalb dieser Domäne die Ursache in der *Konfiguration* der Ressourcen der vom Cloud-Anbieter bereitgestellten Services. Ein Beispiel für einen Sicherheitsvorfall in dieser Domäne ist eine Datenleck, bei dem geheime oder personenbezogene Daten eines Unternehmens exfiltirert werden, indem ein fehl konfigurierter S3 Bucket ausgenutzt wird.
- **Infrastructure Domain**: Sicherheitsvorfälle innerhalb dieser Domäne umfassen daten- oder netzwerkbezogene Aktivitäten, wie Prozesse und Daten auf Cloud-Computing-Instanzen, Datenverkehr innerhalb Virtual Private Networks als auch Containerdienste. Die Reaktion auf Sicherheitsvorfälle im Infrastrukturbereich erfordert häufig das Sammeln vorfallbezogener Daten für forensische Analysen. Dies beinhaltet typischerweise Interaktionen mit dem Betriebssystem einer Cloud-Instanz und kann in verschiedenen Fällen auch die Nutzung spezifischer APIs der Cloud-Anbieter umfassen. Hierbei kann eine Kombination aus APIs der Cloud-Anbieter als auch andere Digital Forensics/Incindet Response (DFIR) Tools eingesetzt werden. Es empfiehlt sich, die Durchführung von forensischen Analysen in dedizierten Recheninstanzen durchzuführen. Beispiele für die Behandlung von Sicherheitsvorfällen in dieser Domäne sind die Analyse von Packet Captures (PCAPs) (*Mitschnitte von Netzwerkpaketen*), Festplattenspeicherblöcken oder die Auswertung von flüchtigem Speicher (Random-Access Memory (RAM)) von Server-Instanzen.
- **Application Domain**: Sicherheitsvorfälle innerhalb dieser Domäne treten im Quellcode von Anwendungen oder in der Software, welche in den Cloud Services bzw. Infrastruktur eingesetzt wird [vgl. 17, Introduction].

3.2 Gemeinsamkeiten

Im vorherigen Kapitel wurde eine Reihe von Besonderheiten und Unterschiede von Incident Response in der Cloud im Vergleich zu on-premise Umgebungen erläutert. Trotz alledem gibt es eine Vielzahl an Eigenschaften, die auch im Cloud Umfeld ihre Gültigkeit behalten.

3.2.1 Incident Response Prozess

Der allgemeine Prozess zur Reaktion auf Sicherheitsvorfälle - wie bereits in Kapitel 2.1.3 beschrieben - behält auch in Cloud Umgebungen seine Gültigkeit. Aufgrund dessen verwendet auch AWS im *AWS Security Incident Response Guide* das NIST Phasenmodell (vgl. Abbildung 2) [vgl. 17, Introduction].

3.2.2 Erforderliches Fachwissen

Trotz der fortschreitenden Automatisierung und der hochentwickelten Managementtools in Cloud-Umgebungen bleibt fundiertes Fachwissen für die Behandlung von Sicherheitsvorfällen unerlässlich. Dies liegt daran, dass Sicherheitsvorfälle in der Cloud oft komplex und vielschichtig sind, was ein tiefes Verständnis der spezifischen Cloud-Architektur, der zugrunde liegenden Technologien sowie der aktuellen Bedrohungslandschaft erfordert. Experten mit Fachkenntnissen müssen in der Lage sein, Sicherheitslogs und Warnmeldungen effektiv zu interpretieren, die oft subtile Anzeichen von Sicherheitsverletzungen beinhalten können. Darüber hinaus ist Fachwissen entscheidend für die Anpassung und Konfiguration von Sicherheitstools und -protokollen, die in der dynamischen und skalierbaren Natur der Cloud-Umgebungen funktionieren müssen. In der Praxis bedeutet dies, dass menschliche Expertise für das Erkennen, Analysieren und Reagieren auf Sicherheitsvorfälle unverzichtbar bleibt [vgl. 18, Folie 9].

3.2.3 OS Logs

In Kapitel 3.1.2 wurde bereits die Besonderheit von Logs im Cloud erläutert. Und auch wenn Logs von Control Plane Aktivitäten im Cloud Umfeld einen hohen Stellenwert haben, tragen native Operating System (OS) Logs eine große Rolle und müssen ebenfalls beschafft, überwacht und analysiert werden. Einerseits kann in

Compliance-Standards die Aufbewahrung und Analyse von Betriebssystemprotokollen als Teil des Auditprozesses gefordert werden. Diese Protokolle liefern den Nachweis, dass angemessene Sicherheitsmaßnahmen vorhanden waren und dass während eines Vorfalls bestimmte Maßnahmen ergriffen wurden. Andererseits zeichnen OS Logs detaillierte Informationen zu Systemaktivitäten auf, wie zum Beispiel Netzwerkverbindungen, Benutzeraktivitäten, Prozessausführen oder Dateizugriffe. Somit sind sie eine wichtige Informationsquelle und helfen dabei, die Ursachen des Vorfalls, das Ausmaß der Kompromittierung und die von Angreifern verwendeten Methoden zu identifizieren. Diese Informationen sind wichtig, um den Angriffsvektor zu verstehen und ähnliche Vorfälle in der Zukunft zu verhindern.

3.2.4 Grundsätze der IT-Sicherheit

Unabhängig davon, ob es sich um Cloud- oder on-premise Umgebungen handelt, bleibt Incident Response ein wesentlicher Bestandteil einer IT-Sicherheitsstrategie. Grundlegende Prinzipien wie zum Beispiel *Least Privielge* - die Vergabe lediglich derer Berechtigungen, welche für die erfolgreiche Ausführung von Tätigkeiten erforderlich sind - behalten auch in Cloud-Umgebungen ihre Gültigkeit und Relevanz, da sie bei der Prävention von Sicherheitsvorfällen beitragen [vgl. 17, Introduction].

4 Konzeption der Incident Response Szenarien

Zu Beginn dieses Kapitels werden verschiedene Incident Response *Szenarien* konzipiert, die - basierend auf Veröffentlichung von Auswertungen des AWS Customer Incident Response Teams, jährlichen Reports von Cloud Security Forschenden sowie auf Basis veröffentlichter Sicherheitsvorfälle in Organisationen - zu den gängigsten Arten von Sicherheitsvorfällen zählen, von denen Organisation mit IT-Landschaften in der Cloud konfrontiert sind. Deshalb ist die Wahrscheinlichkeit sehr hoch, dass Unternehmen mit IT-Systemen in public Cloud Umgebungen sich diesen Sicherheitsvorfällen ausgesetzt sehen.

Die einzelnen Szenarien werden in den nachfolgenden Abschnitten in diesem Kapitel anhand des typischen Vorgehens, der verwendeten Angriffsvektoren und der vorhandenen Schwachstellen und/oder Miskonfigurationen, welche ausgenutzt werden, im Detail beschrieben.

Motivation für die Verwendung von Incident Response Szenarien Die gewählten Szenarien eignen sich, um die Wichtigkeit bestimmter Incident Response *Capablitites (Fähigkeiten)* von Organisationen hervorzuheben sowie diese gegebenenfalls auf- oder auszubauen.

Im *NIST Special Publication 800-61 Rev. 2* wird betont, dass die Entwicklung und Verwendung von Incident Response Szenarien für Unternehmen von großer Bedeutung sind. Die NIST empfiehlt zusätzlich zu den praktischen Szenarien eine Reihe von begleitenden Fragen zu stellen, die während der Behandlung von Sicherheitsvorfällen zu klären sind. Solche Szenarien bieten eine kosteneffiziente und effektive Methode, um die Fähigkeiten von Incident Response Teams zur Reaktion auf Sicherheitstorfälle zu prüfen und auszubauen sowie potenzielle Schwachstellen in bestehenden Prozessen aufzudecken.

Durch die Konfrontation eines Incident Response Teams mit ausgewählten Szenarien wird nicht nur das theoretische Wissen der Teammitglieder geprüft, sondern auch deren praktische Anwendungsfähigkeiten in realistischen Kontexten geschärft.

Beispielsweise könnte die Durchführung eines Szenarios aufzeigen, dass eine Verzögerung in der Reaktion auftritt, weil dem Team bestimmte Tools und Logs fehlen oder weil ein anderes Team, welches an der Behandlung des Sicherheitsvorfalls beteiligt ist, aufgrund der geltenden SLAs außerhalb der Geschäftszeiten keinen Support bietet [vgl. 2, Appendix A—Incident Handling Scenarios].

Auch das BSI stellt im IT-Grundschutzkompendium (GSK) Baustein DER 2.1 gewisse Anforderungen als Teil der „Überprüfung der Effizienz des Managementsystems zur Behandlung von Sicherheitsvorfällen“. Dies sollte laut BSI sowohl durch regelmäßige angekündigte als auch unangekündigte Übungen sowie Planspiele zur Behandlung von Sicherheitsvorfällen durchgeführt werden [vgl. 3, DER.2.1.A22].

4.1 Szenario I: Compromised Credentials

Das erste zu betrachtende Szenario handelt von *Compromised Credentials*, sprich der Kompromittierung von Anmeldedaten. Abbildung 7 gibt einen Überblick über das Szenario und stellt mögliche Ursachen dar, zeigt wie diese von Angreifern ausgenutzt werden könnten und welche Auswirkungen dies zur Folge hätte.

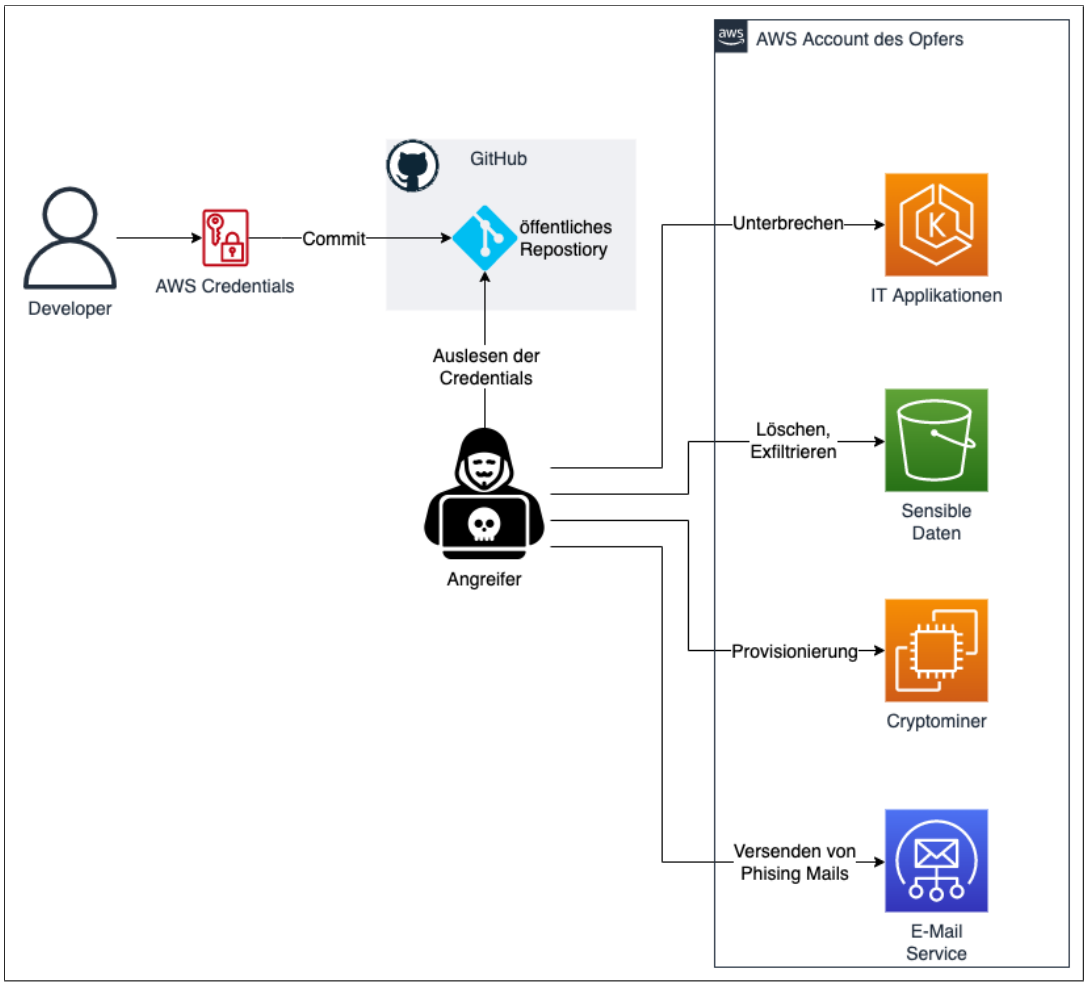


Abbildung 7: Szenario I - Überblick

4.1.1 Beschreibung

Im Kontext des Public Cloud Computing stellt das Szenario *Kompromittierung von Anmeldeinformationen* eine der bedeutendsten Bedrohungen für die Sicherheit und Integrität von Cloud-Diensten dar. Dieses Szenario umfasst Situationen, in denen Angreifer auf verschiedenen Wegen Zugriff auf Cloud-basierte Anmeldeinformationen erlangen. Dieses Szenario stellt das erste Einfallstor für potentielle Angreifer dar. Im MITRE ATT&CK Framework für Cloud Umgebungen fällt dieses Szenario in die Kategorie der Techniken für *Initial Access* [20].

Von diesem Incident Response Szenario lassen sich noch viele weitere Szenarien für Sicherheitsvorfälle, wie zum Beispiel Datenlecks oder Unterbrechung der Verfügbarkeit von Applikation, ableiten.

4.1.2 Ursachen

In der Cloud Computing Landschaft sind kompromittierte Anmeldeinformationen ein häufig auftretendes Sicherheitsproblem, dessen Ursachen vielfältig und komplex sein können. Einer der gängigsten Ursachen für die Kompromittierung der Anmeldeinformationen tritt durch sogenannte *Exposed Credentials* - also offengelegte Anmeldeinformationen - auf. Eine primäre Ursache ist oft die unsachgemäße Handhabung und Speicherung von Anmeldeinformationen, wie das hart codierte Hinterlegen von API-Schlüsseln und Passwörtern in ungeschützten, öffentlich zugänglichen Repositories. Dies kann durch menschliches Versagen oder Unkenntnis über sichere Praktiken ausgelöst werden.

Die wohl größte Quelle für Exposed Credentials sind öffentliche git Repositories, wie zum Beispiel *GitHub*. Das Unternehmen GitGuardian, welches sich auf die Detektion von *secrets* (API-Keys, Passwörter, Verschlüsselungsschlüssel etc.) in Quellcode spezialisiert, veröffentlichte im *The State of Secrets Sprawl 2023* die folgenden Zahlen für offengelegte Anmeldeinformationen: alleine im Jahr 2022 wurden 3 Millionen einzigartige *secrets* entdeckt, die in über 10 Millionen öffentlichen GitHub commits auftauchten. Des Weiteren sind Anmeldeinformationen für Cloud-Anbieter die dritthäufigste Kategorie mit 20,2% aller erkannten secrets. Das sind knapp 600.000 Fälle bzw. rund 68 öffentliche commits mit secrets pro Stunde [21, S. 7–10].

Eine weitere Quelle für unbeabsichtigt veröffentlichte Anmeldeinformationen sind Package Indizes für Programmiersprachen, wie zum Beispiel Python Package Index (PyPi). Ein britischer Forscher fand 2023 insgesamt 57 *gültige* AWS Access Keys, indem er alle Packages in PyPi untersuchte. Die Tatsache, dass zu den betroffenen Organisation auch AWS zählte, zeigt, dass unbeabsichtigt veröffentlichte Credentials ein weitverbreitetes Problem sind [22].

4.1.3 Auswirkungen

Die kompromittierten Anmeldeinformationen zu Cloud Accounts können Angreifern unbefugten Zugang zu sensiblen Daten und Ressourcen verschaffen. Angreifer können versuchen Daten zu manipulieren, zu löschen oder zu exilftrieren. Des Weiteren versuchen Angreifer häufig Recheninstanzen für sogenannte Cryptominer in den kompromittierten Cloud Umgebungen zu provisionieren. Eine weitere Art von weiterführenden Sicherheitsvorfällen ist die Zerstörung von IT-Systemen, welche sich in den kompromittieren Cloud Accounts befinden, indem laufende Server terminiert

werden. Zudem kann dies auch dazuführen, dass die kompromittierten Ressourcen für DDoS Angriffe oder das Versenden von Phishing Mails verwendet werden. Somit kann diese Art von Sicherheitsvorfällen weit reichenden Einfluss auf die Sicherheitsziele Vertraulichkeit, Verfügbarkeit, Integrität als auch Authentizität haben.

4.1.4 Empirische Relevanz

Die Relevanz dieses Szenarios lässt sich anhand zahlreicher empirischer Untersuchungen verdeutlichen:

1. **Betonung des Risikos des Szenarios in den *Top Threats to Cloud Computing - The Egregious 11***: Bei der Cloud Security Alliance (CSA) handelt es sich um eine Vereinigung aus Organisation, deren Ziel die Verbesserung der Sicherheit im Cloud Umfeld ist. Sie wurde 2008 gegründet und wird von den größten Cloud Anbietern (AWS, GCP, Microsoft und VMware) unterstützt. Unter anderem veröffentlicht die CSA in regelmäßigen Abständen eine Liste der verbreitetsten *Cloud Threats (Bedrohungen)* [11, S. 22].

   Kompromittierte Anmeldedaten fallen dabei in die Kategorien *Insufficient Identity, Credential, Access and Key Management* als auch *Account Hijacking* [23, S. 16–21].

2. ***State of Cloud Security 2023***: Im Report zur Sicherheitslage in der Cloud von Datadog aus 2023 wird betont, dass langlebige Anmeldedaten (*long-lived credentials*) sowie fehlende Absicherung von Anmeldedaten durch Multi-Faktor-Authentisierung (MFA) zu den häufigsten Ursachen von Security Breaches darstellen [24, Fact 1 & Fact 2].

3. ***Cost of a Data Breach Report 2023***: IBM veröffentlichte 2023 einen Bericht zu den durchschnittlichen Kosten, welche mit Sicherheitsvorfällen einher gehen. Darin stellten *compromised credentials* in 15% der Fälle den zweitgrößten Angriffsvektor dar [25, S. 20].

4.2 Szenario II: Compromised S3 Buckets

Das zweite zu betrachtende Incident Response Szenario handelt von *Compromised Amazon Simple Storage Service (S3) Buckets*, sprich der Kompromittierung von S3 Buckets.

S3 ist ein skalierbarer und hochverfügbarer Cloud-Speicherdienst von AWS, der es Cloud-Nutzern ermöglicht, große Mengen von Daten in sogenannten *Buckets* zu speichern und darauf zuzugreifen [26].

Es ist einer der ältesten Services, den AWS seit März 2006 anbietet. 2021 gab AWS bekannt, dass zu diesem Zeitpunk bereits über 100 Billionen (10^{14}) Datenobjekte in S3 gespeichert wurden und der Service regelmäßig Belastungsspitzen mit Requests pro Sekunde im zweistelligen Millionenbereich erreichte [27].

Abbildung 8 zeigt einen Überblick des Szenarios mit möglichen Ursachen sowie potentielle Angriffsmöglichkeiten.

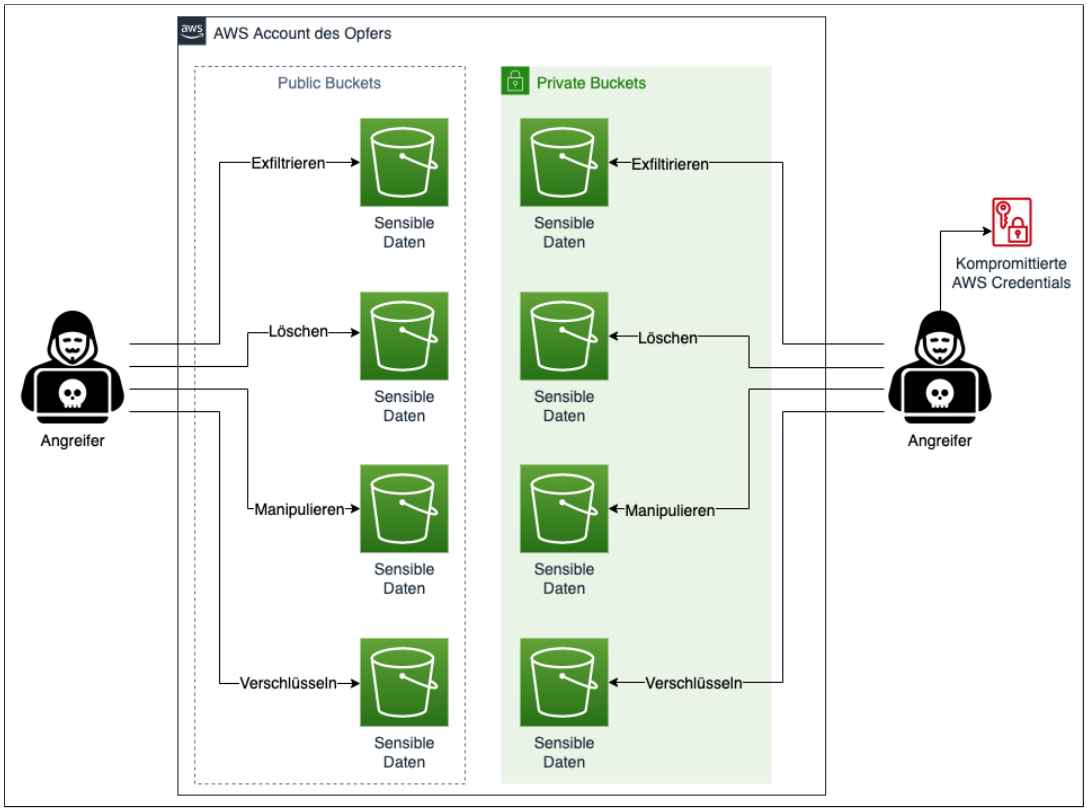


Abbildung 8: Szenario II - Überblick

4.2.1 Beschreibung

Das Szenario *Compromised S3 Buckets* bezieht sich auf Sicherheitsvorfälle innerhalb der AWS Cloud-Infrastruktur, bei denen S3 Buckets betroffen sind. Diese Vorfälle sind durch den unbefugten Zugriff auf, sowie der Löschung, Veränderung oder Exfiltration von Daten aus S3 Buckets gekennzeichnet.

4.2.2 Ursachen

Die Hauptursachen für kompromittierte S3 Buckets lassen sich in zwei Kategorien unterscheiden, welche beide in den Verantwortungsbereich der Cloud-Nutzer gemäß des Shared Responsibility Model (vgl. Abbildung 5) fallen:

Miskonfiugration der S3 Buckets

Die erste Kategorie bezieht sich auf die Miskonfiguration der S3 Buckets. Dabei führen getroffene Konfigurationen der Cloud-Nutzer am S3 Bucket zu schwachen bzw. nicht vorhandenen Zugriffskontrollen. Dies kann dazu führen, dass Cloud-Nutzer unbeabsichtigt ihre S3 Buckets öffentlich zugänglich machen - in diesem Fall spricht man von sogenannten *Public Buckets*.

Ausnutzen kompromittierter Anmeldeinformationen

Die zweite Kategorie bezieht sich auf die Ausnutzung kompromittierter Anmeldeinformationen (vgl. hierzu Kapitel 4.1). Denn selbst wenn Cloud-Nutzer den Zugriff auf ihre S3 Buckets korrekt steuern und somit nicht öffentlich zugänglich machen, können kompromittierte Anmeldeinformationen von Angreifern ausgenutzt werden, um damit auf sensible Daten in den Cloud Umgebungen ihrer Opfer zuzugreifen.

4.2.3 Auswirkungen

AWS Buckets sind Infrastrukturressourcen in AWS, die für die Speicherung von großen Menge an Daten verwendet werden. Die Integrität und Sicherheit dieser Buckets ist von entscheidender Bedeutung für Organisationen, da sie oft sensible Informationen enthalten. Ein kompromittierter AWS Bucket bedeutet, dass die Sicherheitsmaßnahmen, die zum Schutz der darin gespeicherten Daten implementiert wurden, umgangen oder außer Kraft gesetzt wurden. Dies kann zu weiteren verschiedenen sicherheitsrelevanten Ereignissen führen:

1. **Datenexfiltration**: Unbefugte Entwendung von Daten, bei der sensible Informationen aus dem Bucket kopiert und an einen externen Standort übertragen werden.
2. **Datenlöschung**: Unautorisierte Entfernung oder Vernichtung von Daten, was zu Datenverlust oder Betriebsstörungen führen kann.

3. **Datensabotage**: Modifikation von Daten im Bucket, um Fehlinformationen zu verbreiten oder die Integrität der Daten zu beeinträchtigen.
4. **Datenverschlüsselung**: Verschlüsselung von Daten im Bucket mit anschließender Löschung des Schlüssels.

Allerdings sei an dieser Stelle erwähnt, dass Angriffe auf die Verfügbarkeit von Daten mittels Verschlüsselung durch Angreifer - was eine gängige Technik von Ransomware Angriffen in on-premise Landschaften darstellt - in Cloud-Umgebungen bisher noch berichtet wurde. Zwar haben Gietzen in *S3 Ransomware Part 1: Attack Vector* und Traxler in *Cloud-Native Ransomware* einen theoretischen Angriffsvektor erarbeitet. Doch dieser konnte auch vier Jahre nach der ersten Veröffentlichung bisher nicht nachgewiesen werden. Deshalb wird dieser theoretische Angriffsvektor der Datenverschlüsselung eines S3 Buckets in den nachfolgenden Betrachtungen ausgelassen.

4.2.4 Empirische Relevanz

Die Relevanz dieses Szenarios lässt sich sowohl anhand empirischer Untersuchungen als auch zahlreichen veröffentlichten Vorfällen in Organisationen hervorheben:

1. **Betonung des Risikos des Szenarios in den *Top Threats to Cloud Computing - The Egregious 11***: Die CSA führt in ihrem Bericht der elf größten Sicherheitsrisiken im Cloud Computing aus dem Jahr 2020 das Risiko von *Data Breaches* (Datenschutzverletzung) an erster Stelle auf. Die CSA schließt damit alle Arten von Informationen, die nicht zur Veröffentlichung bestimmt waren, einschließlich – aber nicht beschränkt auf – persönliche Gesundheitsinformationen, Finanzinformationen, personenbezogene Daten, Geschäftsgeheimnisse und geistiges Eigentum [23, S. 7–9]. Kompromittierte S3 Buckets fallen in diese Kategorie.
2. **Public S3 Bucket Kompromittierung von genetischen und gesundheitsrelevanten Daten durch *Vitagene***: *Vitagene* (inzwischen *1Health.io*) ist eine US-amerikanische Firma, die sich auf die Sequenzierung von DNA im Verbraucherbereich spezialisiert. Die US-amerikanische Bundesbehörde Federal Trade Commission (FTC) reichte im Juni 2023 Beschwerde gegen *Vitagene* aufgrund schwerwiegender Datenschutzverletzungen ein. Das Unternehmen erstellte gegen 2016 die ersten Public Buckets und wurde 2017 darüber von AWS informiert. Eine eigens von *Vitagene* beauftragter Penetration Test fand zudem heraus, dass DNA Datensätze in S3 Buckets öffentlich zugänglich

waren. Im Juni und Juli 2019 wurde das Unternehmen mehrfach von externen Forschenden über diesen Zustand informiert. Diese Umstände führten zur Veröffentlichung von Gesundheits- und Genetikdaten von über 2.600 Kunden [30].

3. **Private Bucket Kompromittierung der *No Fly List* durch *Commute Air***: *Commute Air* ist eine US-amerikanische Fluggesellschaft. Im Januar 2023 wurde bekannt, dass die Kompromittierung eines S3 Bucket von *Commute Air* zur Offenlegung der sogenannten *No Fly List* des US Department of Homeland führte. Dabei handelte es sich nicht um einen Public Bucket. Allerdings konnten über einen Jenkins Server Anmeldeinformationen kompromittiert werden (vgl. Szenario 1), mit deren Hilfe die Angreifer auf die AWS Infrastruktur von *Commute Air* zugreifen. Davon waren über 1,5 Millionen Datensätze von Personen betroffen [31].

4. **Manipulation von Daten eines S3 Buckets der *LA Times***: 2018 entdeckten ein Security Forschende ein sogenanntes *crypo-jacking script* - dabei wird Rechenleistung im Browser von Kunden benutzt, um Crypto-Währungen zu schürfen - auf der Website der US-amerikanischen Tageszeitung *LA Times*. Dieser Vorfall ist ein Beispiel wie ein kompromittierter S3 für die Manipulation von Daten genutzt wurde, da die Angreifer durch Miskonfigurationen der *LA Times* in der Lage waren die crypto-jacking scripts zu injizieren [32].

4.3 Szenario III: Cryptomining Activities

Das dritte zu betrachtende Incident Response Szenario befasst sich mit *Cryptomining Activities*, also dem Schürfen von digitalen Crypto-Währungen durch Angreifer unter der Verwendung von Rechenleistung auf Infrastrukturressourcen der Cloud-Nutzer.

Abbildung 9 zeigt einen Überblick des Szenarios mit möglichen Ursachen sowie potentielle Angriffsmöglichkeiten.

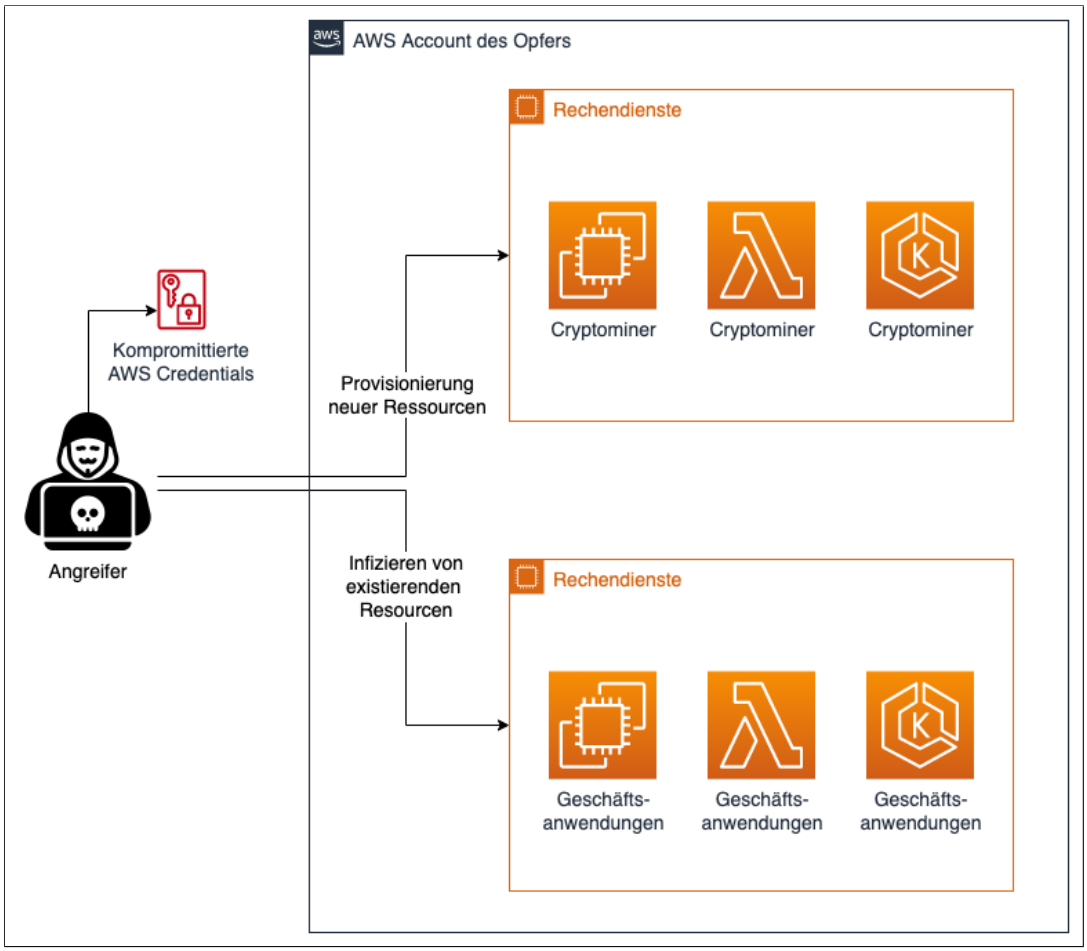


Abbildung 9: Szenario III - Überblick

4.3.1 Beschreibung

Cryptomining-Angriffe in Cloud-Umgebungen stellen eine spezielle Form von Sicherheitsvorfällen da, bei welchen Angreifer unbefugt die Rechenressourcen von Cloud-Nutzern kapern, um digitale Crypto-Währungen zu schürfen.

4.3.2 Ursachen

Je nachdem in welcher Form die ungewollten Cryptomining Aktivitäten auftreten, unterscheiden sich die Hauptursachen:

Provisionierung neuer Ressourcen

In diesem Fall erstellen Angreifer neue Infrastrukturessourcen für Rechendienste, um darauf Cryptominer zu platzieren. Dabei werden häufig EC2 Instanzen aber auch anderen Services für Rechendienste, wie zum Beispiel Pods innerhalb Kubernetes Cluster, *Amazon Sagemaker* Instanzen oder auch Prozesse in *AWS Lambda* Funktionen ausgeführt. Dies geschieht unter Ausnutzung kompromittierter Anmeldeinformationen (vgl. hierzu Kapitel 4.1).

Infizieren existierender Ressourcen

Bei der zweiten Kategorie von Cryptomining Sicherheitsvorfällen werden keine neuen Infrastrukturessourcen für Rechendienste erstellt, sondern bestehende Ressourcen infiziert. Dabei können die gleichen Rechendienste wie in der oben beschriebenen Kategorie ausgenutzt werden. Dies kann unter der Ausnutzung kompromittierter Anmeldeinformationen (vgl. hierzu Kapitel 4.1) oder von Netzwerkschwachstellen geschehen, die die Injektion von Schadcode erlauben.

4.3.3 Auswirkungen

Die Folgen von Cryptomining Sicherheitsvorfällen für Organisationen können - je nachdem, ob neue Ressourcen von den Angreifern erstellt werden oder bestehende Ressourcen verwendet werden - vielfältig sein:

1. **Erhöhte Kosten**: Cryptomining verbraucht erhebliche Rechenressourcen. In einer Cloud-Umgebung führt dies zu erhöhtem Ressourcenverbrauch, was sich in höheren Kosten für die Cloud-Nutzer niederschlägt.
2. **Reduzierte Leistung**: Da Cryptomining CPU- und GPU-Ressourcen intensiv nutzt, kann es zu einer deutlichen Verringerung der Leistungsfähigkeit der Infrastruktur kommen.
3. **Eingeschränkte Ressourcenverfügbarkeiz**: Die für das Mining verwendeten Ressourcen stehen nicht für legitime Geschäftsanwendungen zur Verfügung, was zu Betriebsstörungen führen kann

All diese Punkte wirken sich auf die finanzielle Seite von Organisationen aus. Entweder in direkter Form über erhöhte Kosten für Cloud-Infrastruktur oder aber in indirekter Form durch das Beeinträchtigen von Geschäftsanwendungen oder Betriebsstörungen.

4.3.4 Empirische Relevanz

Die Relevanz dieses Szenarios lässt sich sowohl anhand empirischer Untersuchungen als auch zahlreichen veröffentlichten Vorfällen in Organisationen hervorheben:

1. **Betonung des Risikos des Szenarios in den *Top Threats to Cloud Computing - The Egregious 11***: Die CSA führt in ihrem Bericht der elf größten Sicherheitsrisiken im Cloud Computing aus dem Jahr 2020 das Risiko von *Abuse and Nefarious Use of Cloud Services* auf, womit der Missbrauch und schändliche Nutzung von Cloud-Diensten gemeint sind. Dazu gehört auch das *Mining* von digitalen Währungen [23, S. 38–44].
2. **Auswertung von Sicherheitsvorfällen durch AWS**: Das AWS Customer Incident Response Team gab 2022 bekannt, dass Cryptomining Aktivitäten in Accounts der Cloud-Nutzer zu den häufigsten Sicherheitsvorfällen gehören, die es behandelt [33].
3. **Erkenntnisse aus dem *The State of Cloud Security Report 2022***: *Snyk* ist ein Unternehmen, welche eine „Developer Security Platform“ anbietet und sich auf Cloud Security spezialisiert. In ihrem Bericht aus dem Jarh 2022 veröffentlichten sie ihre Auswertung, laut welcher 23% aller Sicherheitsvorfälle von Kunden im Cloud Bereich in die Kategorie *Cryptomining* fielen [34, S. 4].
4. **Cryptomining Vorfall bei Uber Tochtergesellschaft**: *Drizly LLC* ist eine Tochtergesellschaft von *Uber Technologies Inc.* und betreibt eine E-Commerce Plattform für Alkoholbestellungen. 2022 wurde in einer Beschwerde der US-amerikanischen Bundesbehörde bekanntgegeben, dass das Unternehmen als Folge kompromittierter Anmeldeinformationen Opfer von Cryptomining Aktivitäten in ihrer AWS Umgebung wurden [35].
5. **Cryptomining Vorfall bei *DXC***: Das US-amerikanische IT-Beratungs- und Dienstleistungsunternehmen *DXC* wurde 2017 Opfer einer Cryptomining Attack auf deren AWS Infrastruktur. Innerhalb von vier Tagen wurden in Summe 244 Cryptominer provisioniert, wodurch ein Schaden durch zusätzliche Cloud Kosten in Höhe von 64.000 USD entstand. Die Ursache waren hart codierte

AWS Credentials, die in einem öffentlichen Repository bereitgestellt wurden [36].

5 Aufbau des weiteren Vorgehens

In dieser Master-Thesis wird die Integration von cloud-native Services innerhalb der einzelnen Phasen des Incident Response Prozesses betrachtet. Wie zuvor in Kapitel 2.1.3 erläutert, wird dabei das Phasenmodell der *NIST Special Publication 800-61 Rev. 2* aufgrund der weitläufigen Verbreitung des Modells verwendet (vgl. Abbildung 2).

Außerdem beziehen sich die Betrachtungen der cloud-native Services speziell auf die Services, welche von Amazon Web Services (AWS) angeboten werden, da es sich hierbei um den Cloud-Anbieter mit dem größten Marktanteil handelt. Im zweiten Quartal 2023 betrug der Marktanteil von AWS 32%. Dies entspricht fast dem kombinierten Marktanteil von Microsoft Azure und GCP, welche respektive bei 22% und 11% lagen [19].

Die Konzeption der cloud-nativen Incident Response wird in dieser Master-Thesis in zwei Kategorien unterteilt: *allgemeingültige* und *szenariospezifische* Aktivitäten.

5.1 Betrachtung der allgemeingültige Aktivitäten

Die allgemeingültigen Aktivitäten umfassen ausschließlich vorbereitende Maßnahmen beim Einsatz cloud-nativer Services, die unabhängig der gewählten Incident Response Szenarien anwendbar und empfehlenswert sind.

Das Phasenmodell der *NIST Special Publication 800-61 Rev. 2* unterteilt die *Prepare*-Phasen in zwei Kategorien:

1. **Preparing to handle incidents**: Hierzu zählen alle Maßnahmen, Aktivitäten und Werkzeuge, welcher der allgemeinen *Vorbereitung auf die Behandlung* von Sicherheitsvorfällen dienen.
2. **Preventing incidents**: Hierzu zählen alle Maßnahmen, Aktivitäten und Werkzeuge, welcher der *Prävention des Eintretens* von Sicherheitsvorfällen dienen [vgl. 2, S. 21–23].

Die Integration und Konfiguration bestimmter AWS Services als Teil der *Preparing to handle Incidents*-Phase, haben einen weitestgehend allgemeingültigen Anwendungsbereich. Sie sind somit unabhängig von den Incident Response Szenarien anwendbar und werden deshalb in einem einzelnen Kapitel (vgl. Kapitel 6) betrachtet.

5.2 Betrachtung der szenariospezifische Aktivitäten

Da sich die szenariospezifischen Aktivitäten innerhalb der restlichen Phasen der Behandlung von Sicherheitsvorfällen - sprich in den Phasen *Detection & Analysis* sowie *Containment, Eradication & Recovery* - je nach Art des Vorfalls stark unterscheiden können, wird die Integration cloud-nativer Services in diesen Phasen für jedes verschiedene Szenario in einem gesonderten Kapitel betrachtet. Das gleiche gilt für die Prävention des Eintretens bzw. der Vermeidung von Incidents, weshalb diese auch für die einzelnen Szenarien betrachtet werden.

Für jedes Szenario werden dabei verschiedene cloud-native Services vorgestellt, die alleinstehend oder in Kombination anderer Services, für die Durchführung der jeweiligen Aktivitäten in den einzelnen Phasen des Incident Response Zyklus integriert und genutzt werden können.

6 Allgemeingültige Services in der Vorbereitung auf Sicherheitsvorfälle

In diesem Kapitel wird die Integration relevanter AWS Services als Teil der *Prepare*-Phase des NIST Incident Response Phasenmodell betrachtet. Dabei liegt der Fokus auf Vorbereitungen, die Organisationen unabhängig von den später betrachtenden Incident Response Szenarien treffen sollten. Dies betrifft insbesondere das Teilgebiet *Preparing to handle Incidents*.

6.1 Einrichten einer AWS Account Struktur

6.1.1 Begriffserklärung

Eine wichtige Vorkehrung, die Unternehmen in der Vorbereitung auf Sicherheitsvorfälle treffen sollten, ist die Einrichtung einer Account Struktur ihrer AWS Umgebung.

Ein AWS Account dient zum Einen als logischer Container für Infrastrukturressourcen und Services. Alle Ressourcen, die von Cloud-Nutzern erstellt und genutzt werden, sind eindeutig durch einen sogenannten Amazon Resource Name (ARN) identifiziert, der die Account-ID enthält, in dem sich die Ressource befindet. Außerdem erfüllen AWS Accounts eine Funktion als grundlegende Sicherheitsgrenze für Cloud Ressourcen, da diese standardmäßig nicht für Nutzer aus anderen Accounts zugänglich sind [37].

6.1.2 Zweck und Nutzen

Für Unternehmen empfiehlt es sich eine AWS Account Struktur aufzubauen, in der Applikationen logisch isoliert sind. So ist eine Empfehlung, dass die verschiedenen Umgebungen einer Applikation (zum Beispiel *Dev*, *Test* und *Prod*) in jeweils eigenen, dedizierten AWS Accounts aufgebaut werden. Das hat zum einen den Vorteil, dass Kosten verbrauchsgerecht aufgeteilt werden. Aus Sicht der IT Sicherheit bietet dies

den weiteren Vorteil, dass der sogenannte *Blast Radius* minimiert wird. So sorgen die Grenzen der Accounts dafür, dass im Falle einer Account Kompromittierung die Auswirkungen eingedämmt werden und nicht über die Account Grenzen reichen.

6.1.3 Aufbau mittels AWS Organizations

Ein wichtiger AWS Service für den Aufbau einer Account Struktur ist *AWS Organizations*. Dabei handelt es sich um einen Service zur Verwaltung mehrerer AWS Accounts und der Konsolidierung zu einer sogenannten *Organization*. Durch diesen Service können mehrere AWS Accounts zu sogenannten Organizational Units (OUs) gruppiert und zentral verwaltet werden [38, What is AWS Organizations?]. Dadurch lassen sich hierarchische Strukturen abbilden, um beispielsweise organisatorische Strukturen nach Fachbereichen eines Unternehmens widerzuspiegeln.

Ein nützlich Feature von *AWS Organizations* sind sogenannte Service Control Policys (SCPs), wodurch die maximalen Berechtigungen innerhalb eines Accounts festgelegt werden können. Dadurch können bestimmte Services oder Aktionen auf der gesamten Account oder OU Ebene gesteuert werden [38, What is AWS Organizations?]. Dies ist zum Beispiel sinnvoll, um

- Compliance Anforderungen umzusetzen, indem nur gewisse geografische Regionen freigeschaltet werden.
- Hohe Kosten zu vermeiden, indem bestimmte AWS Services oder Instanztypen blockiert werden.
- Änderungen an kritischen Infrastrukturkomponenten - zum Beispiel innerhalb der Accounts der *Prod* Umgebung - zu verhindern.

6.1.4 Empfehlungen

AWS empfiehlt in ihrer *AWS Security Reference Architecture* die Verwendung einer hierarchischen Struktur, bei der Accounts in dedizierte OUs für die Erfüllung bestimmter Funktionalitäten gruppiert werden. Beispielweise sollte ein zentraler Account eingerichtet werden, der für die Speicherung und Verwaltung aller Logs aus allen Accounts innerhalb einer Organization dient. Zudem ist es sinnvoll eine OU für Security Accounts einzurichten. Darin können dann beispielsweise ein Account nur für forensische Analyse und ein weiterer Account, der als *Delegated Administrator*

für die AWS Security Services konfiguriert ist (dazu mehr in Kapitel 6.3), enthalten sein [39, Dedicated accounts structure].

Abbildung 10 zeigt, wie eine Multi-Account Landschaft eines Unternehmens hierarchisch nach OUs gegliedert werden kann. Dabei werden dedizierte Accounts für die zentrale Speicherung von Logs, der Entwicklung von Security Tooling sowie ein eigenständiger Account für forensische Untersuchungen eingerichtet.

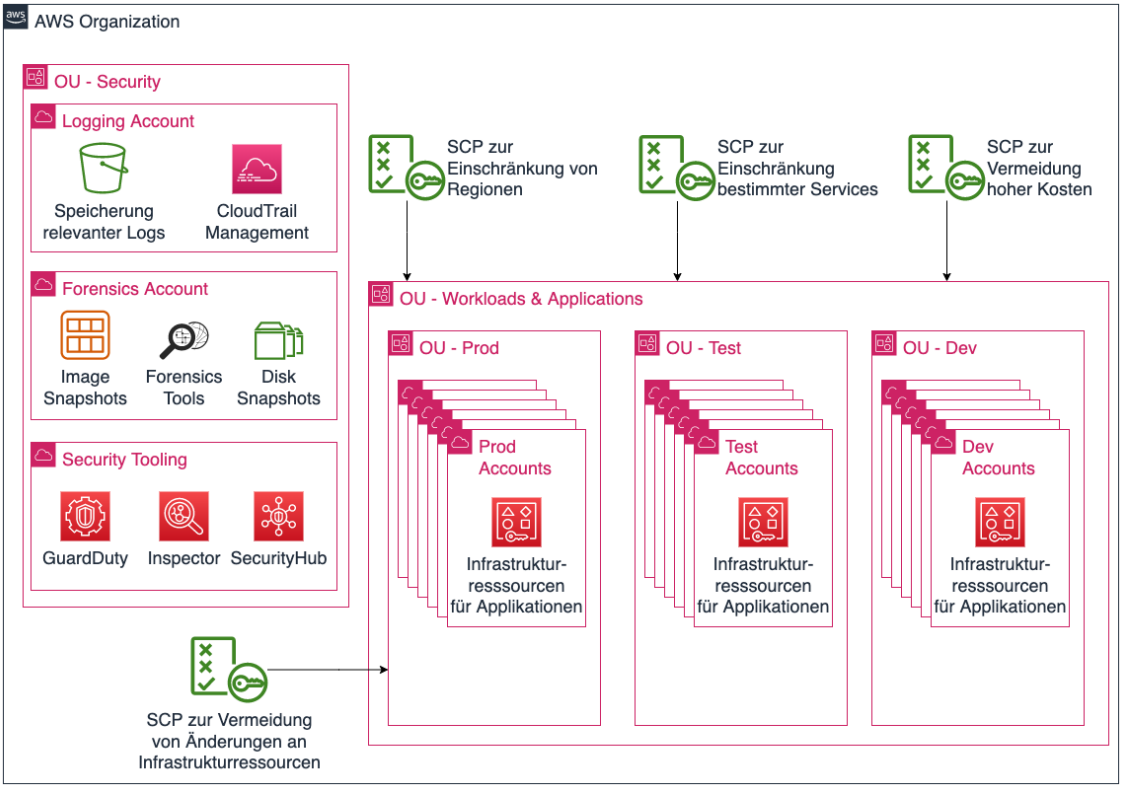


Abbildung 10: Empfehlung für Multi-Account Struktur

6.2 Logs

Logs sind im Rahmen der Incident Response von erheblicher Relevanz. Sie ermöglichen detaillierte Einblicke in Aktivitäten innerhalb eines Netzwerks oder Systems, weshalb sie besonders in der *Detection & Analysis* Phase notwendig sind, um ungewöhnliche oder potenziell schädliche Vorfälle zu erkennen, identifizieren und analysieren. Zudem werden sie für forensische Analysen benötigt, um Beweismittel bereitzustellen und aufgetretene Sicherheitsvorfälle zu rekonstruieren.

Wie bereits in Kapitel 3.1.2 beschrieben, variiert sowohl die generelle Verfügbarkeit als auch der Detaillierungsgrad der Logs im Cloud Umfeld je nach gewähltem

Servicemodell (vgl. Abbildung 6).

6.2.1 Auswahl der notwendigen Logquellen

In der Vorbereitung auf Sicherheitsvorfälle sollten Unternehmen bereits im Vorfeld zunächst die relevanten und notwendigen Log Quellen für die jeweiligen IT-Systeme oder Applikationen, welche in den AWS Accounts betrieben werden, identifizieren.

McAbee u. a. empfehlen Unternehmen sich bei der Auswahl der relevanten Logs die folgenden Fragen zu stellen:

1. Welche Compliance Richtlinien bzw. rechtliche Vorschriften gelten hinsichtlich Logs?
2. Welche AWS Services werden bereits genutzt?
3. Welche AWS Services verarbeiten sensible oder personenbezogene Daten beziehungsweise haben darauf Zugriff?
4. Welche Bedrohungen (*threats*) werden als relevant eingestuft? [40]

   Dazu eignet sich zum Beispiel das sogenannte *Threat modeling (Bedrohungsmodellierung)*, wodurch die Bedrohungen, Risiken und geeignete Minderungsmaßnahmen identifiziert und strukturiert betrachtet werden können [41].

Sobald die relevanten und notwendigen Logquellen identifiziert wurden, ist der nächste Schritt in der Vorbereitung die Einrichtung bzw. Aktivierung dieser Logs. AWS stellt für Kunden das Opensource Projekt *Assisted Log Enabler for AWS* bereit, um Kunden bei der Aktivierung sowie der Zentralisierung verschiedener Logquellen zu unterstützen [42].

Nachfolgend werden die wichtigsten Logquellen in AWS beschrieben, welche je nach den zugrundeliegenden Umständen in Unternehmen aktiviert und eingerichtet werden sollten.

CloudTrail Logs

AWS CloudTrail ist ein Logging Service von AWS der für das Logging von Benutzeraktivitäten und API-Calls innerhalb eines Accounts verantwortlich ist. Aktionen, die von sogenannten *Principals* (Benutzer, Rollen, Services) durchgeführt werden,

werden dabei in sogenannten *CloudTrail Events* erfasst und gespeichert [43, Concepts]. Wie bereits in Kapitel 3.1.2 erwähnt, unterscheidet man hier zwischen *Management Events* und *Data Events*. Die Bedeutung von *Data Events* im Kontext der Incident Response und Analyse von kompromittierten S3 Buckets wird im Detail in Kapitel 8.1 beschrieben.

Anhang A.2 führt beispielhaft ein CloudTrail Event auf. Dabei handelt es sich um ein *Management Event*, welches die Löschung eines S3 Buckets aufzeichnet.

Insgesamt gibt es drei Arten, wie man mit CloudTrail *Events* aufzeichnen kann:

1. **Event history**: Dies ist die Standard Einstellung, die in jedem Account aktiv ist. Sie kann nicht geändert werden und ist so konfiguriert, dass alle *Management Events* aufgezeichnet und für 90 Tage aufbewahrt werden.

2. **Trails**: Ein sogenannter *Trail* ist eine Konfiguration, durch die CloudTrail Events (sowohl *Management Events* als auch *Data Events*) aufgezeichnet und in einem definierten S3 Bucket gespeichert und somit länger als die Standardaufbewahrungsfrist von 90 Tagen aufbewahrt werden können. Optional kann hier auch die Weiterleitung der CloudTrail an andere AWS Services wie zum Beispiel *Amazon CloudWatch* oder *Amazon EventBridge* eingerichtet werden. Die Einrichtung eines *Trails* empfiehlt sich für alle Unternehmen, deren Anforderungen an die Aufbewahrungsfrist größer als 90 Tage ist. Da die CloudTrail Events bei dieser Variante in einem S3 Bucket gespeichert werden, kann dies als Input in bestehende Security Information and Event Management (SIEM)-Systeme (zum Beispiel *Splunk*) genutzt werden.

3. **CloudTrail Lake**: *CloudTrail Lake* ist ein vollständig verwalteter Data Lake zum Erfassen, Speichern, Zugreifen und Analysieren von Logs in AWS. CloudTrail Lake konvertiert die Events von Java Script Object Notation (JSON) in das *Apache ORC* Format, welches für schnelle Datenabfragen optimiert ist. CloudTrail Lake ermöglicht es mittels Structured Query Language (SQL) Abfragen über CloudTrail Events durchzuführen [43, What Is AWS CloudTrail?].

Seit 2018 gibt es die Möglichkeit die Aufzeichnung von CloudTrail Events zentral und vereinfacht über sogenannte *organization trails* für eine gesamte AWS Organization (also mehrere AWS Accounts) zu verwalten [44].

Für Unternehmen, deren Anforderungen eine längere Aufbewahrungsfrist für Logs von 90 Tagen vorsieht, empfiehlt sich definitiv *Management Events* mittels eines *organization trail* aufzuzeichnen.

VPC Flow Logs

Amazon Virtual Private Cloud (VPC) Flow Logs sind ein Feature, durch welches der Netzwerkverkehr innerhalb eines VPC aufgezeichnet werden kann. Flow Logs können für das gesamte Netzwerk, einzelne Subnetze oder einzelne Netzwerkschnittstellen eingerichtet werden. Der Mitschnitt des Netzwerkverkehrs (*flow*) wird in sogenannten *Flow Log Records* aufgezeichnet und gespeichert [45, VPC Flow Logs].

Im Anhang A.1 ist ein Beispiel eines Flow Log Records zu finden. Die enthaltenen Felder eines VPC Flow Log Records sind in der ersten Zeile abgebildet.

VPC Flow Logs sind eine wichtige Quelle, um einerseits detektive Mechanismen auf Netzwerkebene zu etablieren. Andererseits sind sie auch erforderlich, um die Analyse während eines Sicherheitsvorfalls bzw. nach eines Sicherheitsvorfalls, um bei bei der forensischen Auswertung die möglichen Ursachen, das Ausmaß und betroffene Systeme zu identifizieren; zum Beispiel um die Verbreitung von Malware innerhalb eines Netzwerks nachzuvollziehen.

Route 53 Resolver Logs

Amazon Route 53 ist ein Domain Name System (DNS)-Service, der für die Registrierung sowie Verwaltung von Domains genutzt werden kann und als rekursiver DNS Server genutzt werden kann [46, Amazon Route 53 concepts]. Die *53* im Namen ist dabei eine Anspielung auf die von der Internet Assigned Numbers Authority (IANA) zugewiesene Port Nummer für DNS Abfragen.

Die Logs, die von *Amazon Route 53* erfasst werden können, lassen sich in zwei Kategorien unterteilen:

1. **Öffentliche DNS Query Logs**: Die Protokollierung öffentlicher DNS Abfragen für Domains, die in Amazon Route 53 gehostet werden, ermöglicht die Erfassung spezifischer Abfrageinformationen. Dazu gehören der angeforderte Domain- oder Subdomain-Name, der Zeitstempel der Anfrage, der Typ des DNS Datensatzes, der Standort des Route 53-Edge-Servers, der die Anfrage beantwortet hat, sowie der Antwortcode.
2. **Resolver Query Logs**: *Amazon Route 53 Resolver* ist standardmäßig in VPC integriert. Die Erfassung von Resolver Abfrage Logs liefert dieselben Informationen wie öffentliche Abfragen, jedoch mit zusätzlichen Details wie der Instanz-ID der Ressource, von der die Abfrage ausging [40].

AWS empfiehlt den Aufzeichnung von *CloudTrail Events*, *VPC Flow Logs* und *Route 53 Resolver Logs* und bezeichnet diese als „*basic logging trifecta*“ [17, Select and enable log sources].

AWS Config Logs

AWS Config ist ein Service von AWS, der auf die Überwachung und Verwaltung der Konfiguration von AWS Ressourcen abzielt. In seinem Kern ermöglicht er die kontinuierliche Überwachung und Aufzeichnung der Konfigurationen von AWS Ressourcen, um die Einhaltung von Richtlinien und Vorschriften sicherzustellen. Ein Schlüsselkonzept in AWS Config sind die sogenannten *Configuration Items*. Diese stellen eine Momentaufnahme der Eigenschaften, Beziehungen und aktuellen Zustände einer AWS Ressource zu einem bestimmten Zeitpunkt dar. Jedes *Configuration Item* umfasst detaillierte Informationen wie die Erstellungszeit, Beziehungen zu anderen Ressourcen und Änderungshistorie [47, Concepts].

Aus Sicht der Vorbereitung auf die Durchführung bei der Behandlung von Sicherheitsvorfällen (*preparing to handle incidents*) spielt AWS Config eine wichtige Rolle für Unternehmen. So sollten *Configuration Items* als Logquelle aufzeichnet und bestenfalls - wie in Kapitel 6.1.4 beschrieben - in einem zentralen Logging Account gespeichert werden. Diese Aufzeichnungen sind entscheidend, um zu verstehen, was während eines Sicherheitsvorfalls passiert ist. Sie ermöglichen es Incident Response Teams, die Ursache eines Vorfalls genau zu identifizieren und festzustellen, ob eine Konfigurationsänderung zu einem Sicherheitsproblem geführt hat. Damit is es ein wichtiger Service, um die Dynamik und Schnelllebigkeit von Ressourcen in Cloud Umgebungen, die bereits in Kapitel 3.1.5 angesprochen wurden, in Incident Response Prozessen zu berücksichtigen.

AWS Config bietet zudem mittels sogenannter *Config Rules* auch die Möglichkeit, Regeln zu definieren, die automatisch überprüfen, ob die Ressourcenkonfigurationen den festgelegten Anforderungen entsprechen. Dies erleichtert die Einhaltung von Compliance-Standards und verbessert die Sicherheit und Effizienz in der Cloud-Infrastruktur [47, Concepts].

Config Rules sind somit aus Sicht der Verhinderung von Sicherheitsvorfällen (*preventing incidents*) eine wichtige Rolle, weil dadurch Regeln definiert werden können, die automatisch überprüfen, ob die Ressourcenkonfigurationen den Sicherheitsbestimmungen und -richtlinien entsprechen. Dies hilft, potenzielle Sicherheitsrisiken proaktiv zu identifizieren und zu mindern, bevor sie zu echten Sicherheitsvorfällen

werden. So können beispielsweise *Config Rules* erstellt werden, um zu prüfen, dass EC2 Instanzen keinen SSH Zugriff aus dem Internet erlauben.

Weitere servicespezifische Logquellen

Zudem gibt es einer Reihe weiterer servicespezifischer Logs, die für Unternehmen relevant sein könnten. Dazu gehören beispielsweise *Elastic Load Balancing Logs*, *WAF Logs*, *Amazon Elastic Kubernetes Service (Amazon EKS) Audit Logs*, *Lambda Logs* sowie *EC2 Instanz OS Logs*.

Da diese Logquellen meist nur für spezielle Arten von Workloads relevant sind, sei an dieser Stelle an den *AWS Security Incident Response Guide* verwiesen, der im Appendix A eine ausführliche und informative Zusammenfassung enthält [17, Appenix A - Logging and events].

6.2.2 Einrichten von Analyse Mechanismen für Logs

Die NIST empfiehlt als Teil der *Preparation* Phase die Verfügbarkeit von sogenannten *Digital forensic workstations* sicherzustellen, welche die Incident Response Teams bei der Analyse von Log Daten unterstützen [2, S. 22]. Auch wenn die *NIST Special Publication 800-61 Rev. 2* aufgrund der geringen Relevanz des Cloud Computings zur Zeit der Veröffentlichung den Einsatz cloud-nativer Services nicht betrachtet, ist es für Unternehmen unerlässlich im Vorfeld für geeignete Tools zur Analyse von Logs zu sorgen. Dies ist notwendig, um Sicherheitsvorfälle im Detail zu analysieren, die Ursachen zu identifizieren sowie die Auswirkungen innerhalb der Cloud Umgebung zu bestimmen.

In den Abschnitten des Kapitels 6.2.1 wurden bereits eine Reihe möglicher relevanter Loquellen genannt, die bei Unternehmen zu großen Mengen an Logs führen können. Damit diese beim Eintreten eines Sicherheitsvorfalls effektiv und effizient ausgewertet werden können, müssen Unternehmen bereits in der Vorbereitung auf Sicherheitsvorfälle passende Log Analyse Tools ausgewählt und implementiert haben.

Bei der Auswahl von Log Analyse Tools sollten die Personen-, Prozess- und Technologieaspekte innerhalb der Unternehmen berücksichtigt werden. Es sollte auch bedacht werden, dass Tools zur Log Analyse optimal funktionieren, wenn die Anzahl der zu scannenden Logs innerhalb der Grenzen des Tools gehalten werden [17, Select and implement querying mechanisms for logs].

AWS schlägt in *Logging strategies for security incident response* eine Reihe verschiedener Mechanismen zur Analyse von Logs vor, welche in den nachfolgenden Abschnitten beschrieben werden [40].

Amazon Athena

Amazon Athena ist ein interaktiver Query Service, der die direkte Analyse von Daten in S3 mithilfe von Standard-SQL vereinfacht [48, What is Amazon Athena?].

AWS stellt eine Opensource Lösung - das sogenannte *AWS Security Analytics Bootstrap* - zur Einrichtung von *Athena* für Kunden bereit, die Logs mittels cloud-nativer Services analysieren wollen, über keine SIEM Lösung verfügen oder Logs in einem AWS Account analysieren wollen, die nicht in einem zentralen Logging Account (vgl. Abbildung 10) gespeichert sind [49].

AWS bietet den Service in zwei verschieden Preismodellen an. Standardmäßig werden die Abfragen basierend auf der Größe der gescannten Daten in Terabyte abgerechnet. Alternativ gibt es das Preismodell basierend auf sogenannter *Provisioned Capacity*. Dabei wird eine bestimmte Menge an dedizierter Rechenleistung (in Form von Virtual Central Processing Unit (vCPU) und RAM in Gigabyte) vorab bereitgestellt. Die Kosten bei diesem Preismodell beziehen sich auf die *Dauer* der Query und nicht die Größe der zu scannenden Datenmenge [50, Flexible pricing].

Amazon Athena empfiehlt sich für Organisationen, die bereits eine große Menge an Logquellen in S3 gespeichert haben und komplexere Analysen per SQL-Abfragen durchführen wollen und über keine separate SIEM Lösung verfügen.

CloudTrail Event History

Wie bereits in Kapitel 6.2.1 beschrieben, ist diese Variante standardmäßig und kostenlos in jedem AWS Account aktiviert und zeichnet alle *Management Events* für 90 Tage auf. Damit können Queries sowohl in der Weboberfläche als auch per API durchgeführt werden. Allerdings muss hier betont werden, dass die Query Möglichkeiten limitiert sind und nur für simple Abfragen geeignet ist. Außerdem dient dieser Service nur zur Log Analyse von CloudTrail *Management Events* und nicht der anderen Logquellen, welche in Kapitel 3.1.2 aufgezählt wurden.

Aufgrund der eingeschränkten Query Möglichkeiten, der limitierten Datenlage als auch der limitierten Aufbewahrungsfrist von 90 Tagen, sollten sich Unternehmen

nicht auf *CloudTrail Event History* als einzige Möglichkeit zur Analyse von Logs verlassen - auch wenn der Service kostenlos angeboten wird. Stattdessen sollte dies als letzter Ausweg im *worst case* angesehen werden.

CloudTrail Lake

Wie bereits in Kapitel 6.2.1 beschrieben, ist *CloudTrail Lake* ein vollständig verwalteter Data Lake, der eine bestimmte Art definiert, wie Events aufgezeichnet und gespeichert werden können. Da die Daten vor der Speicherung bereits in das *Apache ORC* Format konvertiert werden, sind somit komplexere SQL effizient durchführbar.

Ein *CloudTrail Lake* bietet eine Reihe sogenannter *data stores* an, die als unveränderliche Logquellen importiert und bis zu zehn Jahre aufbewahrt werden können. Dadurch können sowohl AWS-native Logquellen, wie zum Beispiel *Config Logs*, als auch Logs von Drittanbietern, wie zum Beispiel *GitHub*, *CyberArk* oder *CrowdStrike*, sowie benutzerdefinierte Quellen integriert werden [43, Event data stores & Integrations].

Auch wenn AWS den geringen Aufwand für die Verwaltung und den Betrieb des Services als Vorteil betont, ist dieser Aufwand nur für standardmäßige, AWS-native Logquellen gering. Sobald komplexere Analysen und SQL-Abfragen aus verschiedenen Logquellen erwünscht sind, steigt auch der Aufwand für die Konfiguration und Nutzung von *CloudTrail Lake*.

Manuelle Analyse roher Logdaten mittels Kommandozeile

Bei dieser Variante werden rohe Logdaten manuell über die Kommandozeile mittels Terminalbefehle analysiert. Beispiele für Linux Dienstprogramme als Werkzeuge für die Analyse in der Kommandozeile sind *grep, sed, awk, jq, cut, sort* oder *uniq*. Dabei handelt es sich um eine sehr veraltete Methode, die mit erheblichem manuellen Aufwand, weitreichender Expertise und geringer Effizient im Gegensatz zu automatisierten Analysetools verbunden sind.

Deshalb sollte diese Ansatz ebenfalls als letzter Ausweg zur Analyse verwendet werden oder wenn der Einsatz einer der zuvor beschriebenen Services aus Budgetgründen oder aufgrund geltender Unternehmensrichtlinien nicht möglich ist.

Dediziertes SIEM von Drittanbietern

Als letzte Variante sei hier auf den Einsatz einer dedizierten SIEM Lösung von Drittanbietern - wie zum Beispiel *Splunk, Datadog* oder *IBM QRadar* - verwiesen. Diese verfügen über komplexe und effiziente Abfragemethoden, Möglichkeiten für zentrale Visualisierungen in Form von Dashboards sowie Echtzeit Überwachung- und Alarmierungsmöglichkeiten (diese Incident Response Funktion wird in Kapitel 6.2.3 beschrieben.) Dazu muss jedoch der Export der Logdaten der verschiedenen Logquellen aus den AWS Accounts zur SIEM Lösung umgesetzt werden. Je nach Anzahl der Logquellen und Menge an Daten führt dies zu größerem Implementierungsaufwand sowie zusätzlichen Kosten für den Datenverkehr aus AWS heraus (*egress costs*).

Diese Variante empfiehlt sich aufgrund der oben beschriebenen Komplexität und Kosten in der Regel nur für große Unternehmen, mit bestehenden SIEM Lösungen und entsprechender Expertise und Budget für die technische Umsetzung.

6.2.3 Aufbau von Alarmsystemen auf Basis der Logs

AWS bietet eine Reihe cloud-nativer Security Services an, die eine Alarmierungsfunktion (*Alerting*) bereitstellen. Diese Services werden nachfolgenden in Abschnitt 6.3. Solche Alarmsysteme, die auf Basis von gesammelten Logs das Auftreten von Sicherheitsvorfällen erkennen, sind eine wichtige Kompetenz in der ganzheitlichen Durchführung von Incident Response, welche Unternehmen auf- bzw. ausbauen sollten.

Alerting Systeme kommen hauptsächlich in der Phase *Detection & Analysis* zum Einsatz. Dennoch sollten Unternehmen bereits in der Vorbereitung auf Sicherheitsvorfälle geeignete System implementieren, die durch frühzeitige Erkennung dazu beitragen, dass Sicherheitsvorfälle frühzeitig und schnell eingedämmt beziehungsweise verhindert werden können.

Da diese detektiven Maßnahmen je nach Sicherheitsvorfall stark variieren können, werden in den nachfolgenden Kapiteln zu den verschiedenen Incident Response Szenarien typische IoCs aufgezeigt sowie auf charakteristische Events in Logs hingewiesen, auf Basis deren detektive Alerting Maßnahmen aufgebaut werden können.

6.3 Implementierung von Security Services

6.3.1 Amazon GuardDuty

Amazon GuardDuty ist ein Service zur Bedrohungserkennung (*Threat Detection Service*), der eine Reihe von Datenquellen (CloudTrail Management und Data Events, VPC Flow Logs, DNS Logs und S3 Logs) analysiert, um Gefahren und Anomalien innerhalb eines AWS Accounts zu erkennen. Zudem verwendet es Threat-Intelligence-Feeds, wie zum Beispiel *malicious IP* oder *malicious domain lists*, sowie Machine Learning, um ungewöhnliche oder bösartige Aktivitäten zu erkennen. Zum Beispiel, wenn Credentials plötzlich aus ungewöhnlichen geografischen Regionen verwendet werden oder dafür genutzt werden, um ungewöhnliche Aktivitäten innerhalb eines AWS Accounts durchzuführen. GuardDuty ist auch in der Lage kompromittierte EC2 Instanzen zu erkennen, die als Bitcoin Miner oder zur Verbreitung von Malware genutzt werden. Solche Erkenntnisse werden von GuardDuty als sogenannte *Findings* bezeichnet, die eine alarmierende Funktion haben [51, What is Amazon GuardDuty?].

Auch wenn GuardDuty eine Vielzahl der Logquellen als Informationsquelle für die Detektion von Angriffen nutzt, die auch in Kapitel 6.2.1 aufgeführt sind, müssen diese Logquellen von Kunden nicht als Voraussetzung aktiviert werden.

Da es sich bei GuardDuty um einen sogenannten *Threat Detection Service* handelt, wird er vor allem in der *Detection* Phase genutzt. Dennoch empfiehlt es sich für Unternehmen, die IT-Systeme in AWS betreiben, den Service bereits in der Vorbereitung auf die Behandlung von Sicherheitsvorfällen zu aktivieren, um in der Lage zu sein, Vorfälle frühzeitig zu erkennen und auf diese schnell zu reagieren.

6.3.2 Amazon Inspector

Amazon Inspector ist ein *Vulnerability Management* Service, der Software Schwachstellen als auch Risiken durch Netzwerk Schwachstellen durch automatisierte Scans von EC2 Instanzen, Lambda Funktionen sowie Container Images in Container Registries identifizieren kann.*Amazon Inspector* verfügt über eine Integration zu *AWS Organizations* wodurch die Verwaltung von automatisierten Scans in Multi-Account Landschaften stark vereinfacht wird [52, What is Amazon Inspector?].

Sobald eine Schwachstelle in einer Ressource identifiziert wurde, wird dadurch ein sogenanntes *Finding* generiert. Die folgenden drei Arten von Findings werden von *Amazon Inspector* generiert:

1. **Package vulnerability**: Diese Findings beziehen sich auf Schwachstellen in verwendeten Softwarepaketen, welche von Common Vulnerabilities and Exposures (CVE) betroffen sind. Das CVE-System ist eine Referenzmethode für öffentlich bekannte Schwachstellen und Gefährdungen der Informationssicherheit. Angreifer können diese ungepatchten Schwachstellen ausnutzen, um die Vertraulichkeit, Integrität oder Verfügbarkeit von Daten zu gefährden oder auf andere Systeme zuzugreifen.

2. **Code vulnerability**: Anders als bei den oben genannten Package Vulnerabilities wurden hier Schwachstellen im Code gefunden, der von Cloud-Nutzern selbst geschrieben wurde. Dabei wird der Quellcode auf Schwachstellen geprüft, die von Angreifern ausgenutzt werden könnten. Dazu gehören beispielsweise fehlende Validierung und Sanitisierung von User Input, schwache kryptografische Methoden oder Code, der für SQL Injections anfällig ist.

3. **Network vulnerability**: Diese Findings beziehen sich auf die Netzwerkerreichbarkeit und weisen auf offene Netzwerkpfade hin, über welche EC2 Instanzen aus dem Internet erreichbar sind. Dadurch werden potentielle Netzwerkschwachstellen durch fehl konfigurierte sowie übermäßig offene Firewallregeln identifiziert [52, Finding types in Amazon Inspector].

Amazon Inspector ist ein weiterer essentieller Service, der Unternehmen bei der Prävention von Sicherheitsvorfällen unterstützt. Durch regelmäßige Scans Unternehmen Sicherheitsrisiken kontinuierlich identifizieren und beheben. Der Service generiert detaillierte Berichte über die entdeckten Schwachstellen, inklusive Empfehlungen zur Behebung. Diese proaktive Herangehensweise ermöglicht es Unternehmen, die Sicherheit ihrer AWS Umgebungen zu stärken und die Wahrscheinlichkeit des Eintretens von Sicherheitsvorfällen signifikant zu reduzieren.

Auch wenn am Markt eine Vielzahl an *Vulnerability Management* Tools verfügbar sind, eignet sich *Amazon Inspector* besonders für Unternehmen mit erheblichen Infrastrukturressourcen, die über eine Multi-Account Landschaft verteilt sind. Dies liegt an der einfachen, zentralen Verwaltung als auch der Tatsache, dass der für die Scans notwendige *Inspector Agent* bereits bei den gängigsten EC2 Images vorinstalliert ist. Dadurch wird der Konfigurations- und Wartungsaufwand für die Bereitstellung für Unternehmen stark reduziert.

6.3.3 AWS Security Hub

AWS Security Hub ist ein Service, der sowohl in der *Preparation* Phase des Incident Response Phasenmodells für die Prävention von Sicherheitsvorfällen als auch in der*Detection & Analysis* Phase zum Einsatz kommt.

AWS Security Hub ist darauf ausgelegt, die Überwachung von Compliance- und Sicherheitsanforderungen über mehrere AWS Dienste hinweg zu konsolidieren und zu vereinfachen, indem es *Findings* aus verschiedenen AWS Services - wie zum Beispiel *Amazon GuardDuty* oder *Amazon Inspector* - sowie auch aus Tools von Drittanbietern aggregiert, organisiert und priorisiert [53, What is AWS Security Hub?].

Durch diese Integration bietet Security Hub eine zentrale, vollumfängliche Sicht auf die Sicherheitslage innerhalb der AWS-Umgebung, was besonders während der Behandlung von Sicherheitsvorfällen Vorteile für Unternehmen bietet.

Die Hauptfunktion von *AWS Security Hub* besteht darin, die Einhaltung von Sicherheitsstandards und die Konfiguration von Services kontinuierlich zu überwachen und zu bewerten. Diese Überwachung umfasst die Identifizierung und Behebung von Schwachstellen, die Überwachung der Netzwerksicherheit sowie die Verwaltung von Zugriffsrechten. Dies geschieht durch die Überprüfung von Regeln in Form sogenannter *Security Controls*, die sich auf die Konfiguration einer bestimmten Ressource beziehen. So gibt es Beispiel eine bestimmte *Security Control*, die prüft, ob ein S3 Bucket serverseitig verschlüsselt ist. Mehrer *Security Controls* können wiederum in einem *Security Standard* zusammengefasst werden, welche den Fokus auf bestimmte Compliance Frameworks - wie zum Beispiel die *NIST SP 800-53* oder der *Center for Internet Security (CIS) AWS Foundations Benchmark* - legen und die Einhaltung dieser Frameworks prüfen [53, Security controls and standards in AWS Security Hub].

Dies Funktionalität ist für Unternehmen besonders im Kontext der Prävention von Sicherheitsvorfällen eine große Hilfe, indem es proaktive Erkenntnisse und Empfehlungen zur Verbesserung der Sicherheitsmaßnahmen liefert. Dies ermöglicht es Organisationen, potenzielle Bedrohungen und Schwachstellen frühzeitig zu erkennen und entsprechende Maßnahmen zu ergreifen, um Sicherheitsvorfälle zu verhindern, bevor sie entstehen. Die Kombination aus automatisierten Sicherheitschecks und der Möglichkeit, individuelle Alarme und Aktionen zu konfigurieren, macht AWS Security Hub zu einem wertvollen Instrument für die Incident Response in Unternehmen.

6.4 Aufbau eines Asset Inventory

6.4.1 Motivation

Der Aufbau eines stringenten *Asset Inventory* ist ein wichtiger Teil der *Preparation* Phase des Incident Response Modells. Ein Asset Inventory unterstützt die Behandlung von Sicherheitsvorfällen, indem Incident Response Teams darüber wichtige Metadaten von betroffenen Ressourcen abfragen können. Dies ist zum Beispiel wichtig, um schnell identifizieren zu können, ob es sich bei kompromittierten Daten eines S3 Buckets um sensible Daten - wie zum Beispiel Gesundheitsdaten von Kunden - oder um Daten mit geringerer Datenschutzklassifizierung - wie zum Beispiel öffentlich Daten - handelt. Zudem können in einem Asset Inventory die relevanten Applikationsverantwortlichen oder Ansprechpartner für den Betrieb hinterlegt werden, die bei einem Sicherheitsvorfall informiert werden müssen.

6.4.2 AWS Config

Der Service *AWS Config* wurde bereits in Kapitel 6.2.1 als relevante Logquelle - speziell durch die Informationen aus Logs der *Config Items* - beschrieben. Durch die kontinuierliche Aufzeichnung und Auswertung der Konfigurationen von AWS-Ressourcen ermöglicht AWS Config eine genaue und dynamische Bestandsaufnahme aller Ressourcen innerhalb eines AWS Accounts und ist somit ideal für den Aufbau eines Asset Inventory.

Durch die automatische und kontinuierliche Bestandsaufnahme in Form der *Configuration Items* können Schatten-IT Systeme aufgedeckt werden. Darüber hinaus bietet AWS Config integrierte Dashboards und Reports, die einen Überblick über die aktuelle Konfiguration und den Status der Assets bieten. Diese Dashboards können angepasst werden, um spezifische Informationen hervorzuheben, die für das jeweilige Unternehmen relevant sind.

Mittels einheitlicher Tagging Konvention

Ein Asset Inventory kann durch die Anreicherung weiterer Informationen in Form von Tags erweitert werden. Tags können genutzt werden, um Informationen aus verschiedenen Dimensionen bereitzustellen.

- **Technical Tags**: Diese Tags können technische Informationen wie zum Beispiel *Application ID*, *Environment (Dev, Test, Prod)* oder *Version* enthalten.
- **Business Tags**: Diese Tags können geschäftsrelevante Informationen bereitstellen. Zum Beispiel *Projekt IDs*, *Applikationsverantwortliche*, *Kostenstelle* oder die Zugehörigkeit der Information zu einem bestimmten Kunden in Form einer *Kunden ID*.
- **Security Tags**: Darüber können Ressourcen beispielsweise hinsichtlich ihrer *Datenschutzklasse (öffentlich, intern, vertraulich, geheim)* oder der *Geschäftskritikalität (unkritisch, kritisch, hochkritisch)* kategorisiert werden

Wenn Unternehmen in der Vorbereitung auf die Behandlung von Sicherheitsvorfällen diese wichtigen Informationen in einem Asset Inventory für Incident Response Teams bereitstellen, dann können diese Teams schneller die relevanten Stakeholder involvieren, sowie die weiteren Schritte in Abhängigkeit der Geschäftskritikalität priorisieren und einstufen [54, Tagging categories].

Tags sind für fast alle AWS Ressourcen verfügbar. Damit die gesetzten Tags jedoch ihren Informationsgehalt entfalten können, empfiehlt es sich eine einheitliche und konsequente Tagging Konvention aufzubauen. Dadurch wird ein stringenter Aufbau von Tags garantiert, der spätere Automatisierungs- und Abfragemechanismen ermöglicht.

Für Unternehmen empfiehlt es sich ihre Tagging Konvention durch sogenannte *Tag Policies* durchzusetzen. Dabei handelt es sich um Richtlinien, welche das Setzen von Tags ressourcenübergreifend über alle AWS Accounts in einer *AWS Organization* durch bestimmte Regeln standardisieren [38, Tag Policies].

6.5 Implementierung von Backup Strategien

6.5.1 Bedeutung von Backups im Kontext von Incident Response

Backups und effektive Backupstrategien sind ein kritischer Bestandteil der betrieblichen Resilienz und spielen eine entscheidende Rolle in der *Recovery*-Phase des Incident Response Phasenmodells. In diesem Kontext dienen Backups als wesentliche Sicherheitsnetze, die es Unternehmen ermöglichen, nach einem Datenverlustereignis - in dieser Betrachtung konkret als Folge eines Sicherheitsvoralls - ihre Geschäftskontinuität aufrechtzuerhalten. Eine sorgfältig geplante Backupstrategie sollte die regelmäßige Sicherung wichtiger Daten und Systemkonfigurationen umfassen, wobei

sowohl die Frequenz als auch der Umfang der Backups auf die spezifischen Bedürfnisse und Risikoprofile des Unternehmens abgestimmt sein müssen.

In der *Recovery*-Phase des Incident Response Phasenmodells ermöglichen Backups eine schnellere Wiederherstellung von Daten und Systemen, wodurch die Ausfallzeiten minimiert und die Auswirkungen auf das Geschäft reduziert werden. Ebenso wichtig ist die regelmäßige Überprüfung und das Testen der Backup- und Wiederherstellungsverfahren, um sicherzustellen, dass sie im Falle eines echten Incidents effektiv funktionieren. Insgesamt sind Backups ein unverzichtbares Element im Incident Response Plan eines Unternehmens, das nicht nur zur Wiederherstellung nach einem Vorfall beiträgt, sondern auch die Gesamtresilienz gegenüber verschiedenen Bedrohungen stärkt.

Damit Unternehmen in der *Recovery* Phase auf existierende Backups zurückgreifen können, erfordert dies, dass die Konzipierung und Umsetzung von Backup Strategien bereits zuvor in der *Preparation* Phase durchgeführt wurde.

6.5.2 Umsetzung von Backup Strategien in AWS Umgebungen

AWS bietet für die meisten Datenspeicher- als auch Datenbankservices native, sofort einsatzfähige Backupfunktionalitäten an. Als Beispiel können hier die Features *DynamoDB Point-in-time recovery* oder *Amazon RDS Snapshots* aufgeführt werden, wodurch Cloud-Nutzern das automatisierte und kontinuierliche Erstellen von Datensicherungen als auch die Wiederherstellung auf Basis dieser enorm vereinfacht wird.

Für große Unternehmen, die eine Vielzahl von IT-Systemen und Anwendungen über mehrere AWS Accounts betreiben, empfiehlt sich der Einsatz von *AWS Backup*. Dabei handelt es sich um einen vollständig verwalteten Service, der die Zentralisierung und Automatisierung der Datensicherung serviceübergreifend und über mehrere Accounts hinweg vereinfacht. Cloud-Nutzer können sogenannte *Backup Plans* definieren, welche die Frequenz sowie das Zeitfenster der Backup-Erstellung festlegen, um somit Unternehmensrichtlinien zentral zu verwalten und über die gesamte AWS Organization auszurollen. Zudem können Backups in sogenannten *Backup Vaults* an einer zentralen Stelle - zum Beispiel in einem dedizierten *Backup Account*, auf den nur ein ausgewählter Personenkreis Zugriff hat - gespeichert werden [55, What is AWS Backup?].

Dies kann im schlimmsten Falle nach einem eingetretenem Sicherheitsvorall bei der Wiederherstellung von Daten oder als Reaktion auf ein Ransomware Angriff, bei dem Daten in einem bestimmten System oder Account verschlüsselt wurden, helfen.

Abbildung 11 zeigt eine beispielhafte Implementierung einer organisationsweiten Backupstrategie, bei der ein zentral verwalteter *Backup Plan* in einem dedizierten Backup Account existiert. Dieser definiert, dass täglich inkrementelle Backups im Zeitraum von 01:00 - 04:00 Uhr erstellt und in einem zentralen *Backup Vault* abgelegt werden. Diese Datensicherungsrichtlinie gilt für alle verwendeten Datenbank- und Speicherservices in den AWS Accounts des Unternehmens.

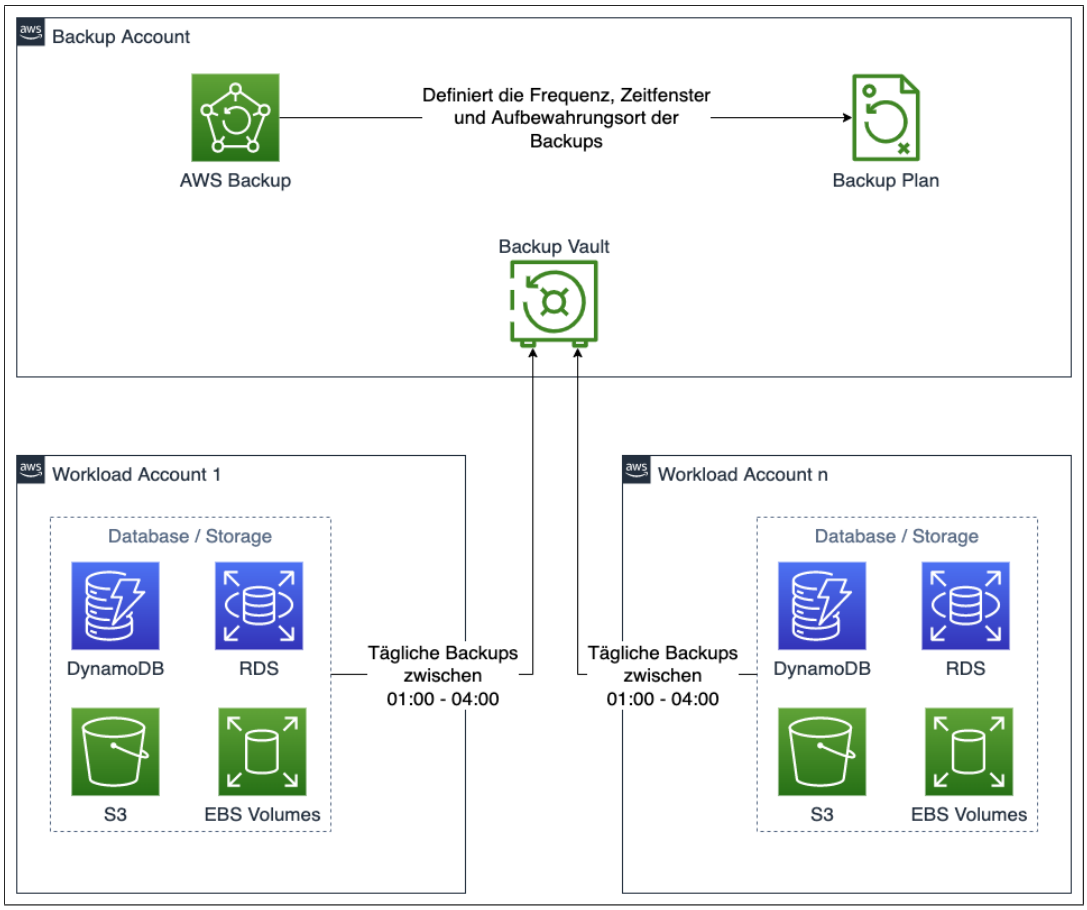


Abbildung 11: Implementierung einer Datensicherungsstrategie mittels AWS Backup

7 Szenario I: Compromised Credentials

In diesem Kapitel werden relevante AWS Services vorgestellt, die während der Behandlung von Sicherheitsvorfällen eingesetzt werden können, welche auf kompromittierte Anmeldeinformationen basieren.

Auch wenn bei der Konzeption des Szenarios in Kapitel 4.1 erwähnt wurde, dass einer der häufigsten Ursachen der Kompromittierung von Credentials das Einbinden in öffentlichen git Repositories darstellt, wird in diesem Szenario dies nicht als einzige Ursache betrachtet. Weitere denkbare Ursachen könnte eine Kompromittierung als Resultat eines Phishing Angriffs, durch Spyware oder auch Social Engineering sein.

Für die durchzuführenden Aktivitäten als Teil des Incident Response Prozesses spielt es zunächst keine Rolle *wie* Credentials kompromittiert wurden, sondern lediglich *dass* Credentials kompromittiert wurden.

7.1 Begriffserklärung

Für die weitere Ausführung sind zunächst einige Begriffserklärungen notwendig. Generell unterscheidet man bei AWS zwischen *long-lived (langlebigen)* und *short-lived (kurzlebigen)* Credentials.

Long-lived Credentials Langlebige Credentials, sind Anmeldeinformationen für sogenannte *IAM Users* in AWS. Sie sind eine Kombination aus der `Access Key ID` und dem `Secret Access Key`. Wie der Name bereits impliziert, haben diese Credentials kein zeitlich festgelegten Ablauf der Gültigkeit. Cloud-Nutzer haben allerdings die Möglichkeit die Credentials eines *IAM Users* zu deaktivieren, sowie zu reaktivieren oder komplett zu löschen. In CloudTrail Events werden langlebige Credentials im Feld `accessKeyId` durch den Präfix `AKIA` spezifiziert (vgl. Anhang A.2 Zeile 8).

Im Kontext der Incident Response, ist es wichtig, die folgenden Puntke zu beachten:

- Ein Access Key kann nach der Deaktivierung wieder reaktiviert werden.
- Ein Access Key kann nicht wiederhergestellt werden, nachdem er gelöscht wurde.
- Ein IAM User kann jeweils maximal zwei Access Keys haben.
- Sobald ein Access Key deaktiviert oder gelöscht wird, kann diese nicht mehr für die Durchführung von API Aufrufen in AWS verwendet werden [17, Technique and access containment].

Short-lived Credentials Kurzlebige Credentials, sind Anmeldeinformationen für sogenannte *IAM Roles* in AWS. Sie bestehen wie langlebige Credentials aus einer Kombination aus der `Access Key ID`, dem `Secret Access Key` sowie einem zusätzlichen `Session Token`. Sie sind dadurch gekennzeichnet, dass sie eine zeitlich begrenzte Gültigkeitsdauer haben, welche in der *Session Duration* festgelegt wird und zwischen 15 Minuten und 36 Stunden liegt. In CloudTrail Events werden kurzlebige Credentials im Feld `accessKeyId` durch den Präfix `ASIA` spezifiziert.

7.2 Preparation - Preventing Incidents

Das Ziel der *Preparation-* Phase ist es, Sicherheitsvorfälle vorbeugend zu verhindern.

AWS eine Reihe von *Security best practices in IAM*, welche die Kompromittierung von Anmeldeinformationen verhindern können. Diese werden in den nachfolgenden Abschnitten erläutert.

7.2.1 git-secrets

git-secrets ist ein von AWS bereitgestelltes Opensource Projekt, mit Hilfe dessen sowohl einzelne Commits als auch die gesamte git Commit History gescannt werden kann, um zu verhindern, dass diese sensible Daten wie zum Beispiel AWS Access Keys enthalten [57].

7.2.2 Vermeiden Technische User

Unter einem *Technischen User* versteht man einen IAM User, dessen Access Keys von Applikationen, Servern oder Programmen genutzt werden, um mit den AWS APIs zu kommunizieren. Unwissen, Zeitdruck sowie menschliche Fehler können dazu führen, dass diese Access Keys auf unsichere Weise gespeichert werden - zum Beispiel als Klartext in Umgebungsvariablen - was das Risiko der Kompromittierung von Anmeldeinformationen birgt.

Deshalb empfiehlt AWS Technische User grundsätzlich zu vermeiden und stattdessen kurzlebige Credentials von IAM Rollen zu verwenden [56].

So können Applikationen, welche auf EC2 Instanzen laufen, temporäre Credentials durch die Verknüpfung der Instanz mit einer IAM Rolle erhalten. Für Applikationen, welche nicht auf AWS Infratsukturressourcen ausgeführt werden aber dennoch mit AWS APIs interagieren müssen, empfiehlt sich der Einsatz von *IAM Roles Anywhere*. Dabei wird jede Applikation mit signierten Client-Zertifikaten ausgestattet, mit deren Hilfe vordefinierte IAM Rollen angenommen werden können.

7.2.3 Federation für menschliche Nutzer

Generell empfiehlt AWS jedoch auch, dass langlebige Credentials durch IAM User für *menschliche Nutzer* vermieden werden. Stattdessen wird in diesem Fall empfohlen auf die sogenannte *Federation* mit Hilfe des Services *AWS IAM Identity Center* zu setzen [56].

Federation bezeichnet einen Prozess, in dem eine externe Identitätsquelle zur Authentifizierung und Autorisierung von Benutzern für den Zugriff auf AWS Ressourcen genutzt wird. Diese externe Identitätsquelle kann ein Unternehmensverzeichnis wie Active Directory oder ein Identity Provider (IdP) sein, der das Security Assertion Markup Language (SAML) Protokoll unterstützt. Durch die Implementierung einer Federation können Unternehmen ihre bestehenden Authentifizierungssysteme nutzen, um den Zugriff auf AWS-Dienste zu steuern, ohne separate AWS-Benutzerkonten erstellen zu müssen. Dies ermöglicht eine zentrale Verwaltung der Benutzeridentitäten und Zugriffsrechte, was die Sicherheit erhöht und die Verwaltung vereinfacht.

7.2.4 MFA aktiveren

Falls die oben genannte Empfehlung der Verwendung von Federation durch einen IdP für menschliche Nutzer aus bestimmten Gründen in Unternehmen nicht umgesetzt werden kann, ist es unerlässlich, dass zusätzlich zu den long-lived Credentials eine Absicherung durch MFA erfolgt. Dies ist eine weitere Schutzmaßnahme, die im Falle kompromittierter Credentials die Nutzung dieser verhindern kann [56].

7.2.5 Credentials regelmäßig rotieren

Sowohl im Falle von long-lived Credentials für *Technische User* als auch für die Verwendung durch menschliche Nutzer, sollten die verwendeten Access Keys regelmäßig rotiert werden. Schweizer empfiehlt, dass diese Rotation der Credentials alle 90 Tage durchgeführt werden sollte [58].

AWS Config (vgl. Kapitel 6.2.1) bietet eine *Config Rule* an, welche automatisch prüft, ob Access Keys innerhalb eines definierbaren Zeitraums rotiert wurden. Sobald durch diese Config Rule ein Access Keys identifiziert wird, der diesen Zeitraum überschreiten, wird ein `NON_COMPLIANT` Finding erzeugt [47, access-keys-rotated].

7.3 Detection

Das Ziel der *Detection*- Phase ist es, potentielle sicherheitsrelevante Ereignisse (vgl. auch Definition aus Kapitel 2.1.1) zu identifizieren.

7.3.1 Alarmquellen

Alarme über kompromittierte Credentials bzw. Sicherheitsvorfälle im allgemeinen können aus verschiedenen Quellen abstammen:

1. **Detektion durch AWS**: In diesem Fall werden Cloud-Nutzer direkt von AWS benachrichtigt, dass Credentials öffentlich freigelegt wurden. Dies geschieht durch Benachrichtigungen per Mail an die E-Mail Adresse, die bei der Erstellung des AWS Accounts verwendet wurde, sowie an die E-Mail Adresse, die (optional) als Security Kontakt des Accounts konfiguriert wurde. Zusätzlich wird ein Support Ticket in dem jeweiligen AWS Account erstellt. Im Anhang B befindet sich ein Auszug aus einer Standardbenachrichtigung, die von AWS

versendet wurde, nachdem ein Access Key (in diesem Fall bewusst) in einem *GitHub* Repository veröffentlicht wurde.

2. **Detektion durch externe Parteien**: In diesem Fall werden Cloud-Nutzer von externen Parteien - wie zum Beispiel Security Forschende - über die Kompromittierung informiert.

3. **Detektion durch interne Mechanismen**: Anders als in den zuvor genannten Fällen, erfolgt die Detektion hier durch Mechanismen, die durch von den Cloud-Nutzern konzipiert und ausgerollt wurden. Dazu gehört der Threat Detection Service *GuardDuty* sowie benutzerdefinierte Detektionsmaßnahmen.

4. **Nachgelagerte Erkennung der Kompromittierung**: Im schlimmsten Fall werden kompromittierte Credentials nur indirekt durch Serviceausfälle von Applikationen, deren Infrastrukturressourcen sich in den kompromittierten AWS Accounts befinden, oder durch ungewöhnlich hohe Kosten in der monatlichen Abrechnung erkannt. Dies deutet auf einen geringen Reifegrad der detektiven Fähigkeiten eines Unternehmens hin.

Da Organisationen den größten Einfluss auf den Erfolg der internen Detektionsmechanismen haben, werden diese nachfolgend betrachtet und erläutert. Diese detektiven Mechanismen lassen sich grundsätzlich in zwei Kategorien unterteilen:

1. **Verhaltensbasierte Erkennung**: Diese Erkennung beruht auf mathematischen Modellen, die üblicherweise als maschinelles Lernen oder künstliche Intelligenz bezeichnet werden. Die Erkennung erfolgt *inferentiell*, sodass der Alarm nicht zwangsläufig ein tatsächliches Ereignis widerspiegelt.

2. **Regelbasierte Erkennung**: Diese Form der Erkennung ist deterministisch; Cloud-Nutzer können genau festlegen, über welche Aktivitäten sie informiert werden wollen, was eine gewisse Sicherheit bietet [17, Detective control implementations].

Moderene Threat Detection Systems - darunter auch *GuardDuty* - verfügen in der Regel über beide Formen der Erkennung.

7.3.2 Erkennung mittels GuardDuty

Wie zuvor in Kapitel 6.3.1 beschrieben handelt es sich bei GuardDuty um einen sogenannten Threat Detection Service, der als solcher in der Lage ist Sicherheitsvorfälle zu erkennen. Erkannte Threats werden durch GuardDuty in einem sogenann-

ten Finding dargestellt. In Bezug auf kompromittierte Credentials sind besonders die Findings vom Typ *IAM* relevante IoCs, die zur Detektion genutzt werden können.

Diese Findings enthalten immer den Resource Type `Access Key`. Sie beruhen auf Anomalieerknunng - zum Beispiel, wenn die Credentials aus ungewöhnlichen geografischen Regionen verwendet werden oder Aktionen durchgeführt werden, die nicht in das gewöhnliche Verhaltensprofil passen - sowie sogenannter *Threat Intelligence Feeds*, wie zum Beispiel bekannte bösartige IP-Adressen [51, GuardDuty IAM finding types].

Im Anhang A.3 ist ein beispielhaftes Finding vom Typ `UnauthorizedAccess: IAMUser/MaliciousIPCaller` zu finden. Dabei handelt es sich um eine *regelbasierte Erkennung*, da ein deterministischer Ansatz in Form von sogenannten *IP reputation lists* angewandt wird. Aus dem `Title` in Zeile 124 geht hervor, dass die Access Keys für die Ausführung bestimmter APIs von einer bekannten, bösartigen IP-Adresse genutzt wurden. GuardDuty aggregiert in diesem Finding bereits mehrere Events zu einem Finding, was sich beispielsweise unter `AdditionalInfo` in den Zeilen 118 bis 120 erkennen lässt: dort befinden sich weitere Informationen wie die Anzahl aller beobachteter API Aufrufe sowie der erste und letzte Zeitpunkt, in dem diese auftraten.

Auch wenn GuardDuty bereits eine Vielzahl an Information in einem Finding aggregiert, empfiehlt sich dennoch eine weitere Analyse der betroffenen Access Keys. Dazu eignen sich jedoch die Informationen, welche bereits in einem GuardDuty Finding enthalten sind. Im Falle kompromittierter Credentials sind besonders die Werte der `AccessKeyDetails` (im Beispiel in Zeile 11 bis 16) von großer Relevanz. So lassen sich daraus direkt die `AccessKeyId` sowie die `PrincipalId` ableiten, welche bei der weiteren Analyse als auch bei den Maßnahmen in der *Containment, Eradication & Recovery* Phase benötigt werden.

7.3.3 Implementierung benutzerdefinierter Detektionsmaßnahmen

Neben sofort-einsatzfähigen AWS Services - wie zum Beispiel GuardDuty - für die Detektion von Sicherheitsvorfällen, gibt es auch die Möglichkeit benutzerdefinierte Detektionsmechanismen auf Basis verschiedener Logquellen (vgl. Kapitel 3.1.2) in Kombination mit weiteren AWS Services zu implementieren.

Ein gängiger Ansatz ist es, regelbasierte Detektionsmechanismen basierend auf bestimmten CloudTrail Events aufzubauen. Dabei werden bestimmte API Aufrufe als

Indicators of Compromise (IoC) eingestuft. Durch eine *EventBridge Rule*, wird in Reaktion auf diese API Aufrufe ein Amazon Simple Notification Service (SNS) Topic getriggert, wodurch ein definierter Empfängerkreis beispielsweise per E-Mail benachrichtigt wird.

Nachfolgend ist in Abbildung 12 eine Referenzarchitektur für diesen Ansatz dargestellt. In dem abgebildeten Beispiel wurden die CloudTrail Events für das Stoppen, Neustarten und die Terminierung von EC2 Instanzen als kritische Events eingestuft, welche erkannt werden und zu einer Benachrichtigung des Incident Response Teams führen sollen.

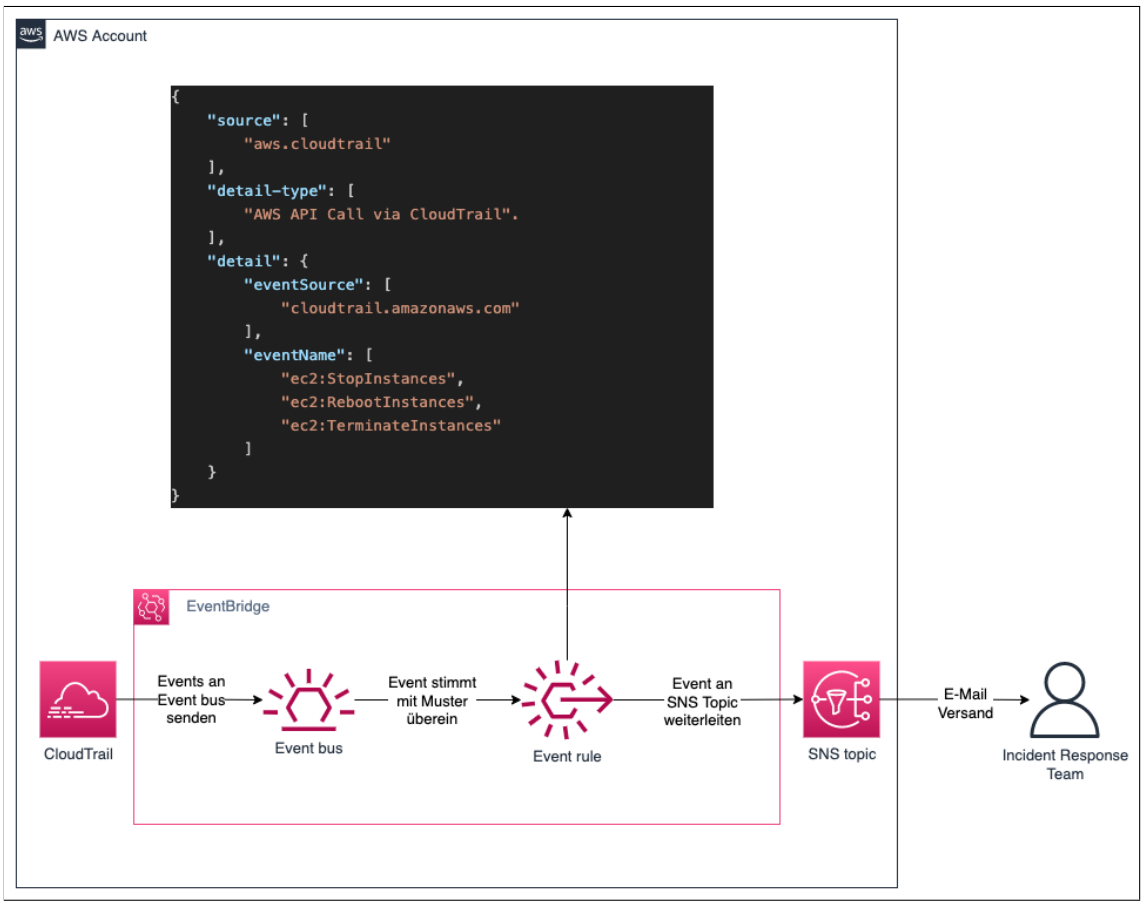


Abbildung 12: Detektive Maßnahme auf Basis bestimmter CloudTrail Events

Die CloudTrail Events, die als IoC eingestuft werden und worüber ein Incident Response Team informiert werden soll, können von Organisation zu Organisation stark variieren. Es empfiehlt sich diese in Abhängigkeit eines *Threat Modelling* zu wählen. Hier sei auch erwähnt, dass es schwierig ist, bestimmte API Aufrufe kategorisch als Sicherheitsvorfälle einzustufen, was wiederum zu *false-positives* (Fehlalarmen)

führen kann. So lässt sich nicht pauschal sagen, ob das Stoppen einer EC2 Instanz durch einen Angreifer mit dem Ziel der Serviceunterbrechung oder durch Routinewartungsarbeiten eines Operations Team erfolgte.

An dieser Stelle sei auch an *Sensitive IAM Actions* verwiesen, wo Farris eine Sammlung kritischer CloudTrail Events, die auf kompromittierte Credentials hinweisen, veröffentlichte [59].

7.4 Analysis

Während dieser Phase wird eine umfassende Log-Analyse durchgeführt, mit dem Ziel, Alarme zu validieren, den Umfang eines Sicherheitsvorfalls zu identifizieren und die Auswirkungen einer möglichen Kompromittierung zu bewerten.

- Die **Validierung** des Alarms ist der Ausgangspunkt der Analysephase. Incident Response Teams suchen nach Log-Einträgen aus verschiedenen Quellen und nehmen Kontakt zu den relevanten Applikationsverantwortlichen auf.
- Das sogenannte **Scoping** ist der nächste Schritt, bei dem alle beteiligten Ressourcen inventarisiert werden und die Kritikalität des Alarms nach Zustimmung der Stakeholder angepasst wird, sobald ein Fehlalarm ausgeschlossen wurde. Dadurch wird der Rahmen für die folgenden Schritte identifiziert.
- Abschließend beschreibt die **Auswirkungsanalyse** das Identifizieren des Ausmaß der Sicherheitsvorfalls sowie tatsächlicher geschäftlicher Unterbrechungen [17, Validate, scope, and assess impact of alert].

7.4.1 Vorgehensweise

Die folgenden Fragestellungen eignen sich für die Vorgehensweise bei der Analyse kompromittierter Credentials.

1. Zu welchem Zeitpunkt wurde der kompromittierte `Access Key` zum ersten Mal verwendet?
2. Welche Aktionen/API Aufrufe wurden mit den kompromittierten Credentials durchgeführt?
3. Waren diese Aktionen erfolgreich?
4. Wurden neue Ressourcen erstellt bzw. existierende modifiziert?

5. Wurden neue IAM Identitäten erstellt bzw. existierende modifiziert?
6. Von welcher IP Adresse wurden die kompromittierten Credentials genutzt?
7. Wurde diese IP Adresse von weiteren Identitäten genutzt?

Farris hat die Vorgehensweise bei der Analyse inklusive der relevanten Fragestellungen in einem Flussdiagramm zusammengefasst, welches in Abbildung 13 dargestellt ist.

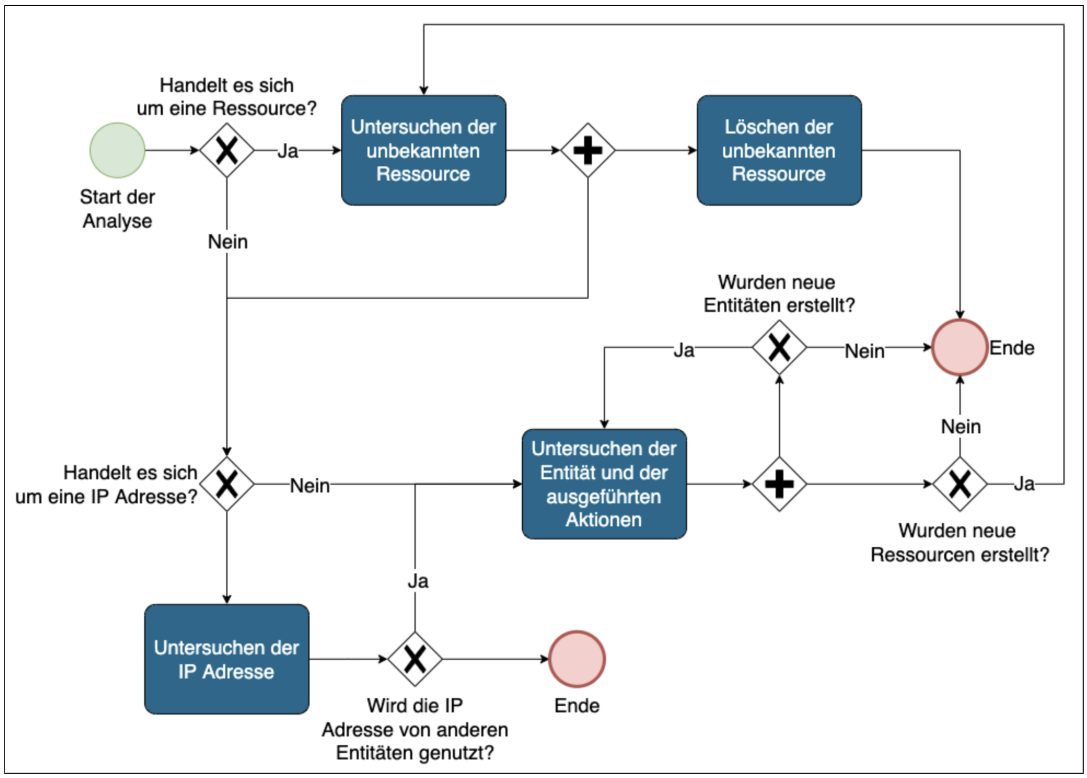


Abbildung 13: Flussdiagramm für Analyse [eigene Darstellung in Anlehnung an 60, CloudTrail Investigation]

7.4.2 Analyse mittels Amazon Athena

Damit Organisationen überhaupt in der Lage sind weiterführende Analysen durchführen zu können, muss eine Möglichkeit der Analyse von Logquellen implementiert sein. Dazu gibt es verschiedene Möglichkeiten, welche bereits in Kapitel 6.2.2 erläutert wurden. In diesem Abschnitt wird dabei der cloud-native Service *Amazon Athena* näher betrachtet und entsprechende Queries geliefert, welche zur Beantwortung der oben aufgeführten Fragestellungen dienen.

Im Kontext kompromittierter Credentials, ist der wichtigste Input für die Durchführung weiterer Analysen, die `Access Key ID`, welche kompromittiert wurde. Wie im vorherigen Abschnitt beschrieben, kann diese bereits direkt aus einem GuardDuty Finding ausgelesen (vgl. Anhang A.3 Zeile 12) bzw. aus CloudTrail Events abgeleitet werden (vgl. Anhang A.2 Zeile 8).

Nachfolgende sind beispielhafte Queries aufgeführt, die zur Beantwortung der Fragestellungen aus Kapitel 7.4.1 herangezogen werden können.

Quelltext 7.1: Athena Query: Zu welchem Zeitpunkt wurde der komrpomittierte Access Key zum ersten Mal verwendet?

```
1 SELECT eventtime, * FROM 'table-name'
2 WHERE useridentity.accesskeyid = '<access_key_id>'
```

Quelltext 7.2: Athena Query: Welche Aktionen/API Aufrufe wurden mit den kompromittierten Credentials durchgeführt?

```
1 SELECT eventname, * FROM 'table-name'
2 WHERE useridentity.accesskeyid = '<access_key_id>'
```

Quelltext 7.3: Athena Query: Waren diese Aktionen erfolgreich?

```
1 SELECT eventname, errorcode, errormessage, * FROM 'table-name'
2 WHERE useridentity.accesskeyid = '<access_key_id>'
```

Quelltext 7.4: Athena Query: Von welcher IP Adresse wurden die kompromittierten Credentials genutzt?

```
1 SELECT sourceipaddress, * FROM 'table-name'
2 WHERE useridentity.accesskeyid = '<access_key_id>'
```

Quelltext 7.5: Athena Query: Wurde diese IP Adresse von weiteren Identitäten genutzt?

```
1 SELECT useridentity, * FROM 'table-name'
2 WHERE sourceipaddress = '<sourceip_of_previous_query>'
```

7.5 Containment

Das Ziel der *Containment*-Phase ist es, die Auswirkungen eines Sicherheitsvorfalls zu begrenzen bzw. zu minimieren.

Im Bezug auf kompromittierte Credentials, zielt diese Phase darauf ab, dass die Credentials blockiert werden, um weitere Ausbreitungen und Schaden durch Angreifer einzuschränken. Falls die Anayse ergab, dass ein kompromittierter Access Key genutzt wurde, um weitere Access Keys zu erstellen, so müssen diese neu erstellten Access Keys ebenfalls eingedämmt werden.

Die *Containment*-Phase bei kompromittierten Credential besteht im wesentlichen aus dem *Einschränken von Berechtigungen* und dem *Widerrufen der Access Keys*.

7.5.1 Einschränkung von Berechtigungen

Diese Maßnahmen zielen darauf ab, die effektiven Berechtigungen der kompromittierten Credentials einzuschränken. Dadurch wird zum Beispiel verhindert, dass die Credentials genutzt werden können, um EC2 Instanzen für Cryptomining zu starten oder geheime Daten aus S3 Buckets zu lesen.

Je nachdem wie die Kompromittierung von Credentials erkannt wurde, können diese Einschränkungen bereits durch AWS ohne Mitwirken des Cloud-Nutzers umgesetzt worden sein bzw. müssen durch die betroffenen Cloud-Nutzer selbst durchgeführt werden.

Isolierung durch AWS

Falls die Kompromittierung durch AWS erkannt wurde, werden die betroffenen Access Keys automatisch mit einer Quarantäne Policy (`AWSCompromisedKey QuarantineV2 Permissions`) belegt (vgl. Benachrichtigungsmail von AWS in Anhang B).

Diese Quarantäne Policy schränkt jedoch nicht alle Berechtigungen des Access Keys ein, sondern lediglich bestimmte Aktionen, die von AWS als *„hoch-riskante Aktionen“* eingestuft werden, um gleichzeitig Serviceunterbrechungen von Cloud-Nutzern zu minimieren. Diese hoch-riskanten Aktionen lassen sich in die nachfolgenden Kategorien unterteilen.

1. **Verhindern von Privilege Escalation**: Es sind eine Vielzahl an `iam` Aktionen blockiert, die einem Angreifer die Möglichkeiten der Privilege Escalation - also das Erweitern der bestehenden Berechtigungen - nimmt.
2. **Verhindern von Datenlöschung in S3 Buckets**: Es werden bestimmte `s3` Aktionen blockiert, die für das Löschen von Objekten benötigt werden. Zudem wird auch verhindert, dass Buckets öffentlich zugänglich - also zu sogenannten *Public Buckets* - gemacht werden können. Dennoch können die Credentials auch mit der Quarantäne verwendet werden, um Daten aus S3 Buckets zu lesen, da diese Aktion (`s3:GetObject`) nicht blockiert wird.

3. **Verhindern von Cryptominern**:Es wird verhindert, dass Cryptominer gestartet werden können, indem eine Reihe von Rechenservices - wie zum Beispiel *EC2*, *Lambda* oder *LightSail* - blockiert werden.
4. **Verhindern von Services mit Vorauszahlung**: Es werden bestimmte Aktionen blockiert, die für Services mit hohen Vorauszahlungen benötigt werden, wie zum Beispiel der Kauf von *Saving Plans*. Dadurch werden die finanzielle Auswirkungen von Cloud-Nutzern abgeschwächt.

Eine detaillierte Aufführung aller Einschränkung ist aus der vollständigen Quarantäne Policy in Anhang C zu entnehmen.

Hier muss betont werden, dass die Isolierung durch AWS nur einen Teil der möglichen Berechtigungen einschränkt und ein weiteres Handeln durch benutzerdefinierte Isolierung erforderlich ist.

Benutzerdefinierte Isolierung

Zusätzlich zur automatisch angehängten Quarantäne Policy empfiehlt AWS auch ein weitreichenderes Einschränken der Berechtigungen mittels benutzerdefinierter Isolierung [17, Technique and access containment].

Die drastischste Isolierung erfolgt über eine sogenannte *Deny-All Policy*, bei der alle Berechtigungen eines Access Keys explizit entzogen werden. Der Quelltext 7.6 stellt eine solche Policy dar.

Quelltext 7.6: Deny-All Policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": "*",
            "Effect": "Deny",
            "Resource": "*"
        }
    ]
}
```

Allerdings müssen sich Incident Response Teams darüber bewusst sein, dass dadurch auch Applikationen eingeschränkt werden, welche den Access Key verwenden.

Deshalb sollten in dieser Phase die relevanten Stakeholder und Entscheidungsträger involviert sein.

Eine abgeschwächte Version dieser Policy kann eine Ausnahmeregelung für bestimmte IP Adressbereiche enthalten, die als vertrauensvoll eingestuft werden. Dazu wird eine Bedingung in die Policy integriert, welche die IP Adresse untersucht, von der die Access Keys benutzt werden. Der Quelltext 7.7 stellt eine solche Policy dar.

Quelltext 7.7: Deny-All Policy mit Ausnahme für bestimmte IP Adressen

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "*",
            "Resource": "*",
            "Condition": {
                "IpAddress": {
                    "aws:SourceIp": "<WHITELISTED_IP_ADDRESS>"
                }
            }
        }
    ]
}
```

7.5.2 Widerrufen von Access Keys

Zusätzlich zur Eindämmung der kompromittierten Credentials durch das Entziehen von Berechtigungen empfiehlt AWS die Access Keys zu widerrufen. [17, Technique and access containment]. Dabei unterscheidet sich das Vorgehen je nachdem, ob es sich um *long-lived* oder *short-lived* Credentials handelt (vgl. Kapitel 7.1).

Long-lived Credentials

Wie bereits in Kapitel 7.1 beschrieben, kann der Zugriff auf einen AWS Account mittels Access Keys beschränkt werden, indem der Access Key deaktiviert oder gelöscht wird.

Für die Widerrufung langlebiger Credentials empfiehlt es sich, den jeweiligen Access Key zunächst lediglich zu deaktivieren. Da durch das Widerrufen des Access Keys Systeme betroffen sein können, die unter keinen Umständen unterbrochen werden dürfen, ergibt sich dadurch die Möglichkeit den Access Key zu reaktivieren. Auch hier ist es wieder wichtig, dass für solche Entscheidungen das Incident Response Team die relevanten Stakeholder und Entscheidungsträger involviert.

Die Deaktivierung eines bestimmten Access Keys kann vom Incident Response Team in der Webkonsole, per API oder durch folgenden AWS Command Line Interface (CLI) Befehl durchgeführt werden.

Quelltext 7.8: CLI Befehl zum Deaktivieren eines Access Keys

```
1 aws iam update-access-key \
2     --access-key-id AKIAI... \
3     --status Inactive
```

Short-lived Credentials

Short-lived Credentials werden typischerweise für IAM Rollen ausgestellt. Wie bereits in Kapitel 7.1 beschrieben und wie der Name impliziert, sind sie nur für einen begrenzten Zeitraum (zwischen 15 Minuten und 36 Stunden) gültig. Deshalb besteht bei kompromittierten short-lived Credentials für Angreifer ein kleineres Zeitfenster, um diese auszunutzen. Das ist auch der Grund, warum es sich aus einer Risikobetrachtung heraus empfiehlt möglichst mit short-lived Credentials zu arbeiten.

Sollte die Kompromittierung von short-lived Credentials jedoch erkannt werden, empfiehlt AWS dennoch diese ebenfalls zu widerrufen [17, Technique and access containment].

Anders als bei langlebigen Access Keys können diese kurzlebigen Credentials nicht direkt gelöscht oder deaktiviert werden. Das Widerrufen von short-lived Credentials geschieht über das Anhängen einer Policy an die betroffene IAM Rolle. Der Aufbau ist ähnlich zur *Deny-All* Policy (vgl. Quelltext 7.6), wobei der zusätzliche Conidition Key `aws:TokenIssueTime` integriert ist, welcher prüft, zu welchem Zeitpunkt die kurzlebigen Credentials der jeweiligen IAM Rolle ausgestellt wurden. Somit lassen sich alle short-lived Credentials widerrufen, die vor einem bestimmten Zeitpunkt ausgestellt wurden [61, Revoking IAM role temporary security credentials].

Im Quelltext 7.9 ist eine solche Policy beispielhaft abgebildet. In diesem Fall würden alle Credentials, welche vor dem 10. Dezember 2023 ausgestellt wurden, effektiv nicht

mehr nutzbar sein, da alle Aktionen geblockt werden.

Quelltext 7.9: Deny-All Policy für Credentials, die vor einem bestimmten Zeitpunkt ausgestellt wurden.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Deny",
        "Action": "*",
        "Resource": "*",
        "Condition": {
            "DateLessThan": {
                "aws:TokenIssueTime": "2023-12-10T00:00:00Z"
            }
        }
    }
}
```

Das Widerrufen der short-lived Credentials durch das Anhängen der oben genannten Policy hat zur Folge, dass auch legitime Applikationen ihren Zugriff auf AWS Ressourcen verlieren, falls sie vor einem bestimmten Zeitpunkt ausgestellt wurden. Sie können jedoch nach erfolgreicher Authentifizierung neue short-lived Credentials beantragen.

7.6 Eradication

Das Ziel der *Eradication*-Phase ist es, nicht autorisierte Ressourcen oder Artefakte, die im Zusammenhang mit Sicherheitsvorfällen stehen, zu entfernen.

Die Aktivitäten in dieser Phase sind abhängig von den durchgeführten Aktionen, die mit Hilfe der kompromittierten Credentials ausgeführt wurden. Deshalb ist es wichtig, bereits in der Analyse Phase zu prüfen, wie die Credentials genutzt wurden (vgl. Fragestellungen aus Kapitel 7.4.1).

Ein Angreifer könnte beispielsweise eine Reihe von EC2 Instanzen für Cryptomining provisioniert oder zusätzliche Access Keys erstellt haben, um weitere Einstiegspunkte in den kompromittierten AWS Account zu schaffen. Diese Ressourcen müssten dann von einem Incident Response Team entfernt werden.

Hierbei ist es jedoch wichtig zu beachten, dass regulatorische Anforderungen an die Erstellung und Aufbewahrung forensischer Artefakte erfüllt werden müssen. Falls

kompromittierte Access Keys zum Beispiel genutzt wurden, um EC2 Instanzen zu modifizieren, empfiehlt es sich vor dem Löschen der Ressourcen Snapshots zu erstellen und diese in einem dedizierten Account für forensische Untersuchen aufzubewahren [17, Eradication].

7.7 Recovery

Das Ziel der *Recovery*-Phase ist es, die von Sicherheitsvorfällen betroffene System wieder in ihren ursprünglichen, sicheren Zustand und Regelbetrieb zu versetzen. Ähnlich wie bei der *Eradication*-Phase sind die durchzuführenden Maßnahmen abhängig von dem eingetretenen Schaden bzw. den durchgeführten Aktionen, die in der Analyse des Sicherheitsvorfalls identifiziert wurden.

Das NIST führt in *NIST Special Publication 800-61 Rev. 2* eine Reihe von allgemeinen Maßnahmen für die *Recovery*-Phase auf: die Wiederherstellung von Systemen auf Basis von Backups, den Neuaufbau von Systemen von Grund auf, die Installation von Patches, das Ändern von Passwörtern und die Verschärfung der Netzwerkperimetersicherheit [2, S. 35].

Diese Maßnahmen lassen sich in Bezug auf AWS-Umgebungen konkretisieren, indem beispielsweise Systeme durch Datenstände aus einem *Backup Vault* (vgl. Kapitel 6.5.2) wiederhergestellt werden oder ein eingetretener Sicherheitsvorfall als Anlass genommen wird, um alle Access Keys zu rotieren. Zudem können weitere vorbeugende Maßnahmen, welche in Kapitel 7.2 erläutert wurden, implementiert werden, um das Sicherheitsniveau in AWS Umgebungen zu erhöhen.

8 Szenario II: Compromised S3 Buckets

In diesem Kapitel werden relevante AWS Services vorgestellt, die während der Behandlung von Sicherheitsvorfällen eingesetzt werden können, welche auf kompromittierten S3 Buckets basieren.

8.1 Preparation - Preparing to handle Incidents

Bei diesem Szenario handelt es sich um einen Sonderfall, bei dem es zusätzlich notwendige Maßnahmen gibt, die während der *Preparation*-Phase getroffen werden müssen, um überhaupt in der Lage zu sein eintretende Sicherheitsvorfälle zu behandeln. Um kompromittierte S3 Buckets effektiv und effizient zu behandeln, sind weitere Maßnahmen zu den in Kapitel 6 aufgeführten Maßnahmen umzusetzen. Diese beziehen sich konkret auf das Aktivieren von Logs, welche standardmäßig nicht vorhanden aber für die Behandlung kompromittierter S3 Buckets zwingend notwendig sind.

Das liegt daran, dass es sich bei den üblichen Folgen von kompromittierten S3 Buckets - nämlich Daten exfiltrieren, löschen oder modifizieren - um sogenannte *data plane operations* handelt, welche standardmäßig nicht geloggt werden. Diese Operationen sind in Abbildung 8 dargestellt.

8.1.1 Notwendigkeit für das Logging von data plane operations

Die Notwendigkeit für das Logging von Data Events ergibt aus rechtlichen Anforderungen an Unternehmen. Als Beispiel sei hier die Datenschutz-Grundverordnung (DSGVO) aufgeführt, welche Organisationen dazu verpflichtet Sicherheitsvorfälle in Bezug auf personenbezogene Daten an die Aufsichtsbehörden als auch die betroffenen Personen zu melden. Diese Pflicht ergibt sich aus den Artikeln 33 und 34. In Artikel 33 sind die Inhalte festgeschrieben, welche dabei dokumentiert und gemeldet werden müssen. Diese umfassen alle „im Zusammenhang mit der Verletzung des Schutzes personenbezogener Daten stehenden Fakten“ [62, Art. 33 (5)].

Ohne das Logging von *data plane operations* sind Organisationen nicht in der Lage die folgenden Fragestellungen zu beantworten:

- Welche Daten wurden aus einem S3 Bucket exilftriert, modifiziert oder gelöscht?
- Durch wen (Access Key und/oder IP-Adresse) wurden die Daten aus einem S3 Bucket exilftriert, modifiziert oder gelöscht?
- In welchem Zeitraum wurden die Daten aus einem S3 Bucket exilftriert, modifiziert oder gelöscht?

Für S3 Buckets gibt es zwei Optionen, um *data plane operations* zu loggen: *CloudTrail Data Events* und *S3 Server Access Logs*. Diese Optionen werden in den nachfolgenden Abschnitten näher untersucht und bewertet.

8.1.2 CloudTrail Data Events

Wie in Kapitel 6.2.1 beschrieben, werden standardmäßig lediglich *control plane operations (Management Events)* erfasst, weshalb *data plane operations (Data Events)* explizit aktiviert werden müssen. Im Anhang A.4 ist ein beispielhaftes CloudTrail Data Event aufgeführt.

Vorteile

Das Erfassen von *data plane operations* mittels CloudTrail Data Events bietet die folgenden Vorteile gegenüber S3 Server Access Logs:

- **Skalierbarkeit**: CloudTrail Data Events lassen sich sehr einfach skalieren, indem ein sogenannter *Organizational Trail* (vgl. Kapitel 6.2.1) erstellt wird. Dadurch werden automatisch alle *data plane operations* von allen S3 Buckets in allen AWS Accounts einer Organization erfasst.
- **Zentrale Aufbewahrung der Logs**: CloudTrail Data Events aus verschiedenen AWS Accounts können einfach in einem dedizierten Logging Account gespeichert werden, indem dies als Ziel beim Erstellen eines *Organizational Trail* angegeben wird.

Nachteile

Der größte Nachteil der Erfassung von *data plane operations* durch CloudTrail Data Events liegt in den **zusätzlichen Kosten**. Zusätzlich zu den Kosten für die reine Speicherung der Data Events, fallen Kosten für die Generierung dieser Data Events an. Im Dezember 2023 fallen hierfür $0,10 pro 100.000 Data Events an [63, Paid Tier]. Gerade für Organisationen mit einer Vielzahl an S3 Buckets und hochfrequenten Abrufmustern können dadurch signifikante Zusatzkosten entstehen.

8.1.3 S3 Server Access Logs

Die zweite Variante zum Erfassen von *data plane operations* sind S3 Server Access Logs. Dabei handelt es sich um das detaillierte Aufzeichnen von Requests, die gegen S3 Buckets durchgeführt werden [64, Logging requests with server access logging]. Im Anhang A.5 ist ein beispielhaftes S3 Server Access Log aufgeführt.

Vorteile

Das Erfassen von *data plane operations* mittels S3 Server Access Logs bietet die folgenden Vorteile gegenüber CloudTrail Data Events:

- **Kosten**: Das Protokollieren der *data plane operations* verursacht keine zusätzlichen Kosten. Es entstehen lediglich Kosten für die Speicherung der S3 Server Access Logs.
- **Höherer Detaillierungsgrad der Logs**: Im Vergleich zu den Logs in Form von CloudTrail Data Events werden bei S3 Server Access Logs noch zusätzliche Felder aufgezeichnet: *Bucket Owner, Object Size* und *HTTP Referer*.

Nachteile

S3 Server Access Logs beinhalten jedoch eine Reihe von Nachteilen gegenüber CloudTrail Data Events:

- **Aufwändige Skalierung**: Die Skalierung der Implementierung von S3 Server Access Logs gestaltet sich deutlich aufwendiger als CloudTrail Data Events. Das liegt vorallem an einer fehlenden Integration des Services zu *AWS Organization*, weshalb es mit erheblichem Mehraufwand verbunden ist, S3 Server

Access Logging für alle Buckets in allen AWS Accounts einer Organisation auszurollen.

- **Zentralisierung nicht möglich**: S3 Server Access Logs können lediglich dezentral gespeichert werden: Die Server Access Logs müssen in einem weiteren S3 Bucket gespeichert werden, der in der gleichen Region und dem gleichen AWS Account sein muss. Somit ist es nicht möglich, alle Server Access Logs in einem dedizierten zentralen Logging Account zu speichern.
- **Best Effort Logging**: Die Bereitstellung der S3 Server Access Logs erfolgt auf einer *„Best Effort Basis“*. Das bedeutet, dass bestimmte *data plane operations* möglicherweise mit erheblichem Zeitversatz, in mehrfacher Ausführung oder überhaupt nicht übermittelt werden [64, Logging requests with server access logging].

8.2 Preparation - Preventing Incidents

Wie bereits in Kapitel 4.2.2 beschrieben, können die Ursachen für kompromittierte S3 Buckets fälschliche Konfigurationen der Buckets oder kompromittierte Credentials sein. In diesem Abschnitt werden Mechanismen zum Vermeiden kompromittierter S3 Buckets aufgrund von Miskonfigurationen beschrieben. Für Maßnahmen zur Vermeidung kompromittierter Credentials sei auf Kapitel 7.2 verwiesen.

Miskonfigurationen von S3 Buckets führen dazu, dass S3 Buckets öffentlich zugänglich werden, wodurch das Risiko von kompromittierten Daten entsteht. Es gibt eine Reihe verschiedener AWS Services, welche in der Lage sind, solche Miskonfigurationen zu erkennen.

8.2.1 AWS Config Rules

In Kapitel 6.2.1 wurden *AWS Config Rules* bereits erläutert. Diese können genutzt werden, um zu prüfen, ob bestimmte Miskonfigurationen dazu führen, dass Daten aus S3 Buckets gelesen bzw. in die S3 Buckets geschrieben werden können. AWS stellt dafür die folgenden Config Rules bereit:

- `s3-bucket-public-read-prohibited`: Diese Regel prüft, ob S3 Buckets öffentlichen Lesezugriff erlauben [47, s3-bucket-public-read-prohibited].

- `s3-bucket-public-write-prohibited`: Diese Regel prüft, ob S3 Buckets öffentlichen Schreibzugriff erlauben [47, s3-bucket-public-write- prohibited].

8.2.2 AWS Security Hub Controls

In Kapitel 6.3.3 wurde *AWS Secrurity Hub* bereits vorgestellt. Es gibt eine Reihe von Security Hub Controls, welche genutzt werden können, um Miskonfigurationen von S3 Buckets zu identifizieren:

- `[S3.1] S3 Block Public Access setting should be enabled`: Dieser Control prüft, ob die die Blockierung des öffentlichen Zugriffs auf der Account-Ebene konfiguriert ist.
- `[S3.2] S3 buckets should prohibit public read access`: In diesem Control wird geprüft, ob S3 Buckets öffentlichen Lesezugriff zulassen.
- `[S3.3] S3 buckets should prohibit public write access`: In diesem Control wird geprüft, ob S3 Buckets öffentlichen Schreibzugriff zulassen [53, Amazon Simple Storage Service controls].

8.2.3 IAM Access Analyzer

Ein weiterer möglicher AWS Service, um Miskonfigurationen in S3 Buckets zu identifizieren, ist *IAM Access Analyzer*. IAM Access Analyzer ermöglicht es, Ressourcen innerhalb einer AWS Organization zu identifizieren, die mit einer externen Entität geteilt werden. Dadurch können Sicherheitsrisiken durch unbeabsichtigte Zugriffe auf Ressourcen und Daten erkannt werden [61, Identifying resources shared with an external entityt].

So erzeugt IAM Access Analyzer ein Finding, wenn S3 Buckets mit externen Entitäten geteilt sind. Ein solches Finding ist beispielhaft im Anhang A.6 dargestellt.

8.2.4 Amazon Macie

Amazon Macie ist ein weiterer Service, der für die Identifikation von fehlkonfigurierten S3 Buckets genutzt werden kann, um somit die Kompromittierung von S3 Buckets vorzubeugen.

Amazon Macie ist ein Datensicherheitsdienst, der sensible Daten mithilfe von maschinellem Lernen und Mustervergleich erkennt, Einblick in Datensicherheitsrisiken bietet und einen automatisierten Schutz vor diesen Risiken ermöglicht [65, What is Macie?].

Die folgenden Finding Typen weisen auf fehlkonfigurierte S3 Buckets hin:

- `Policy:IAMUser/S3BlockPublicAccessDisabled`
- `Policy:IAMUser/S3BucketPublic`
- `Policy:IAMUser/S3BucketReplicatedExternally`
- `Policy:IAMUser/S3BucketSharedExternally` [65, Types of Amazon Macie findings].

8.3 Detection

In diesem Abschnitt werden AWS Services näher betrachtet, welche für die Erkennung von S3 Bucket Kompromittierung eingesetzt werden können. Dabei werden Detektionsmechanismen durch *Amazon GuardDuty* sowie mögliche Implementierungen von benutzerdefinierten Maßnahmen betrachtet.

8.3.1 Erkennung mittels GuardDuty

Wie bereits in Kapitel 6.3.1 beschrieben, verwendet *Amazon GuardDuty* im Hintergrund *CloudTrail Data Events* - auch wenn diese von den Cloud-Nutzern nicht explizit aktiviert wurden. Auf Basis dieser Logquellen ist GuardDuty in der Lage eine Reihe von Findings zu generieren, die Cloud-Nutzer auf mögliche S3 Bucket Kompromittierungen hinweisen. Diese sind nachfolgend aufgeführt.

- `Exfiltration:S3/AnomalousBehavior`: Dieses Finding weist darauf hin, dass eine IAM-Entität API Aufrufe durchführt, welche von der etablierten Baseline dieser Entität abweichen. Der verwendete API Aufruf ist mit der Exfiltrationsphase eines Angriffs verbunden, in der ein Angreifer versucht, Daten zu *sammeln*. Diese Aktivität ist verdächtig, da die IAM-Entität die API auf ungewöhnliche Weise aufgerufen hat. Dies tritt beispielsweise auf, wenn eine IAM Entität zum ersten Mal eine S3 API aufruft oder diese von einem ungewöhnlichen Standort durchführt [51, Exfiltration:S3/AnomalousBehavior]. Hierbei handelt es sich um eine *verhaltensbasierte Erkennung*.

- `Exfiltration:S3/MaliciousIPCaller`: Dieses Finding weist darauf hin, dass eine S3 API von einer IP-Adresse aufgerufen wurde, die mit bekanntermaßen böswilliger Aktivität in Zusammenhang steht. Die beobachtete API wird häufig mit Exfiltrationstaktiken in Verbindung gebracht, bei denen ein Angreifer versucht, Daten aus einem Netzwerk zu sammeln. Beispiele hierfür sind die Aktionen `GetObject` und `CopyObject` [51, Exfiltration:S3/Malicious-IPCaller]. Hierbei handelt es sich um eine *regelbasierte Erkennung.*
- `Impact:S3/AnomalousBehavior.Delete`: Dieses Finding weist darauf hin, dass eine IAM-Entität API Aufrufe durchführt, welche von der etablierten Baseline dieser Entität abweichen. Der verwendete API Aufruf ist mit einem Angriff verbunden, der versucht, Daten zu *löschen.* Diese Aktivität ist verdächtig, da die IAM-Entität die API auf ungewöhnliche Weise aufgerufen hat. Dies tritt beispielsweise auf, wenn eine IAM Entität zum ersten Mal eine S3 API aufruft oder diese von einem ungewöhnlichen Standort durchführt [51, Impact:S3/AnomalousBehavior.Delete]. Hierbei handelt es sich um eine *verhaltensbasierte Erkennung.*
- `Impact:S3/AnomalousBehavior.Write`: Dieses Finding weist darauf hin, dass eine IAM-Entität API Aufrufe durchführt, welche von der etablierten Baseline dieser Entität abweichen. Der verwendete API Aufruf ist mit einem Angriff verbunden, der versucht, Daten zu *manipulieren.* Diese Aktivität ist verdächtig, da die IAM-Entität die API auf ungewöhnliche Weise aufgerufen hat. Dies tritt beispielsweise auf, wenn eine IAM Entität zum ersten Mal eine S3 API aufruft oder diese von einem ungewöhnlichen Standort durchführt [51, Impact:S3/AnomalousBehavior.Write]. Hierbei handelt es sich um eine *verhaltensbasierte Erkennung.*

8.3.2 Implementierung benutzerdefinierter Detekionsmaßnahmen

Die Implementierung benutzerdefinierter Maßnahmen zur Detektion der Kompromittierung von S3 Buckets ist möglich aber mit deutlich höherem Aufwand und Komplexität verbunden. Das liegt daran, dass mögliche IoCs für dieses Szenario deutlich schwerer durch Aktivitäten aus Logquellen abzubilden sind. In der Regel handelt es sich beim lesenden Zugriff auf Objekte in S3 Buckets um hochfrequente Zugriffe. Außerdem ist es nicht trivial legitime Zugriffe von potentiellen bösartigen Zugriffen durch heuristische Ansätze in Form von regelbasierte Erkennung zu imple-

mentieren. Dies führt dazu, dass benutzerdefinierte Detekionsmaßnahmen auf Basis von regelbasierter Erkennung meist eine hohe Quote an *false-positives* aufweisen.

Der Aufbau von verhaltensbasierten Erkennungsmaßnahmen erfordert Expertise im Bereich maschinellem Lernen und lässt sich nicht für mehrere Organisationen pauschalisieren. Deshalb wird dies hier nicht weiter betrachtet.

Allerdings besteht die Möglichkeit ein Alarmmechanismus für das Löschen von Daten in S3 Buckets zu implementieren. Das ist dann sinnvoll, wenn es sich dabei um statische Daten handelt und davon auszugehen ist, dass diese Daten im Regelfall nicht auf legitime Weise gelöscht werden. Ein solcher Alarmmechanismus basiert auf *CloudTrail Data Events*, welche von einer *EventBridge Rule* erfasst werden, die wiederum das Incident Response Team per E-Mail auf Basis eines *SNS Topics* alarmiert. Die beschriebene Referenzarchitektur ist analog zu Abbildung 12. Dabei muss lediglich der `eventName` zu `s3:DeleteObject` angepasst werden, damit das Löschen eines Objekts einen Alarm auslöst.

Zudem besteht für einen Angreifer die Möglichkeit mittels einer sogenannten *Lifecycle Configuration*, Daten aus einem S3 Bucket indirekt und zeitversetzt löschen zu lassen. Eine Lifecycle Configuration besteht aus Regeln und Aktionen, welche nach einer bestimmten Zeit eintreffen. Damit können beispielsweise Objekte in S3 Buckets nach dem Ablauf einer definierten Zeit automatisiert zur Löschung veranlasst werden [64, Managing your storage lifecycle]. Deshalb sollten die API Ausführungen der Aktion `s3:PutBucketLifecycleConfiguration`, wie im vorherigen Absatz beschrieben, ebenfalls überwacht werden.

8.4 Analysis

Anders als bei der Analyse von kompromittierten Credentials (vgl. Kapitel 7.4) erfordert die weitere Analyse, dass Logs zu den *data plane operations* in Form von CloudTrail Data Events oder S3 Server Access Logs vorliegen. Ohne diese Logs ist es nicht möglich einen Sicherheitsvorfall zu validieren oder die Auswirkungen einer eingetretenen Kompromittierung zu bestimmen.

8.4.1 Vorgehensweise

Die folgenden Fragestellungen eignen sich als Leitfaden für die Analyse im Falle von kompromittierten S3 Buckets.

1. Welche IAM Entität hat API Aufrufe gegen den verdächtigten S3 Bucket durchgeführt?
2. Von welcher IP Adresse und mit welchem User Agent wurden diese API Aufrufe durchgeführt?
3. Wurden Objekte aus einem S3 Bucket exilftriert?
4. Wurden Objekte aus einem S3 Bucket gelöscht?
5. Wurden neue Objekte einem S3 Bucket hinzugefügt?

Der wichtigste Input für die weitere Analyse ist die ARN des Buckets, bei dem eine Kompromittierung vermutet wird. Im Fall der Detektion durch GuardDuty, lässt sich diese aus dem Feld `Resource` des jeweiligen Findings ablesen.

8.4.2 Analyse mittels Amazon Athena

In diesem Abschnitt werden möglicher Queries für Amazon Athena vorgestellt, mit der Antworten auf die zuvor genannten Fragestellungen für die Analyse kompromittierter S3 Buckets gefunden werden können. Dabei werden mögliche Queries sowohl auf Basis von CloudTrail Data Events als auch von S3 Server Access Logs vorgestellt.

Queries auf Basis von CloudTrail Data Events

Die Athena Queries im Anhang D.1 können verwendet werden, wenn CloudTrail Data Events als Logquelle vorhanden sind. Dabei müssen die Parameter `<bucket-name>`, `<table-name>`, `<start-date>` und `<end-date>` mit dem jeweiligen Bucket Namen, dem Namen der Datenbanktabelle sowie dem zu untersuchenden Zeitraum ausgetauscht werden.

Queries auf Basis von S3 Server Access Logs

Die Athena Queries im Anhang D.2 können verwendet werden, wenn S3 Server Access Logs als Logquelle vorhanden sind. Dabei müssen die Parameter `<bucket-name>`, `<start-date>` und `<end-date>` mit dem jeweiligen Bucket Namen, sowie dem zu untersuchenden Zeitraum ausgetauscht werden.

8.5 Containment

Die Aktivitäten während der *Containment*-Phase bei kompromittierten S3 Buckets sind davon abhängig, ob die Kompromittierung des Buckets aufgrund von Miskonfigurationen oder kompromittierter Credentials eintritt.

8.5.1 Aktivieren des Block Public Access Features

Falls Daten in einem S3 Bucket durch eine Miskonfiguration öffentlich zugänglich gemacht wurde, spricht man von sogenannten *Public Buckets*. Die einfachste und effektivste Methode, um Public Buckets zu schützen, ist das Feature *Block Public Access*. Damit können Cloud-Nutzer verhindern, dass S3 Buckets öffentlich zugänglich gemacht werden. Somit ist es nicht mehr möglich, dass Daten öffentlich gelesen, geschrieben oder modifiziert werden [64, Blocking public access to your Amazon S3 storage].

Seit April 2023 haben alle neu erstellten S3 Buckets das Block Public Access Feature aktiviert [66]. Dennoch sollte in der *Containment*-Phase geprüft werden, ob diese Feature bereits aktiviert ist und dies gegebenenfalls nachträglich aktiviert werden. Dazu kann beispielsweise der nachfolgende CLI Befehl benutzt werden:

Quelltext 8.1: CLI Befehl zum Aktivieren des Block Public Access Features eines S3 Buckets

```
1 aws s3api put-public-access-block \
2     --bucket my-bucket \
3     --public-access-block-configuration "BlockPublicAcls=true,IgnorePublicAcls=true,
          BlockPublicPolicy=true,RestrictPublicBuckets=true"
```

Falls die Kompromittierung eins S3 Buckets allerdings von einer IAM Entität als Folge von kompromittierten Credentials durchgeführt wurde, reicht dieser Schritt noch nicht aus, um den Sicherheitsvorfall einzudämmen. In einem solchen Fall muss zusätzlich die Bucket Policy angepasst werden, was im nächsten Abschnitt erläutert wird.

8.5.2 Anpassen der Bucket Policy

Eine *Bucket Policy* ist eine ressourcenbasierte Richtlinie, mit der die Zugriffsberechtigungen für einen S3 Bucket und die darin enthaltenen Objekte verwaltet werden.

Bucket Policies können Anfragen basierend auf den Elementen in der Policy zulassen oder ablehnen. Zu diesen Elementen gehören der Anforderer, S3 API-Aktionen, Ressourcen und Bedingungen der Anfrage (z. B. die IP Adresse, die für eine Anfrage verwendet wird) [64, Using bucket policies].

Falls während der Analyse eines Sicherheitsvorfalls identifiziert wurde, dass ein S3 Bucket nicht aufgrund der Miskonfiguration sondern durch bestimmte IAM Entitäten kompromittiert wurde, sollten der Zugriff für diese Entitäten durch das Anpassen der Bucket Policy unterbunden werden. Dafür gibt es - ähnlich wie für die *Containment*-Phase kompromittierter Credentials in Kapitel 7.5.1 - unterschiedliche Ausprägungen, wie man Bucket Policies gestalten kann.

Diese Ausprägungen lassen sich in sogenannte *Whitelists* und *Blacklists* unterscheiden. Bei einer Whitelist werden nur bestimmte Personenkreise berechtigt und der Rest ausgeschlossen. Bei einer Blacklist werden nur bestimmte Personenkreise blockiert und der Rest berechtigt.

Die Entscheidung über den gewählten Ansatz zur Eindämmung eines Sicherheitsvorfalls sollte vom Incident Response Team nur nach Rücksprache mit den relevanten Stakeholdern und Entscheidungsträgern getroffen werden.

Whitelisting Ansätze

Durch Bucket Policies, die einen Whitelisting Ansatz verfolgen, können beispielsweise nur bestimmte Personenkreise auf Bais eines definierten IP Adressbereichs oder bestimmter IAM Entitäten für einen S3 Bucket berechtigt werden. Für alle Entitäten, die nicht von dieser Ausnahme betroffen sind, wird der Zugriff auf den Bucket verweigert.

Das ist dann sinnvoll, wenn sichergestellt werden muss, dass bestimmte Applikationen weiterhin Zugriff auf den jeweiligen S3 Bucket haben, um somit Serviceunterbrechungen zu vermeiden.

Whitelisting von IP Adressen In Anhang E.1 ist im Quelltext E.1 eine Bucket Policy dargestellt, die den Zugriff auf einen Bucket verhindert, es sei denn die Requests stammen aus einem definierten IP Adressbereich. Dieser Bereich muss im Parameter `<WHITELISTED_IP_ADDRESS>` eingetragen werden.

Whitelisting von IAM Entitäten In Anhang E.1 ist im Quelltext E.2 eine Bucket Policy dargestellt, die den Zugriff auf einen Bucket verhindert, es sei denn die Requests stammen von definierten IAM Entitäten. Diese Entitäten müssen im Parameter `<WHITELISTED_ENTITIES>` eingetragen werden.

Blacklisting Ansätze

Durch Bucket Policies, die einen Blacklisting Ansatz verfolgen, wird der Zugriff aus bestimmten IP Adressbereiche oder durch bestimmte Entitäten explizit blockiert.

Ein solcher Ansatz kann dann verwendet werden, wenn ein Angreifer eindeutig zu einem bestimmten IP Adressbereich oder bestimmten Entitäten zugeordnet werden konnte. Der Zugriff wird in diesem Fall nur für die explizit aufgeführten IP Adressbereich bzw. Entitäten geblockt. Dadurch kann sichergestellt werden, dass laufende Systeme oder Applikationen weiterhin auf den S3 Bucket zugreifen können.

Blacklisting von IP Adressen In Anhang E.2 ist im Quelltext E.3 eine Bucket Policy dargestellt, die den Zugriff auf einen Bucket nur für Requests verhindert, die aus einem definierten IP Adressbereich stammen. Dieser Bereich muss im Parameter `<BLACKLISTED_IP_ADDRESS>` eingetragen werden.

Blacklisting von IAM Entitäten In Anhang E.2 ist im Quelltext E.3 eine Bucket Policy dargestellt, die den Zugriff auf einen Bucket nur für Requests verhindert, die von bestimmten Entitäten durchgeführt werden. Diese Entitäten müssen im Parameter `<BLACKLISTED_ENTITIES>` eingetragen werden.

8.5.3 Weitere Containment Aktivitäten

Falls bei der Analyse des Sicherheitsvorfalls erkannt wurde, dass die Kompromittierung eines S3 Buckets im Zusammenhang mit Entitäten aus einem AWS Account der eigenen AWS Organization stehen, dann sollten die Incident Response Maßnahmen für kompromittierte Credentials angewandt werden, die in Kapitel 7 beschrieben sind.

8.6 Eradication

Bei der Kompromittierung von S3 Buckets betreffen die Maßnahmen in der *Eradication*-Phase, die Beseitigung modifizierter und neu erstellter Objekte.

Somit sind diese Schritte davon abhängig, wie sich eine S3 Bucket Kompromittierung konkret auswirkt. Durch die Analyse von Cloud Trail Data Events mittels Athena kann identifiziert werden, welche Objekte modifiziert bzw. neu erstellt wurden. Dabei kann der Suchbereich auf einen bestimmten Zeitpunkt oder bestimmte Entitäten eingeschränkt werden.

Falls CloudTrail Data Events oder S3 Server Access Logging zum Zeitpunkt des Sicherheitsvorfalls nicht aktiviert waren, kann das Attribut `Last modified` von Objekten in S3 Buckets verwendet werden, um neue oder modifizierte Objekte zu erkennen. Allerdings kann ohne diese zusätzlichen Logquellen keine Aussage getroffen werden, *von wem* die Objekte modifiziert oder neu erstellt wurden. Dadurch ist es ohne manuellen Aufwand von Applikationsverantwortlichen oder Entwicklerteams kaum möglich, eine legitime Modifizierung von Angreifern zu unterscheiden.

8.7 Recovery

Bei der Kompromittierung von S3 Buckets betreffen die Maßnahmen in der *Recovery*-Phase, die Wiederherstellung von gelöschten Objekten.

Falls bei einem Sicherheitsvorfall Daten aus einem kompromittierten S3 Bucket gelöscht wurden, dann hängt der Erfolg der Wiederherstellung komplett davon ab, ob vor dem Incident in irgendeiner Form Datensicherungen der Objekte erstellt wurden. Das können beispielsweise Datensicherungen aus *Backup Vaults* (vgl. Kapitel 6.5.2) oder aus anderen AWS Accounts sein.

9 Szenario III: Cryptomining Activities

In diesem Kapitel wird der effektive Einsatz relevanter AWS Services als Teil der Behandlung von Sicherheitsvorfällen beschrieben, welche im Zusammenhang mit Cryptomining Aktivitäten stehen.

9.1 Preparation - Preventing Incidents

Die Ursache für unerwünschte Cryptomining Aktivitäten in AWS Accounts von Organisationen basieren immer auf einer Form kompromittierte Credentials. Diese sind notwendig, um bestehende Infrastrukturressourcen zu modifizieren oder neue zu erstellen und damit Cryptomining zu betreiben.

Deshalb sind die Maßnahmen in der *Preparation*-Phase zur Vorbeugung kompromittierter Credentials (vgl. Kapitel 7.2) auch für dieses Szenario anwendbar.

9.2 Detection

9.2.1 Erkennung mittels GuardDuty

Der AWS native Threat Detection Service GuardDuty erzeugt eine Reihe von Findings, die für die Erkennung von Sicherheitsvorfällen im Zusammenhang mit Cryptomining Aktivitäten genutzt werden können. Je nach *Finding Type*, können dadurch Cryptomining Aktivitäten in unterschiedlichen AWS Services erkannt werden.

Nachfolgend sind die unterschiedlichen Finding Types beschrieben sowie die jeweiligen Findings aufgeführt, die auf Cryptomining Aktivitäten hinweisen:

- **EC2 Findings**: Diese Findings beziehen sich auf den Rechenservice Amazon Elastic Compute Cloud (EC2). Die Findings beinhalten immer `Instance` als Resource Type [51, GuardDuty EC2 finding types]. Für diesen Finding Type gibt es derzeit diese Findings:
  - `CryptoCurrency:EC2/BitcoinTool.B`,

  - `CryptoCurrency:EC2/BitcoinTool.B!DNS` und
  - `Impact:EC2/BitcoinDomainRequest.Reputation`

- **Runtime monitoring Findings**: *Runtime monitoring* ist ein Feature von GuardDuty, wodurch Sicherheitsvorfälle in Amazon EKS Clustern zur Laufzeit entdeckt werden können. Seit November 2023 können zusätzlich Sicherheitsvorfälle auf EC2 Instanzen durch die Analyse von Aktivitäten auf dem Host-OS-Level erkannt werden [67]. Für diesen Finding Type werden derzeit die folgenden Findings generiert:
  - `CryptoCurrency:Runtime/BitcoinTool.B`,
  - `CryptoCurrency:Runtime/BitcoinTool.B!DNS`,
  - `Impact:Runtime/BitcoinDomainRequest.Reputation` und
  - `Impact:Runtime/CryptoMinerExecuted`
- **Lambda protection Findings**: Diese Findings beziehen sich auf den Rechenservice *AWS Lambda*. GuardDuty analysiert die Netzwerkaktivitäten von ausgeführten Lambda Funktionen - zum Beispiel in Form von VPC Flow Logs (vgl. Kapitel 6.2.1) - und generiert daraufhin Findings bei erkannten Sicherheitsvorfällen [51, Lambda Protection in Amazon GuardDuty]. Derzeit gibt es für diesen Finding Type lediglich ein Finding:
  - `CryptoCurrency:Lambda/BitcoinTool.B`

9.2.2 Implementierung benutzerdefinierter Detektionsmaßnahmen

In diesem Abschnitt werden einige Ansätze zur Implementierung benutzerdefinierter Detektionsmaßnahmen von Cryptomining Aktivitäten in AWS Accounts vorgestellt.

Monitoring von Route53 Resolver Logs

Bei diesem Ansatz, wird versucht anhand von Route53 Resolver Logs (vgl. Kapitel 6.2.1) DNS Abfragen zu Domains zu erkennen, welche im Zusammenhang mit Cryptomining Aktivitäten stehen. Dabei können sogenannte *Threat Intelligence Feeds* als Quelle für diese Domains verwendet werden.

Abbildung 14 zeigt eine High-Level-Architektur eines solchen Ansatzes. Dabei werden die DNS Abfragen von EC2 Instanzen durch Route53 Resolver Logs zu CloudWatch weitergeleitet. Eine CloudWatch Rule, gleicht dann die abgefragten Domains mit einem Threat Intelligence Feed ab. Sobald es eine Übereinstimmung der abgefragten Domains gibt, wird über ein SNS Topic das Incident Response Team per E-Mail benachrichtigt.

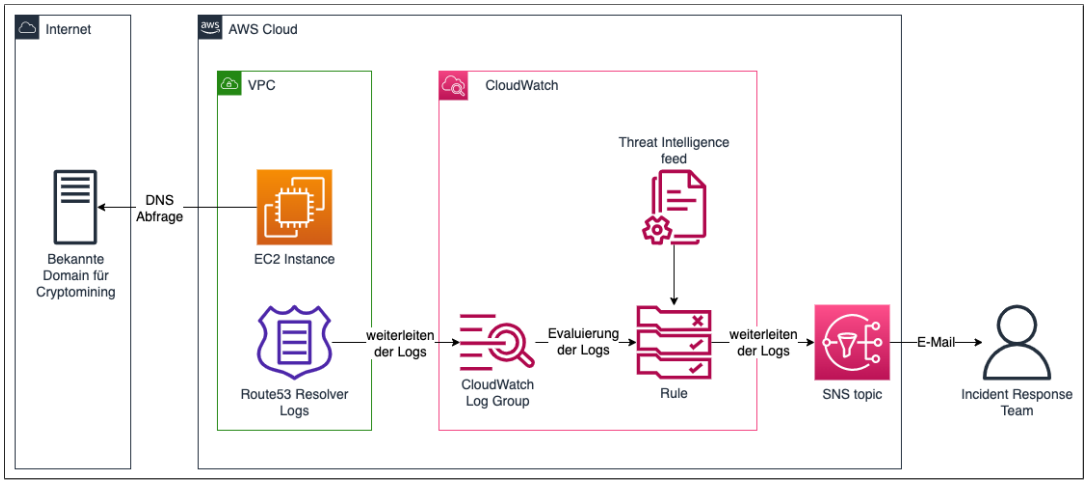


Abbildung 14: Monitoring von Route53 Resolver Logs zur Erkennung von Cryptomining Aktivitäten

Diese Ansatz ist letztlich eine benutzerdefinierte Implementierung des GuardDuty Findings `CryptoCurrency:EC2/BitcoinTool.B!DNS`.

Monitoring von VPC Flow Logs

Bei diesem Ansatz werden anhand von VPC Flow Logs (vgl. Kapitel 6.2.1) Nertzwerkverbindungen von EC2 Intsanzen zu IP Adressen identifiziert, welche im Zusammenhang mit Cryptomining Aktivitäten stehen. Auch hier können wieder Threat Intelligence Feeds als Quelle für bekannte IP Adressen verwendet werden.

Diese Ansatz ist letztlich eine benutzerdefinierte Implementierung des GuardDuty Findings `CryptoCurrency:EC2/BitcoinTool.B`.

Monitoring von CPU/GPU Metriken

Bei diesem Ansatz werden die Central Processing Unit (CPU) und Graphics Processing Unit (GPU) Auslastung von EC2 Instanzen überwacht. Cryptominer führen

meist rechenintensive Operationen aus, was sich auf den infizierten Hosts in Form hoher Auslastung von CPU und GPU auswirkt. Diese Belastungsspitzen können als potentieller Indikator für Cryptomining Aktivitäten genutzt werden. Hier kann auch wieder ein CloudWatch Alarm bei der Überschreitung eines bestimmten Schwellwerts der CPU und GPU Auslastung eingerichtet werden, der dann wiederum das Incident Response Team benachrichtigt.

Allerdings kann es hier zu einer hohen Quote an *false-poisitives* kommen, wenn die CPU oder GPU Auslastung durch legitime Lastspitzen der Applikationen kommt.

9.3 Analysis

9.3.1 Vorgehensweise

Die folgenden Fragestellungen eignen sich als Leitfaden für die Analyse im Falle von erkannten Crypomining Aktivitäten:

1. Welche EC2 Instanzen sind konkret betroffen?
2. Sind diese EC2 Instanzen teil einer Autoscaling Group?
3. Welche Entitäten sind für Änderungen an Ressourcen verantwortlich bzw. starteten neue Ressourcen?
4. Welche Änderungen wurden an den identifizierten Ressourcen durchgeführt
5. Kam es zu Netzwerkverbindungen zu anderen Hosts (sowohl innerhalb des AWS Accounts als auch außerhalb)?

9.3.2 Logquellen für Analyse

Um die oben aufgeführten Fragestellungen zu beantworten, können verschiedene Logquellen herangezogen werden.

GuardDuty Findings Anhand von GuardDuty Findings lassen sich bereits eine Reihe wichtiger Informationen ableiten. Dazu gehören:

- `InstanceDetails`: Aus diesem Feld lassen sich die *Instance ID*, das *VPC* sowie die *IP Adresse* der betroffene EC2 Instanzen ableiten.

- `NetworkConnectionAction`: Aus diesem Feld lassen sich weitere Informationen zur erkannten Netzwerkverbindungen ableiten, wie zum Beispiel die *IP Adresse*, *Connection Direction* sowie Details zu den verwendeten Quell- und Zielports.

VPC Flow Logs und Route53 Logs Anhand der Analyse von VPC Flow Logs und Route 53 Logs lassen sich eine Reihe von Informationen zum Netzwerkverkehr der betroffenen EC2 Instanz ableiten. Zum Beispiel mit welchen IP Adressen und über welche Ports Netzwerkverbindungen ein- bzw. ausgegangen sind oder welche DNS Abfragen ausgeführt wurden.

Config Logs Anhand der Analyse von Config Logs können die aufgezeichneten *Configuration Items* (vgl. Kapitel 6.2.1) Änderungen an den betroffenen EC2 Instanzen identifiziert werden. So lassen sich auch Änderungen an Ressourcen, die mit der EC2 Instanz verknüpft sind - zum Beispiel VPCs, Routing Tabellen, Firewall Konfigurationen etc. - analysieren.

CloudTrail Logs Zudem empfiehtl es sich auch hier wieder CloudTrail Logs zu analysieren, um weitere Informationen zu Entitäten zu erhalten, die Änderungen an Ressourcen ausführten bzw. neue Ressourcen erstellten.

9.4 Containment

Die *Containment*-Phase im Kontext kompromittierter EC2 Intsanzen umfasst die folgenden Aufgaben in einer sequentiellen Reihenfolge: Sicherung forensischer Artefakte, Isolierung des Netzwerks und Isolierung von Credentials.

Poling empfiehlt, dass die Sicherung forensischer Artefakte - speziell eine Kopie des volatilen Arbeitsspeichers - zwingend vor der Isolierung des Netzwerks der betroffenen EC2 Instanzen geschieht. Als Grund dafür führt Poling auf, dass durch die Isolierung des Netzwerks die Informationen zu aktiven Netzwerkverbindungen verloren gehen und somit nicht in einer nachgelagerten Kopie des Arbeitsspeichers gesichert werden können [68, S. 6–11].

9.4.1 Sicherung forensischer Artefakte

Die Sicherung forensischer Artefakte umfasst das Erstellen von Kopien des Arbeitsspeichers sowie dem Erstellen von Snapshots der Datenspeicher.

AWS stellt mit dem *Automated Forensics Orchestrator for Amazon EC2* eine Lösung bereit, die zur Orchestrierung und Automatisierung der Erstellung forensischer Artefakte von EC2 Instanzen genutzt werden kann [69].

Arbeitsspeicher Hierbei handelt es sich um volatile Daten, die beim Abschalten der EC2 Instanz verloren gehen. Der Arbeitsspeicher enthält eine Reihe wertvolle forensischer Artefakte, wie zum Beispiel laufende Prozesse, aktive Netzwerkverbindungen, gelöschte Dateien und Systemlogs [68, S. 6].

Datenspeicher Die Sicherung von Datenspeicher lässt sich im Kontext von EC2 basierten Hosts durch Snapshots der Laufwerke umsetzen. EC2 Instanzen nutzen sogenannte *Amazon Elastic Block Store (EBS) Volumes* als Laufwerke. Wie bereits in Kapitel 3.1.3 erwähnt, ist ein Vorteil der Incident Response in der Cloud, die Verfügbarkeit von APIs, mit deren Hilfe sich die Erstellung von Snapshots automatisieren und beschleunigen lässt. So kann mit dem folgenden CLI Kommando ein Snapshot eines EBS Volumes erstellt werden.

Quelltext 9.1: CLI Befehl zum Erstellen eines Snapshots eines EBS Volumes

```
1 aws ec2 create-snapshot \
2     --volume-id vol-1234567890abcdef0 \
3     --description "This is my root volume snapshot"
```

9.4.2 Netzwerk Isolierung

Die Isolierung des Netzwerks der EC2 zielt darauf ab, die kompromittierte Instanz auf Netzwerkebene einzudämmen. Dadurch wird verhindert, dass die kompromittierte Instanz mit Servern des Angreifers oder weiteren Ressourcen innerhalb des AWS Account kommuniziert. Dies trägt dazu bei die Verbreitung weiterer Cryptominer zu stoppen, mögliche laufende Datenexfiltration zu stoppen und zu verhindern, dass das kompromittierte System für weitere Angriffe genutzt wird.

Die Isolierung des Netzwerks kann durch auf verschiedenen Ebenen mittels verschiedener AWS Services und Methoden durchgeführt werden.

Isolierung auf Instanz Ebene Die feingranularste Form der Netzwerkisolierung erfolgt durch das Anfügen einer isolierenden *Security Group*. Eine Security Group ist eine Art Netzwerk Firewall auf Instanz Ebene, die den eingehenden und ausgehenden Datenverkehr einer EC2 Instanz kontrolliert [45, Control traffic to your AWS resources using security groups].

Durch das Anfügen einer *Isolation Security Group*, welche keine *Allow Rules* für eingehenden und ausgehenden Netzwerkverkehr hat, kann eine EC2 Instanz vollständig isoliert werden.

Isolierung auf Subnetz Ebene Eine weitere Option ist die Isolierung des Datenverkehrs auf Subnetzebene, durch sogenannte Network Access Control Lists (NACLs). Eine NACL erlaubt oder verweigert bestimmten eingehenden oder ausgehenden Datenverkehr auf Subnetzebene [45, Control traffic to subnets using network ACLs].

Anders als Security Groups können NACLs nicht an einzelne EC2 Instanzen angefügt werden, sondern nur an Subnetze. Dadurch werden alle EC2 Instanzen des Subnetzes isoliert, was hilfreich ist, wenn mehrere Instanzen kompromittiert wurden.

9.4.3 Entitäten Isolierung

Sobald bei der Analyse der Cryptomining Aktivtäten eine bzw. mehrere Entitäten identifiziert wurden, müssen diese isoliert werden. An dieser Stelle sei an die Aktivitäten der *Containment*-Phase im Rahmen kompromittierter Credentials verwiesen, welche in Kapitel 7.5 im Detail beschrieben sind.

9.5 Eradication

In der *Eradication*-Phase im Zusammenhang von Cryptomining Aktivitäten ist das Ziel, modifizierte sowie neu erstellte Ressourcen zu beseitigen. Das betrifft konkret alle EC2 Instanzen, auf denen während der Analyse-Phase Cryptominer identifiziert wurden.

Falls bei der Analyse des Sicherheitsvorfalls erkannt wurde, dass neue EC2 Instanzen als Teil einer sogenannten *Auto Scaling Group* erstellt wurde, reicht es nicht aus, nur die jeweiligen EC2 Instanzen zu löschen. Eine Auto Scaling Group besteht aus einer Sammlung von EC2 Instanzen, die zum Zweck der automatischen Skalierung und Verwaltung als logische Gruppierung behandelt werden [70, Auto Scaling groups].

Wenn lediglich einzelne EC2 Instanzen gelöscht werden würden, hätte dies zur Folge, dass die Auto Scaling Group automatisch weiter Instanzen erstellt. Deshalb muss in diesem Fall die gesamte AutoScaling Group beseitigt werden.

9.6 Recovery

In der *Eradication*-Phase im Zusammenhang von Cryptomining Aktivitäten ist das Ziel, Systeme in ihren ursprünglichen funktionierenden Zustand zu versetzen. Das bedeutet konkret, dass kompromittierte Instanzen in ihren ursprünglichen Zustand vor dem Befall von Cryptominern wiederhergestellt werden.

Der Erfolg der Aktivitäten bei der Wiederherstellung ist davon abhängig, welche Maßnahmen in der *Preparation*-Phase getroffen wurden. Falls beispielsweise täglich Snapshots von EBS Volumes durch Amazon Backup erstellt wurden, können betroffene Systeme auf Basis der Sicherungskopien in einem Backup Vault wiederhergestellt werden (vgl. Abbildung 11).

10 Fazit

10.1 Zusammenfassung

In dieser Master-Thesis wurde neben der Betrachtung der theoretischen Grundlagen der Incident Response und des Cloud Computings auch eine Abgrenzung von Incident Reponse in on-premise im Vergleich zu Cloud Umgebungen durchgeführt. Dabei wurden sowohl Gemeinsamkeiten als auch Unterschiede identifiziert. Der größte Unterschied liegt in der Verfügbarkeit und dem Detaillierungsgrad von Logs, welche eine Voraussetzung für effektive Incident Response sind.

In Kapitel 6 konnte gezeigt werden, dass bestimmte Cloud Services vor allem für Maßnahmen in der *Prepare*-Phase des Incident Response Prozess allgemeingültig einsetzbar sind. Dazu gehören der Aufbau einer AWS Account Struktur mittels *AWS Organization* und eines Asset Inventory mittels *AWS Config* sowie die Implementierung einer Backup Strategie mittels *AWS Backup* und der AWS Security Services *Amazon GuardDuty*, *Amazon Inspector* und *AWS Security Hub*. Ein weiterer Bestandteil ist die Auswahl der notwendigen Logquellen, das Einrichten von geeigneten Analyse Mechanismen sowie der Aufbau von Alarmsystemen auf Basis der Logs.

In den Kapiteln 7 bis 9 wurde untersucht, welche AWS Services in welche Phasen des NIST Incident Response Prozesses erfolgreich integriert werden. Dabei konnte gezeigt werden, dass je nach betrachtetem Szenario unterschiedliche Cloud Services in den Phasen *Analyse*, *Containment*, *Eradication* und *Recovery* anzuwenden sind.

10.2 Ausblick

In diesem Abschnitt wird ein Ausblick auf weitere mögliche Forschung im Gebiet der Incident Response in Cloud Umgebungen gegeben.

Der Fokus dieser Thesis liegt auf der Integration von AWS Services in Incident Response Prozesse in Organisationen, da AWS über den größten Marktanteil in Bereich

Cloud Computing verfügt. Dennoch gibt es weitere Cloud-Anbieter wie GCP oder Microsoft Azure, die ebenfalls verschiedene Services anbieten, welche für die Behandlung von Sicherheitsvorfällen eingesetzt werden können. Eine Fortführung der Master-Thesis könnte die Betrachtung der Integrationsmöglichkeiten von Services anderer Cloud-Anbieter in Incident Respose Prozessen sein.

Die letzten Jahren zeigten eine zunehmende Verbreitung des Einsatzes von generativer künstlicher Intelligenz sowie signifikante Fortschritte der Qualität der zugrundeliegenden Large Language Models (LLMs). Ein weiteres mögliches Forschungsgebiet liegt Potentialanalyse des Einsatzes von generativer künstlichen Intelligenz in Incident Response Prozessen. Hier könnte beispielsweise untersucht werden, wie generative künstliche Intelligenz für die automatisierte Behandlung von Sicherheitsvorfällen eingesetzt werden kann.

Literaturverzeichnis

[1] International Organization for Standardization and the International Electrotechnical Commission, "ISO/IEC 27000:2018, Information technology — Security techniques — Information security management systems — Overview and vocabulary," Techn. Ber., Juni 2018. Adresse: https://standards.iso.org/ittf/PubliclyAvailableStandards/c073906_ISO_IEC_27000_2018_E.zip.

[2] P. Cichonski, T. Millar, T. Grance und K. Scarfone, "NIST Special Publication 800-61 Rev. 2, Computer Security Incident Handling Guide," Techn. Ber., Aug. 2012. DOI: 10.6028/nist.sp.800-61r2. Adresse: https://doi.org/10.6028/nist.sp.800-61r2.

[3] Bundesamt für Sicherheit in der Informationstechnik, Deutschland (BSI, *IT-Grundschutz-Kompendium*, 6. Edition. Reguvis, 2023, ISBN: 978-3-8462-0906-6.

[4] Bundesamt für Sicherheit in der Informationstechnik, Deutschland (BSI, *BSI-IT-Grundschutz-Schulung*, https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/IT-Grundschutz/Zertifizierte-Informationssicherheit/IT-Grundschutzschulung/it-grundschutzschulung_node.html, Accessed: 2023-07-23, 2018.

[5] T. Brennan und J. Jolo, "Top 10 Considerations For Incident Response," The OWASP Foundation, Techn. Ber., Dez. 2015, Accessed: 2023-07-26. Adresse: https://owasp.org/www-pdf-archive/Top10ConsiderationsForIncidentResponse.pdf.

[6] The OWASP Foundation, *About the OWASP Foundation*, https://owasp.org/about/, Accessed: 2023-07-26, 2023.

[7] International Organization for Standardization and the International Electrotechnical Commission, "ISO/IEC 27035:2023, Information technology - Information security incident management," Techn. Ber., Feb. 2023.

[8] International Organization for Standardization and the International Electrotechnical Commission, "ISO/IEC 27035-1:2023, Information technology - Information security incident management - Part 1: Principles and process," Techn. Ber., Feb. 2023.

[9] Bundesamt für Sicherheit in der Informationstechnik, Deutschland (BSI, *Cloud Computing Grundlagen*, Accessed: 2023-11-10, Jan. 2021. Adresse: https://www.bsi.bund.de/dok/6622124.

[10] P. M. Mell und T. Grance, "NIST SP 800-145, The NIST definition of cloud computing," Techn. Ber., 2011. DOI: 10.6028/nist.sp.800-145. Adresse: https://doi.org/10.6028/nist.sp.800-145.

[11] Prof. Dr. rer. nat. Nils Gruschka, *Sicherheit im Cloud Computing*, Studienbrief Sommersemester 2022.

[12] Amazon Web Services Inc., "Amazon Web Services: Risk and Compliance," Amazon Web Services Inc., Techn. Ber., März 2021. Adresse: https://docs.aws.amazon.com/whitepapers/latest/aws-risk-and-compliance/welcome.html.

[13] J. Lemon und M. Roddie, "DFIR Evidence Collection and Preservation for the Cloud," SANS DFIR Summit 2022, SANS, Aug. 2022. Adresse: https://sansorg.egnyte.com/dl/e1xvOJOk5W.

[14] Bundesamt für Verfassungsschutz, *Indicators of Compromise*, Mai 2022. Adresse: https://www.verfassungsschutz.de/SharedDocs/glosaareintraege/DE/I/ioc.html.

[15] B. Bracken, *Microsoft 'Logging Tax' Hinders Incident Response, Experts Warn*, Accessed: 2023-11-09, Juli 2023. Adresse: https://www.darkreading.com/remote-workforce/microsoft-logging-tax-hinders-incident-response.

[16] L. Dobberstein, *Under CISA ~~pressure~~ collab, Microsoft makes cloud security logs available for free*, Accessed: 2023-11-09, Juli 2023. Adresse: https://www.theregister.com/2023/07/20/under_cisa_spressures_collaboration_microsoft/.

[17] A. McAbee, "AWS Security Incident Response Guide," Amazon Web Services Inc., Techn. Ber., Jan. 2023. Adresse: https://docs.aws.amazon.com/whitepapers/latest/aws-security-incident-response-guide/aws-security-incident-response-guide.html.

[18] M. Cronin und A. Schneider, “Threat detection and incident response using cloud-native services,” AWS re:invent 2022, Amazon Web Services Inc., Nov. 2022. Adresse: https://d1.awsstatic.com/events/Summits/reinvent2022/SEC309_Threat-detection-and-incident-response-using-cloud-native-services.pdf.

[19] *Amazon Maintains Lead in the Cloud Market*.

[20] The MITRE Corporation, *Initial Access*, Accessed: 2023-11-18, Nov. 2023. Adresse: https://attack.mitre.org/tactics/TA0001/.

[21] GitGuardian, *The State of Secrets Sprawl 2023*, Accessed: 2023-11-18, März 2023. Adresse: https://www.gitguardian.com/state-of-secrets-sprawl-report-2023.

[22] T. Forbes, *I scanned every package on PyPi and found 57 live AWS keys*, Accessed: 2023-11-18, Jan. 2023. Adresse: https://tomforb.es/i-scanned-every-package-on-pypi-and-found-57-live-aws-keys/.

[23] Cloud Security Alliance, *Top Threats to Cloud Computing - The Egregious 11*, Apr. 2020.

[24] Datadog Inc., *State of Cloud Security 2023*, Accessed: 2023-11-18, Nov. 2023. Adresse: https://www.datadoghq.com/state-of-cloud-security/.

[25] IBM, *Cost of a Data Breach Report 2023*, März 2023. Adresse: https://www.ibm.com/reports/data-breach.

[26] Amazon Web Services Inc., *Amazon S3*, Accessed: 2023-11-14, Nov. 2023. Adresse: https://aws.amazon.com/de/s3/.

[27] J. Barr, *Celebrate 15 Years of Amazon S3 with "‘Pi Week"’ Livestream Events*, Accessed: 2023-11-14, März 2021. Adresse: https://aws.amazon.com/de/blogs/aws/amazon-s3s-15th-birthday-it-is-still-day-1-after-5475-days-100-trillion-objects/.

[28] S. Gietzen, *S3 Ransomware Part 1: Attack Vector*, Juni 2019. Adresse: https://rhinosecuritylabs.com/aws/s3-ransomware-part-1-attack-vector/.

[29] K. Traxler, *Cloud-Native Ransomware, How attacks on availability leverage cloud services*, Apr. 2022. Adresse: https://content.vectra.ai/hubfs/downloadable-assets/WhitePaper_Cloud_Native_Ransomware.pdf.

[30] Federal Trade Commission, *FTC Matter/File Number 1923170*, Accessed: 2023-11-16, Juni 2023. Adresse: https://www.ftc.gov/system/files/ftc_gov/pdf/complaint.pdf.

[31] A. Sharma, *U.S. No Fly list shared on a hacking forum, government investigating*, Accessed: 2023-11-16, Jan. 2023. Adresse: https://www.bleepingcomputer.com/news/security/us-no-fly-list-shared-on-a-hacking-forum-government-investigating/.

[32] S. Eskandari, A. Leoutsarakos, T. Mursch und J. Clark, "A First Look at Browser-Based Cryptojacking," in *2018 IEEE European Symposium on Security and Privacy Workshops*, Apr. 2018, S. 58–66. DOI: 10.1109/EuroSPW.2018.00014.

[33] S. de Vera, *AWS CIRT announces the release of five publicly available workshops*, Accessed: 2023-11-14, Dez. 2022. Adresse: https://aws.amazon.com/de/blogs/security/aws-cirt-announces-the-release-of-five-publicly-available-workshops/.

[34] Snyk Limited, *The State of Cloud Security Report 2022*, Accessed: 2023-11-14, Sep. 2022. Adresse: https://resources.snyk.io/state-of-cloud-security.

[35] Federal Trade Commisison, *FTC Matter/File Number 2023185*, Accessed: 2023-11-14, Okt. 2022. Adresse: https://www.ftc.gov/system/files/ftc_gov/pdf/202-3185-Drizly-Complaint.pdf.

[36] P. Kunert, *DXC spills AWS private keys on public GitHub*, Accessed: 2023-11-14, Nov. 2017. Adresse: https://www.theregister.com/2017/11/14/dxc_github_aws_keys_leaked/.

[37] Amazon Web Services Inc., *AWS Account Management, AWS documentation*, Accessed: 2023-11-14, Nov. 2023. Adresse: https://docs.aws.amazon.com/accounts/latest/reference/accounts-welcome.html.

[38] Amazon Web Services Inc., *AWS Organizations documentation*, Accessed: 2023-11-14, Nov. 2023. Adresse: https://docs.aws.amazon.com/organizations/latest/userguide/orgs_introduction.html.

[39] Amazon Web Services, Inc., "AWS Security Reference Architecture," Amazon Web Services Inc., Techn. Ber., Nov. 2023. Adresse: https://docs.aws.amazon.com/prescriptive-guidance/latest/security-reference-architecture/welcome.html.

[40] A. McAbee u. a., *Logging strategies for security incident response*, Apr. 2023. Adresse: https://aws.amazon.com/blogs/security/logging-strategies-for-security-incident-response/.

[41] V. Drake, *Threat Modeling*, Accessed: 2023-11-14, Nov. 2023. Adresse: https://owasp.org/www-community/Threat_Modeling.

[42] *Assisted Log Enabler for AWS, Find resources that are not logging, and turn them on.*

[43] Amazon Web Services, Inc., *AWS CloudTrail documentation*, Accessed: 2023-11-14, Nov. 2023. Adresse: https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-user-guide.html.

[44] Amazon Web Services, Inc., *AWS CloudTrail Adds Support for AWS Organizations*, Accessed: 2023-11-14, Nov. 2018. Adresse: https://aws.amazon.com/about-aws/whats-new/2018/11/aws-cloudtrail-adds-support-for-aws-organizations/.

[45] Amazon Web Services, Inc., *Amazon Virtual Private Cloud documentation*, Accessed: 2023-11-14, Nov. 2023. Adresse: https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html.

[46] Amazon Web Services, Inc., *Amazon Route 53 documentation*, Accessed: 2023-11-14, Nov. 2023. Adresse: https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/Welcome.html.

[47] Amazon Web Services, Inc., *AWS Config documentation*, Accessed: 2023-11-14, Nov. 2023. Adresse: https://docs.aws.amazon.com/config/latest/developerguide/WhatIsConfig.html.

[48] Amazon Web Services, Inc., *Amazon Athena documentation*, Accessed: 2023-11-14, Nov. 2023. Adresse: https://docs.aws.amazon.com/athena/latest/ug/what-is.html.

[49] Amazon Web Services, Inc., *AWS Security Analytics Bootstrap*, Accessed: 2023-11-14, Juni 2023. Adresse: https://github.com/awslabs/aws-security-analytics-bootstrap.

[50] Amazon Web Services, Inc., *Amazon Athena features*, Accessed: 2023-11-14, Nov. 2023. Adresse: https://aws.amazon.com/athena/features/.

[51] Amazon Web Services, Inc., *Amazon GuardDuty documentation*, Accessed: 2023-11-14, Nov. 2023. Adresse: https://docs.aws.amazon.com/guardduty/latest/ug/what-is-guardduty.html.

[52] Amazon Web Services, Inc., *Amazon Inspector documentation*, Accessed: 2023-11-14, Nov. 2023. Adresse: https://docs.aws.amazon.com/inspector/latest/user/what-is-inspector.html.

[53] Amazon Web Services, Inc., *AWS SecurityHub documentation*, Accessed: 2023-11-14, Nov. 2023. Adresse: https://docs.aws.amazon.com/securityhub/latest/userguide/what-is-securityhub.html.

[54] Amazon Web Services, Inc., *Tagging AWS Resources*, Accessed: 2023-11-14, Nov. 2023. Adresse: https://docs.aws.amazon.com/tag-editor/latest/userguide/tagging.html.

[55] Amazon Web Services, Inc., *AWS Backup documentation*, Accessed: 2023-11-14, Nov. 2023. Adresse: https://docs.aws.amazon.com/aws-backup/latest/devguide/whatisbackup.html.

[56] Amazon Web Services, Inc., *Security best practices in IAM*, Accessed: 2023-11-14, Juni 2023. Adresse: https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html.

[57] Amazon Web Services, Inc., *AWS git-secrets*, Accessed: 2023-11-14, Juni 2023. Adresse: https://github.com/awslabs/git-secrets.

[58] M. Schweizer, *Managing aged access keys through AWS Config remediations*, Juni 2020. Adresse: https://aws.amazon.com/blogs/mt/managing-aged-access-keys-through-aws-config-remediations/.

[59] C. Farris, *Sensitive IAM Actions*, Sep. 2023. Adresse: https://github.com/primeharbor/sensitive_iam_actions.

[60] C. Farris, *Incident Response in AWS*, Aug. 2022. Adresse: https://www.chrisfarris.com/post/aws-ir/.

[61] Amazon Web Services, Inc., *AWS Iam documentation*, Accessed: 2023-11-14, Nov. 2023. Adresse: https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html.

[62] Europäische Union, *Datenschutz-Grundverordnung, Verordnung (EU) 2016/679 des Europäischen Parlaments und des Rates vom 27. April 2016 zum Schutz natürlicher Personen bei der Verarbeitung personenbezogener Daten, zum freien Datenverkehr und zur Aufhebung der Richtlinie 95/46/EG*.

[63] Amazon Web Services, Inc., *AWS CloudTrail pricing*, Dez. 2023. Adresse: https://aws.amazon.com/cloudtrail/pricing/.

[64] Amazon Web Services, Inc., *Amazon Simple Storage Service documentation*, Dez. 2023. Adresse: https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html.

[65] Amazon Web Services, Inc., *Amazon Macie documentation*, Dez. 2023. Adresse: https://docs.aws.amazon.com/macie/latest/user/what-is-macie.html.

[66] Amazon Web Services, Inc., *Advanced Notice: Amazon S3 will automatically enable S3 Block Public Access and disable access control lists for all new buckets starting in April 2023*, Dez. 2022. Adresse: https://aws.amazon.com/about-aws/whats-new/2022/12/amazon-s3-automatically-enable-block-public-access-disable-access-control-lists-buckets-april-2023/.

[67] Amazon Web Services, Inc., *Amazon GuardDuty now supports runtime monitoring for Amazon EC2 (Preview)*, Nov. 2023. Adresse: https://aws.amazon.com/about-aws/whats-new/2023/11/amazon-guardduty-runtime-monitoring-amazon-ec2-preview/.

[68] J. Poling, "Instance memory acquisition techniques for effective incident response," Amazon Webservices Inc., Juli 2022. Adresse: https://d1.awsstatic.com/events/aws-reinforce-2022/TDR401_Instance-memory-acquisition-techniques-for-effective-incident-response.pdf.

[69] Amazon Web Services, Inc., *Automated Forensics Orchestrator for Amazon EC2*, Juli 2023. Adresse: https://aws.amazon.com/solutions/implementations/automated-forensics-orchestrator-for-amazon-ec2/.

[70] Amazon Web Services Inc., *Instance metadata and user data, EC2 documentation*, Accessed: 2023-11-14, Nov. 2023. Adresse: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-metadata.html.

Abbildungsverzeichnis

Quelltextverzeichnis

Abkürzungsverzeichnis

Amazon EKS Amazon Elastic Kubernetes Service

Amazon RDS Amazon Relational Database Service

API Application Programming Interface

ARN Amazon Resource Name

AWS Amazon Web Services

BSI Bundesamt für Sicherheit in der Informationstechnik

CIS Center for Internet Security

CISA Cybersecurity and Infrastructure Security Agency

CLI AWS Command Line Interface

CPU Central Processing Unit

CSA Cloud Security Alliance

CSP Cloud Service Provider

CVE Common Vulnerabilities and Exposures

DDoS Distributed Denial of Service

DER Detektion und Reaktion

DFIR Digital Forensics/Incindet Response

DNS Domain Name System

DSGVO Datenschutz-Grundverordnung

EBS Amazon Elastic Block Store

EC2 Amazon Elastic Compute Cloud

ENISA European Network and Information Security Agency

FTC Federal Trade Commission

GCE Google Compute Engine

GCP Google Cloud Platform

GPU Graphics Processing Unit

GSK IT-Grundschutzkompendium

IaaS Infrastructure as a service

IANA Internet Assigned Numbers Authority

ICT Information and Communication Technology

IdP Identity Provider

IoC Indicators of Compromise

ISO International Organization for Standardization

JSON Java Script Object Notation

KRITIS Kritische Infrastrukturen

LLM Large Language Model

MFA Multi-Faktor-Authentisierung

NACL Network Access Control List

NIST National Institute of Standards and Technology

OS Operating System

OU Organizational Unit

OWASP Open Worldwide Application Security Project

PaaS Platform as a service

PCAP Packet Capture

PyPi Python Package Index

RAM Random-Access Memory

S3 Amazon Simple Storage Service

SaaS Software as a service

SAML Security Assertion Markup Language

SCP Service Control Policy

SIEM Security Information and Event Management

SLA Service Level Agreement

SNS Amazon Simple Notification Service

SQL Structured Query Language

SSH Secure Shell

vCPU Virtual Central Processing Unit

VPC Amazon Virtual Private Cloud

WAF Web Application Firewall

A Beispiele für Logs

A.1 VPC Flow Log Reord Beispiel

Quelltext A.1: Example of a VPC FLow Log Record

```
1  version account-id interface-id srcaddr dstaddr srcport dstport protocol packets bytes
       start end action log-status vpc-id subnet-id instance-id tcp-flags type pkt-srcaddr
       pkt-dstaddr region az-id sublocation-type sublocation-id pkt-src-aws-service pkt-dst
       -aws-service flow-direction traffic-path
2  5 532258145957 eni-00271afc6a81d8acb - - - - - - - 1700590779 1700590810 - NODATA vpc-02
       bdfd80211d2b1a9 subnet-08de94822cb1931c6 - - - - - eu-central-1 euc1-az2 - - - - - -
3  5 532258145957 eni-00271afc6a81d8acb - - - - - - - 1700590779 1700590810 - NODATA vpc-02
       bdfd80211d2b1a9 subnet-08de94822cb1931c6 - - - - - eu-central-1 euc1-az2 - - - - - -
4  5 532258145957 eni-00271afc6a81d8acb - - - - - - - 1700590795 1700590826 - NODATA vpc-02
       bdfd80211d2b1a9 subnet-08de94822cb1931c6 - - - - - eu-central-1 euc1-az2 - - - - - -
5  5 532258145957 eni-00271afc6a81d8acb - - - - - - - 1700590798 1700590829 - NODATA vpc-02
       bdfd80211d2b1a9 subnet-08de94822cb1931c6 - - - - - eu-central-1 euc1-az2 - - - - - -
6  5 532258145957 eni-00271afc6a81d8acb 45.148.10.241 192.168.0.177 4082 19 17 1 29
       1700590861 1700590861 ACCEPT OK vpc-02bdfd80211d2b1a9 subnet-08de94822cb1931c6 - 0
       IPv4 45.148.10.241 192.168.0.177 eu-central-1 euc1-az2 - - - - ingress -
7  5 532258145957 eni-00271afc6a81d8acb 44.213.45.216 192.168.0.177 0 0 1 1 28 1700590861
       1700590861 ACCEPT OK vpc-02bdfd80211d2b1a9 subnet-08de94822cb1931c6 - 0 IPv4
       44.213.45.216 192.168.0.177 eu-central-1 euc1-az2 - - EC2 - ingress -
8  5 532258145957 eni-00271afc6a81d8acb 78.128.114.174 192.168.0.177 49751 61766 6 1 40
       1700590848 1700590849 ACCEPT OK vpc-02bdfd80211d2b1a9 subnet-08de94822cb1931c6 - 2
       IPv4 78.128.114.174 192.168.0.177 eu-central-1 euc1-az2 - - - - ingress -
9  5 532258145957 eni-00271afc6a81d8acb - - - - - - - 1700590839 1700590870 - NODATA vpc-02
       bdfd80211d2b1a9 subnet-08de94822cb1931c6 - - - - - eu-central-1 euc1-az2 - - - - - -
10 5 532258145957 eni-00271afc6a81d8acb - - - - - - - 1700590868 1700590868 - NODATA vpc-02
       bdfd80211d2b1a9 subnet-08de94822cb1931c6 - - - - - eu-central-1 euc1-az2 - - - - - -
11 5 532258145957 eni-00271afc6a81d8acb - - - - - - - 1700590868 1700590868 - SKIPDATA vpc
       -02bdfd80211d2b1a9 subnet-08de94822cb1931c6 - - - - - eu-central-1 euc1-az2 - - - -
       - -
```

A.2 CloudTrail Event Beispiel

Quelltext A.2: Example of a CloudTrail Event - Terminate EC2 Instance


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAXX3IOY2SY46KYQMBU",
        "arn": "arn:aws:iam::532258145957:user/chris",
        "accountId": "532258145957",
        "accessKeyId": "ASIAXX3IOY2SY7G4D2AK",
        "userName": "chris",
        "sessionContext": {
            "sessionIssuer": {},
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-11-24T22:07:22Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2023-11-24T22:17:59Z",
    "eventSource": "ec2.amazonaws.com",
    "eventName": "TerminateInstances",
    "awsRegion": "eu-central-1",
    "sourceIPAddress": "<ENTFERNT>",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "instancesSet": {
            "items": [
                {
                    "instanceId": "i-036b4eb40579de5d3"
                }
            ]
        }
    },
    "responseElements": {
        "requestId": "64b1d07f-975f-48bc-ad17-af12d665fe30",
        "instancesSet": {
```

```
            "items": [
                {
                    "instanceId": "i-036b4eb40579de5d3",
                    "currentState": {
                        "code": 32,
                        "name": "shutting-down"
                    },
                    "previousState": {
                        "code": 16,
                        "name": "running"
                    }
                }
            ]
        }
    },
    "requestID": "64b1d07f-975f-48bc-ad17-af12d665fe30",
    "eventID": "f897cc70-4336-4d3a-9044-ca2f877b6655",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "532258145957",
    "eventCategory": "Management",
    "sessionCredentialFromConsole": "true"
}
```

A.3 GuardDuty Finding Beispiel

Quelltext A.3: Example of a GuardDuty Finding for Malicious Caller IP

```
[
    {
        "AccountId": "123456789123",
        "Arn":
           ↪ "arn:aws:guardduty:eu-west-1:532258145957:detector/
           ↪ 76c62318472cc10876385d1106aeae56/finding/e73b1b85f02a4bdc9f11be8
        "CreatedAt": "2023-12-07T21:34:35.709Z",
        "Description": "API GeneratedFindingAPIName was invoked
           ↪ from a malicious IP address 198.51.100.0.",
        "Id": "e73b1b85f02a4bdc9f11be8730afaddc",
        "Partition": "aws",
        "Region": "eu-west-1",
        "Resource": {
            "AccessKeyDetails": {
                "AccessKeyId": "GeneratedFindingAccessKeyId",
                "PrincipalId": "GeneratedFindingPrincipalId",
                "UserName": "GeneratedFindingUserName",
                "UserType": "IAMUser"
            },
            "InstanceDetails": {
                "AvailabilityZone":
                   ↪ "GeneratedFindingInstaceAvailabilityZone",
                "IamInstanceProfile": {
                    "Arn": "arn:aws:iam::532258145957:example/
                       ↪ instance/profile",
                    "Id": "GeneratedFindingInstanceProfileId"
                },
                "ImageDescription":
                   ↪ "GeneratedFindingInstaceImageDescription",
                "ImageId": "ami-99999999",
                "InstanceId": "i-99999999",
                "InstanceState": "running",
                "InstanceType": "m3.xlarge",
                "OutpostArn":
                   ↪ "arn:aws:outposts:us-west-2:123456789000:
                   ↪ outpost/op-0fbc006e9abbc73c3",
```

```
                "LaunchTime": "2016-08-02T02:05:06.000Z",
                "NetworkInterfaces": [
                    {
                        "Ipv6Addresses": [],
                        "NetworkInterfaceId": "eni-bfcffe88",
                        "PrivateDnsName":
                            ↪ "GeneratedFindingPrivateDnsName",
                        "PrivateIpAddress": "10.0.0.1",
                        "PrivateIpAddresses": [
                            {
                                "PrivateDnsName":
                                    "GeneratedFindingPrivateName",
                                "PrivateIpAddress": "10.0.0.1"
                            }
                        ],
                        "PublicDnsName":
                            ↪ "GeneratedFindingPublicDNSName",
                        "PublicIp": "198.51.100.0",
                        "SubnetId": "GeneratedFindingSubnetId",
                        "VpcId": "GeneratedFindingVPCId"
                    }
                ],
                "Platform": null,
                "ProductCodes": [
                    {
                        "Code": "GeneratedFindingProductCodeId",
                        "ProductType":
                            ↪ "GeneratedFindingProductCodeType"
                    }
                ],
                "Tags": [
                    {
                        "Key": "GeneratedFindingInstaceTag1",
                        "Value": "GeneratedFindingInstaceValue1"
                    }
                ]
            },
            "ResourceType": "AccessKey"
        },
```

```
        "SchemaVersion": "2.0",
        "Service": {
            "Action": {
                "ActionType": "AWS_API_CALL",
                "AwsApiCallAction": {
                    "Api": "GeneratedFindingAPIName",
                    "CallerType": "Remote IP",
                    "ErrorCode": "AccessDenied",
                    "RemoteIpDetails": {
                        "City": {
                            "CityName":
                                ↪ "GeneratedFindingCityName"
                        },
                        "Country": {
                            "CountryName":
                                ↪ "GeneratedFindingCountryName"
                        },
                        "GeoLocation": {
                            "Lat": 0,
                            "Lon": 0
                        },
                        "IpAddressV4": "198.51.100.0",
                        "Organization": {
                            "Asn": "-1",
                            "AsnOrg": "GeneratedFindingASNOrg",
                            "Isp": "GeneratedFindingISP",
                            "Org": "GeneratedFindingORG"
                        }
                    },
                    "ServiceName":
                        ↪ "GeneratedFindingAPIServiceName",
                    "AffectedResources": {}
                }
            },
            "Evidence": {
                "ThreatIntelligenceDetails": [
                    {
                        "ThreatListName":
                            ↪ "GeneratedFindingThreatListName",
```

```
                        "ThreatNames": [
                            "GeneratedFindingThreatName"
                        ]
                    }
                ]
            },
            "Archived": false,
            "Count": 1,
            "DetectorId": "76c62318472cc10876385d1106aeae56",
            "EventFirstSeen": "2023-12-07T21:34:35.000Z",
            "EventLastSeen": "2023-12-07T21:34:35.000Z",
            "ResourceRole": "TARGET",
            "ServiceName": "guardduty",
            "AdditionalInfo": {
                "Value": "{\"apiCalls\":[{\"name\":\"Generated-
                    ↪ FindingAPIName1\",\"count\":18,\"firstSeen
                    ↪ \":1512692639,\"lastSeen\":1512692839}],
                    ↪ \"sample\":true}",
                "Type": "default"
            }
        },
        "Severity": 5,
        "Title": "API GeneratedFindingAPIName was invoked from
            ↪ a known malicious IP address.",
        "Type": "UnauthorizedAccess:IAMUser/MaliciousIPCaller",
        "UpdatedAt": "2023-12-07T21:34:35.709Z"
    }
]
```

A.4 CloudTrail Data Event Beispiel

Quelltext A.4: CloudTrail Data Event für das Abrufen eines Objekts in einem S3 Bucket

```
{
    "eventVersion": "1.09",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAXX3IOY2SY46KYQMBU",
        "arn": "arn:aws:iam::532258145957:user/chris",
        "accountId": "532258145957",
        "accessKeyId": "ASIAXX3IOY2STJW4IO5K",
        "userName": "chris",
        "sessionContext": {
            "attributes": {
                "creationDate": "2023-12-11T09:45:57Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2023-12-11T10:07:09Z",
    "eventSource": "s3.amazonaws.com",
    "eventName": "GetObject",
    "awsRegion": "eu-central-1",
    "sourceIPAddress": "REDACTED",
    "userAgent": "[Mozilla/5.0 (Macintosh; Intel Mac OS X
    ↪ 10_15_7)]",
    "requestParameters": {
        "X-Amz-Date": "20231211T100711Z",
        "bucketName": "thesis-ir-scenario-2",
        "X-Amz-Algorithm": "AWS4-HMAC-SHA256",
        "response-content-disposition": "attachment",
        "X-Amz-SignedHeaders": "host",
        "Host": "thesis-ir-scenario-2.s3.
        ↪ eu-central-1.amazonaws.com",
        "X-Amz-Expires": "300",
        "key": "example.gif"
    },
    "responseElements": null,
    "additionalEventData": {
```

```
        "SignatureVersion": "SigV4",
        "CipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
        "bytesTransferredIn": 0,
        "AuthenticationMethod": "QueryString",
        "x-amz-id-2": "tkE2ka8/gMGFCnL2RTHCRMHzZD1UoVF6byvHBAD",
        "bytesTransferredOut": 836367
    },
    "requestID": "VV02JCYJ2TC9KF7N",
    "eventID": "ebfd7e39-c1ce-44f8-bdc5-d1ff56fa33bf",
    "readOnly": true,
    "resources": [
        {
            "type": "AWS::S3::Object",
            "ARN":
               ↪ "arn:aws:s3:::thesis-ir-scenario-2/example.gif"
        },
        {
            "accountId": "532258145957",
            "type": "AWS::S3::Bucket",
            "ARN": "arn:aws:s3:::thesis-ir-scenario-2"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "532258145957",
    "eventCategory": "Data",
    "tlsDetails": {
        "tlsVersion": "TLSv1.2",
        "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
        "clientProvidedHostHeader": "thesis-ir-scenario-2.
           ↪ s3.eu-central-1.amazonaws.com"
    }
}
```

A.5 S3 Server Access Log Beispiel

Quelltext A.5: S3 Server Access Log für das Abrufen eines Objekts in einem S3 Bucket

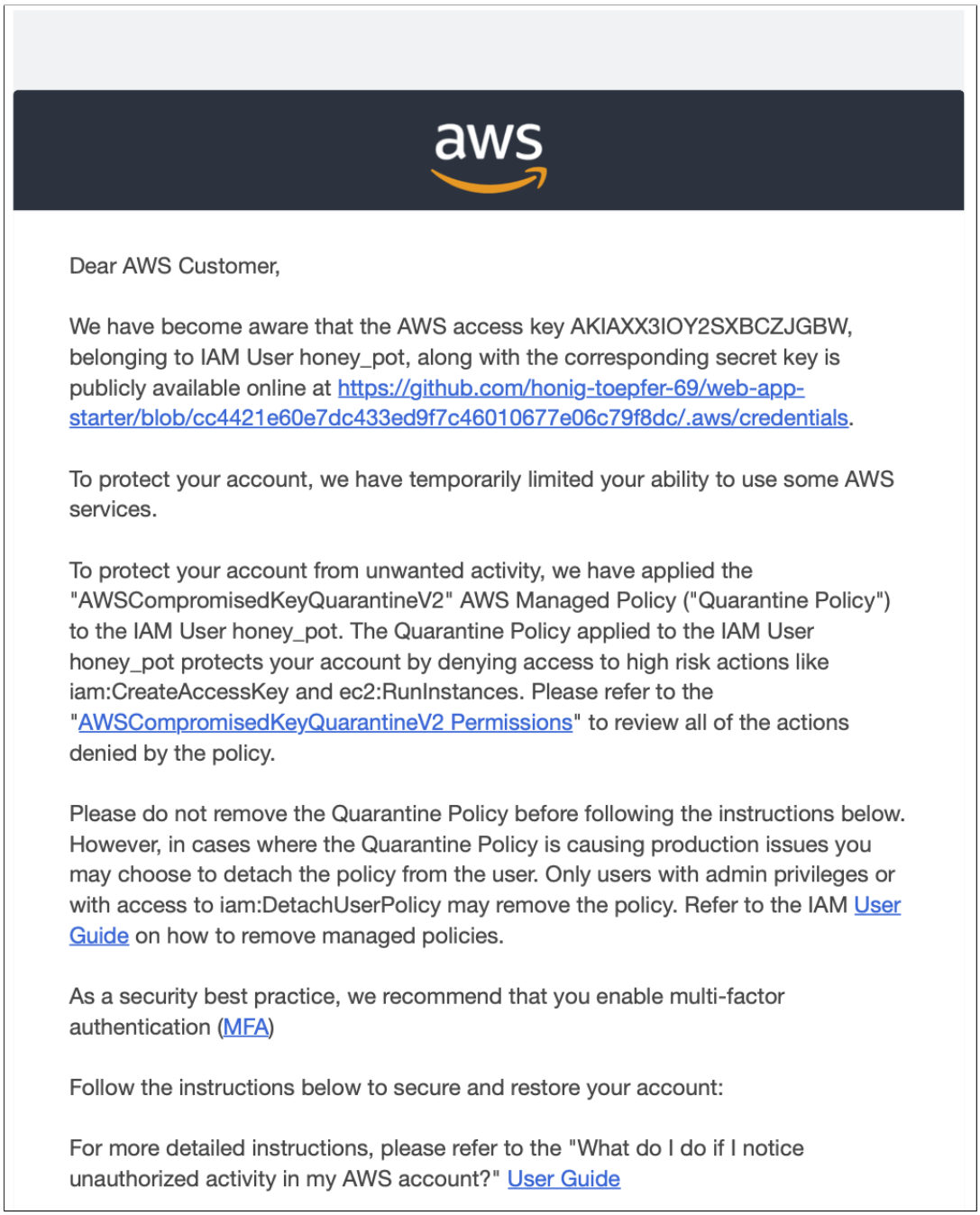
```
1 b4f727103f39ff5886f867180b5a373f7f03b6c1111a335441f188559d57ed54 thesis-ir-scenario-2
      [11/Dec/2023:10:07:09 +0000] 91.3.169.56 arn:aws:iam::532258145957:user/chris
      AJACRSS370F68RSV REST.HEAD.OBJECT example.gif "HEAD /example.gif HTTP/1.1" 200 - -
      836367 25 - "-" "S3Console/0.4, aws-internal/3 aws-sdk-java/1.12.488 Linux
      /5.10.199-167.747.amzn2int.x86_64 OpenJDK_64-Bit_Server_VM/25.372-b08 java/1.8.0_372
       vendor/Oracle_Corporation cfg/retry-mode/standard" - +5
      dWMlAYT92bSFhVWga4pm6PMJZnAhtzMMMCeLQ+4nYxTpwoZ3gx3tBx0UxKdeQ2cCzlMMn/z7BXFXmvvAtepg
      == SigV4 ECDHE-RSA-AES128-GCM-SHA256 AuthHeader thesis-ir-scenario-2.s3.eu-central
      -1.amazonaws.com TLSv1.2 - -
```

A.6 IAM Access Analyzer Finding Beispiel

Quelltext A.6: IAM Access Analyzer Finding für einen öffentlichen S3 Bucket


```
{
    "finding": {
        "id": "0001bff3-67ab-4fbb-954c-470e90f81a50",
        "principal": {
            "AWS": "*"
        },
        "action": [
            "s3:GetObject",
            "s3:ListBucket"
        ],
        "resource": "arn:aws:s3:::thesis-ir-scenario-2",
        "isPublic": true,
        "resourceType": "AWS::S3::Bucket",
        "condition": {},
        "createdAt": "2023-12-11T14:48:56.449000+00:00",
        "analyzedAt": "2023-12-11T14:49:50.618000+00:00",
        "updatedAt": "2023-12-11T14:48:56.449000+00:00",
        "status": "ACTIVE",
        "resourceOwnerAccount": "532258145957",
        "sources": [
            {
                "type": "POLICY"
            }
        ]
    }
}
```

B Benachrichtung über öffentliche Credentials



Dear AWS Customer,

We have become aware that the AWS access key AKIAXX3IOY2SXBCZJGBW, belonging to IAM User honey_pot, along with the corresponding secret key is publicly available online at https://github.com/honig-toepfer-69/web-app-starter/blob/cc4421e60e7dc433ed9f7c46010677e06c79f8dc/.aws/credentials.

To protect your account, we have temporarily limited your ability to use some AWS services.

To protect your account from unwanted activity, we have applied the "AWSCompromisedKeyQuarantineV2" AWS Managed Policy ("Quarantine Policy") to the IAM User honey_pot. The Quarantine Policy applied to the IAM User honey_pot protects your account by denying access to high risk actions like iam:CreateAccessKey and ec2:RunInstances. Please refer to the "AWSCompromisedKeyQuarantineV2 Permissions" to review all of the actions denied by the policy.

Please do not remove the Quarantine Policy before following the instructions below. However, in cases where the Quarantine Policy is causing production issues you may choose to detach the policy from the user. Only users with admin privileges or with access to iam:DetachUserPolicy may remove the policy. Refer to the IAM User Guide on how to remove managed policies.

As a security best practice, we recommend that you enable multi-factor authentication (MFA)

Follow the instructions below to secure and restore your account:

For more detailed instructions, please refer to the "What do I do if I notice unauthorized activity in my AWS account?" User Guide

Abbildung 15: Auszug einer Benachrichtungsmail über öffentliche Credentials

C Quarantäne Policy

Quelltext C.1: Auflistung der Policy „AWSCompromisedKeyQuarantineV2“

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "cloudtrail:LookupEvents",
                "ec2:RequestSpotInstances",
                "ec2:RunInstances",
                "ec2:StartInstances",
                "iam:AddUserToGroup",
                "iam:AttachGroupPolicy",
                "iam:AttachRolePolicy",
                "iam:AttachUserPolicy",
                "iam:ChangePassword",
                "iam:CreateAccessKey",
                "iam:CreateInstanceProfile",
                "iam:CreateLoginProfile",
                "iam:CreatePolicyVersion",
                "iam:CreateRole",
                "iam:CreateUser",
                "iam:DetachUserPolicy",
                "iam:PassRole",
                "iam:PutGroupPolicy",
                "iam:PutRolePolicy",
                "iam:PutUserPermissionsBoundary",
                "iam:PutUserPolicy",
                "iam:SetDefaultPolicyVersion",
                "iam:UpdateAccessKey",
                "iam:UpdateAccountPasswordPolicy",
                "iam:UpdateAssumeRolePolicy",
```

```
32                  "iam:UpdateLoginProfile",
33                  "iam:UpdateUser",
34                  "lambda:AddLayerVersionPermission",
35                  "lambda:AddPermission",
36                  "lambda:CreateFunction",
37                  "lambda:GetPolicy",
38                  "lambda:ListTags",
39                  "lambda:PutProvisionedConcurrencyConfig",
40                  "lambda:TagResource",
41                  "lambda:UntagResource",
42                  "lambda:UpdateFunctionCode",
43                  "lightsail:Create*",
44                  "lightsail:Delete*",
45                  "lightsail:DownloadDefaultKeyPair",
46                  "lightsail:GetInstanceAccessDetails",
47                  "lightsail:Start*",
48                  "lightsail:Update*",
49                  "organizations:CreateAccount",
50                  "organizations:CreateOrganization",
51                  "organizations:InviteAccountToOrganization",
52                  "s3:DeleteBucket",
53                  "s3:DeleteObject",
54                  "s3:DeleteObjectVersion",
55                  "s3:PutLifecycleConfiguration",
56                  "s3:PutBucketAcl",
57                  "s3:PutBucketOwnershipControls",
58                  "s3:DeleteBucketPolicy",
59                  "s3:ObjectOwnerOverrideToBucketOwner",
60                  "s3:PutAccountPublicAccessBlock",
61                  "s3:PutBucketPolicy",
62                  "s3:ListAllMyBuckets",
63                  "ec2:PurchaseReservedInstancesOffering",
64                  "ec2:AcceptReservedInstancesExchangeQuote",
65                  "ec2:CreateReservedInstancesListing",
66                  "savingsplans:CreateSavingsPlan"
67              ],
68              "Resource": [
69                  "*"
70              ]
```

```
71              }
72          ]
73  }
```

D Beispiele für Athena Queries

D.1 Athena Queries für CloudTrail Data Events

Quelltext D.1: CloudTrail Data Events: Welche IAM Entität hat API Aufrufe gegen den verdächtigten S3 Bucket durchgeführt?

```
SELECT
    userIdentity.arn as userArn,
    eventTime,
    eventName,
    json_extract_scalar(requestParameters, '$.bucketName') as bucketName,
    json_extract_scalar(requestParameters, '$.key') as object,
FROM
    '<table-name>'
WHERE
    bucketName = '<bucket-name>'
    AND eventTime BETWEEN '<start-date>' and '<end-date>'
```

Quelltext D.2: CloudTrail Data Events: Von welcher IP Adresse und mit welchem User Agent wurden diese API Aufrufe durchgeführt?

```
SELECT
    sourceIpAddress,
    userAgent,
    eventTime,
    eventName,
    json_extract_scalar(requestParameters, '$.bucketName') as bucketName,
    json_extract_scalar(requestParameters, '$.key') as object,
FROM
    '<table-name>'
WHERE
    bucketName = '<bucket-name>'
    AND eventTime BETWEEN '<start-date>' AND '<end-date>'
```

Quelltext D.3: CloudTrail Data Events: Wurden Objekte aus einem S3 Bucket exilftriert?

```
SELECT
    *,
    json_extract_scalar(requestParameters, '$.bucketName') as bucketName,
    json_extract_scalar(requestParameters, '$.key') as object,
FROM
    '<table-name>'
WHERE
    eventName = 'GetObject'
    AND bucketName = '<bucket-name>'
    AND eventTime BETWEEN '<start-date>' AND '<end-date>'
```

Quelltext D.4: CloudTrail Data Events: Wurden Objekte aus einem S3 Bucket gelöscht?

```
SELECT
    *,
    json_extract_scalar(requestParameters, '$.bucketName') as bucketName,
    json_extract_scalar(requestParameters, '$.key') as object,
FROM
    '<table-name>'
WHERE
    eventName = 'DeleteObject'
    AND bucketName = '<bucket-name>'
    AND eventTime BETWEEN '<start-date>' AND '<end-date>'
```

Quelltext D.5: CloudTrail Data Events: Wurden neue Objekte einem S3 Bucket hinzugefügt?

```
SELECT
    *,
    json_extract_scalar(requestParameters, '$.bucketName') as bucketName,
    json_extract_scalar(requestParameters, '$.key') as object,
FROM
    '<table-name>'
WHERE
    eventName = 'PutObject'
    AND bucketName = '<bucket-name>'
    AND eventTime BETWEEN '<start-date>' AND '<end-date>'
```

D.2 Atehana Queries für S3 Server Access Logs

Quelltext D.6: S3 Server Access Logs: Welche IAM Entität hat API Aufrufe gegen den verdächtigten S3 Bucket durchgeführt?

```
SELECT
    requester,
    *
FROM
    '<table-name>'
WHERE
    parse_datetime(requestdatetime,'dd/MMM/yyyy:HH:mm:ss Z') BETWEEN
        parse_datetime('<start-date>','yyyy-MM-dd:HH:mm:ss')
        AND parse_datetime('<end-date>','yyyy-MM-dd:HH:mm:ss')
```

Quelltext D.7: S3 Server Access Logs: Von welcher IP Adresse und mit welchem User Agent wurden diese API Aufrufe durchgeführt?

```
SELECT
    remoteip,
    useragent,
    *
FROM
    '<table-name>'
WHERE
    parse_datetime(requestdatetime,'dd/MMM/yyyy:HH:mm:ss Z') BETWEEN
        parse_datetime('<start-date>','yyyy-MM-dd:HH:mm:ss')
        AND parse_datetime('<end-date>','yyyy-MM-dd:HH:mm:ss')
```

Quelltext D.8: S3 Server Access Logs: Wurden Daten aus einem S3 Bucket exfiltriert?

```
SELECT
    *
FROM
    '<table-name>'
WHERE
    operation='REST.GET.OBJECT'
    AND parse_datetime(requestdatetime,'dd/MMM/yyyy:HH:mm:ss Z') BETWEEN
        parse_datetime('<start-date>','yyyy-MM-dd:HH:mm:ss')
        AND parse_datetime('<end-date>','yyyy-MM-dd:HH:mm:ss')
```

Quelltext D.9: S3 Server Access Logs: Wurden Objekte aus einem S3 Bucket gelöscht?

```
SELECT
    *
FROM
    '<table-name>'
WHERE
    operation='REST.DELETE.OBJECT'
    AND parse_datetime(requestdatetime,'dd/MMM/yyyy:HH:mm:ss Z') BETWEEN
        parse_datetime('<start-date>','yyyy-MM-dd:HH:mm:ss')
        AND parse_datetime('<end-date>','yyyy-MM-dd:HH:mm:ss')
```

Quelltext D.10: S3 Server Access Logs: Wurden neue Objekte einem S3 Bucket hinzugefügt?

```
SELECT
    *
FROM
    '<table-name>'
WHERE
    operation='REST.PUT.OBJECT'
    AND parse_datetime(requestdatetime,'dd/MMM/yyyy:HH:mm:ss Z') BETWEEN
        parse_datetime('<start-date>','yyyy-MM-dd:HH:mm:ss')
        AND parse_datetime('<end-date>','yyyy-MM-dd:HH:mm:ss')
```

E Beispiele für Bucket Policies

E.1 Bucket Policies mit Whitelisting Ansatz

Quelltext E.1: Bucket Policy mit Whitelisting eines IP Adressbereichs

```
{
    "Version": "2012-10-17",
    "Id": "Whitelist-IP",
    "Statement": [
        {
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:*",
            "Resource": "*",
            "Condition": {
                "NotIpAddress": {
                    "aws:SourceIp": "<WHITELISTED_IP_ADDRESS>"
                }
            }
        }
    ]
}
```

Quelltext E.2: Bucket Policy mit Whitelisting bestimmter Entitäten

```
{
    "Version": "2012-10-17",
    "Id": "Whitelist-IAM-Entity",
    "Statement": [
        {
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:*",
            "Resource": "*",
```

```
10              "Condition": {
11                  "ArnNotEquals": {
12                      "aws:SourceArn": "<WHITELISTED_ENTITIES>"
13                  }
14              }
15          }
16      ]
17  }
```

E.2 Bucket Policies mit Blacklisting Ansatz

Quelltext E.3: Bucket Policy mit Blacklisting eines IP Adressbereichs

```
{
    "Version": "2012-10-17",
    "Id": "Blacklist-IP",
    "Statement": [
        {
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:*",
            "Resource": "*",
            "Condition": {
                "IpAddress": {
                    "aws:SourceIp": "<BLACKLISTED_IP_ADDRESS>"
                }
            }
        }
    ]
}
```

Quelltext E.4: Bucket Policy mit Blacklisting bestimmter Entitäten

```
{
    "Version": "2012-10-17",
    "Id": "Blacklist-IAM-Entity",
    "Statement": [
        {
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:*",
            "Resource": "*",
            "Condition": {
                "ArnEquals": {
                    "aws:SourceArn": "<BLACKLISTED_ENTITIES>"
                }
            }
        }
    ]
}
```