Bericht zum Forschungsseminar „Cloudaufklärung ohne das Verletzen von Sicherheitsmaßnahmen“

Durchgeführt von: **Tom Marinovic**

Betreut durch: Professor Dr. Westfeld

15. Februar 2024

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

1.1 Motivation

Informationssicherheit ist ein Thema, welches in den letzten Jahren zunehmend an Bedeutung gewonnen hat. Insbesondere der kontinuierliche Anstieg der Kommerzialisierung von Angriffen auf Informationssysteme (im Folgenden IT-Systeme) hat das Thema der Informationssicherheit zunhemend in den Fokus der Öffentlichkeit gerückt. Der erste Schritt bei einem jeden Angriff auf IT-Systeme ist die Aufklärung (englisch: reconnaissance), mit dem Ziel, Informationen über das Angriffsziel zu erlangen. Die Art der gesuchten Daten kann dabei sehr unterschiedlich sein. Ein Angreifer kann nach spezifischen Sicherheitslücken suchen oder sich einen generellen Überblick über die Zielinfrastruktur verschaffen wollen, um somit einen gezielteren Angriff zu ermöglichen [4, S. 3]. Neue Entwicklungen im Bereich der IT-Systeminfrastruktur stellen somit neue Ziele für die Aufklärungsversuche von Angreifern dar. Einer dieser neuen Bereiche sind Cloud-Infrastrukturen, dabei handelt es sich um eine Cloud bsierte Infrastruktur, welche einem Kunden gegen ein Monatliches Entgeld zur Verfühgung gestellt wird. Der Begriff Cloud wird im Zuge diesre Arbeit wie folgt definiert: „A computing Cloud is a set of network enabled services, providing scalable, QoS guaranteed, normally personalized, inexpensive computing infrastructures on demand, which could be accessed in a simple and pervasive way.“ [6, S. 139]. Inhaltlich übersetzt bedeutet dies: Cloud Computing ist eine netzwerkbasierte Sammlung von variablen Diensten mit garantierten Eigenschaften, die typischerweise personalisiert und dynamisch skalierbar gegen ein Entgelt abgerufen werden können. Der Abruf erfolgt meistens über das Internet. Eine genauere Betrachtung der Eigenschaften von Cloud-Infrastrukturen erfolgt im Kapitel "Vergleich klassische und Cloud-Infrastruktur".
Aufgrund dieser Gegebenheiten war es von Interesse, einen genaueren Blick auf die Möglichkeiten zur Aufklärung von Cloud-Infrastrukturen zu erhalten. Im Rahmen eines Forschungsseminars sollten Möglichkeiten diesbezüglich anhand praktischer Beispiele überprüft werden. Zudem sollten diese Versuche ohne Verletzung von Sicherheitsmaßnahmen stattfinden. Um diese Aufgabenstellung bearbeiten zu können, musste zunächst eine Forschungsfrage definiert werden, die das Themengebiet angemessen einschränkt. Anschließend sollten zunächst einige theoretische Grundlagen erläutert werden, um im Folgenden anhand praktischer Untersuchungen die Annahmen aus der theoretischen Betrachtung zu überprüfen.

1.2 Forschungsfrage

Wie in der Motivation bereits beschrieben, war das Ziel die Ermittlung von Möglichkeiten zur Aufklärung von Cloudinfrastrukturen, ohne dass Sicherheitsmaßnahmen verletzt werden. Diese Einschränkung ist nötig, da keine Programmierfehler oder Social Engineering Techniken untersucht werden sollten. Der Fokus sollte stattdessen darauf liegen, Konfigurationsfehler auszunutzen. Damit dies im gegebenen Zeitrahmen sinnvoll bearbeitet werden konnten, musste sich auf einen Cloud-Infrastrukturanbieter beschränkt werden. Aufgrund unterschiedlicher Kernstrukturen zwischen den einzelnen Anbietern bestand die Annahme, dass eventuell gefundene Konfigurationsfehler nicht plattformunabhängig sind. Zur Auswahl standen die drei größten Marktanbieter: Amazon's AWS, Microsoft's Azure und Alphabet's Google Cloud. Die Marktanteile der einzelnen Anbieter werden in folgender Grafik dargestellt:

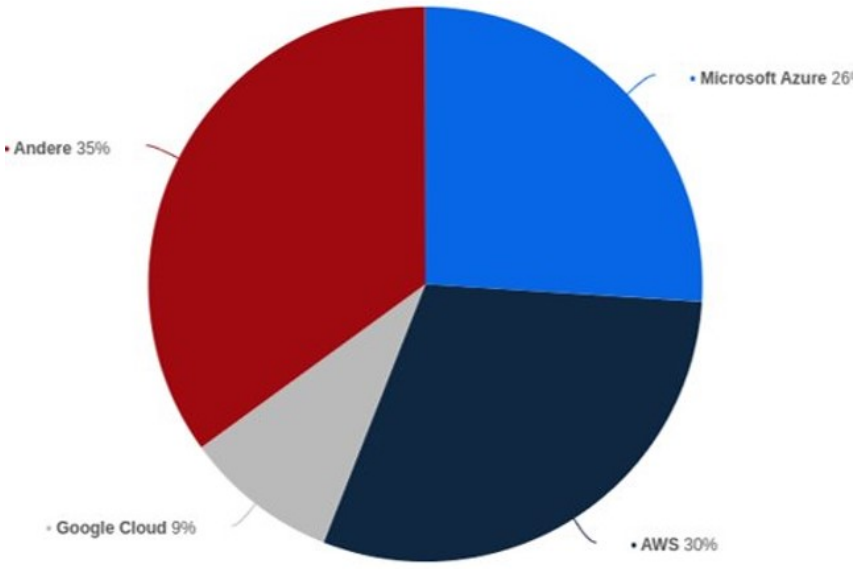

Abbildung 1: Verteilung Marktanteile Cloudanbieter (23.11.23)

Es wurde sich bereits zu Beginn des Projektes entschieden, die Betrachtung auf Microsoft's Azure (im Folgenden nur Azure) zu beschränken, da hier die meisten Vorkenntnisse bestanden. Eine weitere Einschränkung, die es zu beachten galt, ist der Umfang einzelner Cloudlösungen. Aufgrund der Fülle angebotener Optionen war es nicht möglich, sämtliche Funktionalitäten zu betrachten. Stattdessen sollten explorativ verschiedene Möglichkeiten ausprobiert und die vielversprechendsten gezielt untersucht werden. Aus diesen Anforderungen wurde folgende Forschungsfrage abgeleitet: **"Welche Möglichkeiten zur Aufklärung von Azure Cloudinfrastrukturen ermöglichen Konfigurationsfehler, ohne dabei Sicherheitsmaßnahmen zu verletzen?"** Das Vorgehen und die Erkenntnisse im Zuge der Beantwortung dieser Frage werden im Folgenden dargestellt.

2 Theoretische Grundlagen

2.1 Vergleich Klassische und Cloud-Infrastruktur

Initial wurden zunächst Cloud- mit klassischen Infrastrukturen verglichen. Klassisch in diesem Sinne sind Infrastrukturen, welche von Organisationen eigenständig betrieben werden und für deren Erhalt Sie selbst verantwortlich sind. Man spricht auch von einer „In-house“- Architektur, da eine solche Infrastruktur oft in den eigenen Räumlichkeiten einer Organisation untergebracht wird. Die Strukturierung des Vergleichs erfolgte anhand der Kerneigenschaften von Cloud-Architekturen, da die Vermutung bestand das die Unterschiedene hier am stärksten ausgeprägt wären. Diese Vergleichspunkte sind Erreichbarkeit, Größe/Skalierbarkeit, Konfiguration, Verwaltungsinstanz und Updatemanagement. Folgende Grafik fasst die Vergleichsergebnisse zusammen:

Eigenschaft	Klassische Infrastruktur	Cloud Infrastruktur
Erreichbarkeit	beschränkt erreichbar, Abschottung über Firewall und VPN-Zugriffe	weltweit über das Internet erreichbar, Ressourcen werden direkt mit Zugriffsrechten geschützt
Größe & Skalierbarkeit	Festplanung auf mehrjährigen Zeitraum, Puffer wird eingeplant	"Kauf was du brauchst und nicht mehr"-Ansatz sehr flexibel und dynamisch skalierbar
Konfiguration	systematische Planung und einmalige Konfiguration	häufige Anpassung üblich
Verwaltet durch	Administratoren	Administratoren und Endnutzer
Patch-Management	durch interne Verwaltung	durch Cloudanbieter

Abbildung 2: Vergleich Eigenschaften Klassischer und Cloud-Infrastrukturen

Ein besonderes Augenmerk muss auf den Punkt Erreichbarkeit geworfen werden. Cloud-Infrastrukturen sind stets über das Internet erreichbar, weltweit, während eine klassische Infrastruktur üblicherweise abgeschottet wird und nur einzelne Anwendungen von außen erreichbar sind. Dies bedeutet, dass bei Cloud-Lösungen global gezielt nach Fehlern gesucht werden kann, statt unterschiedliche Infrastrukturen testen zu müssen. Ein potenzieller Fehler, ist somit bei allen Kunden der jeweiligen Cloud global möglich. Bestärkend zu diesem Fakt kommt auch hinzu, dass Cloud-Umgebungen nicht nur von ausgebildeten Administratoren, sondern auch von sogenannten Schlüsselnutzern (englisch: Keyuser) verwaltet werden [6, S. 140f]. Das Berechtigungskonzept der Azure Cloud sieht vor, dass Nutzer existieren, welche Änderungen an bestehender Infrastruktur vornehmen und neue anlegen können. Hier bestand die Annahme, dass im Zuge der häufigen Anpassungen in einer Cloud-Umgebung Konfigurationsfehler leichter auftreten können und länger unbemerkt bleiben. Zusätzlich verspricht der homogene Aufbau von Cloud-Infrastrukturen, dass Konfigurationsfehler, falls vorhanden, alle Ressourcen dieser Art für eine Organisation betreffen [4, S. 3]. In einer klassischen Infrastruktur werden diese Risiken abgeschwächt,

da nur ein kleiner Teil der Infrastruktur von außerhalb des Organisationsnetzwerks erreichbar ist. Die dadurch verringerte Angriffsfläche kann auch leichter durch einzelne Personen kontrolliert werden. Der Aufbau einer üblichen Organisationsinfrastruktur sieht zudem vor das nur ausgewähltes Fachpersonal technische Änderungen oder Erweiterungen vornehmen können. Die Möglichkeit von unbeabsichtigten Konfigurationsfehlern wird somit weiter abgeschwächt.
Zusammenfassend lässt sich feststellen, dass der Aufbau von Cloud-Infrastrukturen das Auftreten und den Erhalt von Konfigurationsfehlern im Gegensatz zu klassischen Infrastrukturen begünstigt. Die Suche nach solchen Fehlern sollte sich auf die initiale Konfiguration von Diensten konzentrieren, unter der Annahme, dass hier das Größte Auswirkungspotenziall besteht. In der initialen Konfiguration werden üblicherweise Rahmenbedingungen für einen Service definiert, welche während des Betriebes selten angepasst werden müssen. Zudem lässt der weitverbreitete und gern gelebte Spruch „Never touch a running system“, frei übersetzt „Verändere niemals ein funktionsfähiges IT-System“, darauf Vermuten, dass initiale Fehler am langlebigsten sind.

2.2 Angriffsmöglichkeiten für Cloud-Infrastrukturen

Das Aufstellen der Unterschiede zwischen Cloud- und klassischen Infrastrukturen wirft auch die Frage nach generellen Angriffsmöglichkeiten auf, insbesondere welche Möglichkeiten ein Angreifer aufgrund der Eigenheiten einer Cloud hat, die in klassischen Infrastrukturen nicht gegeben sind. Auch wenn es nicht Teil der Forschungsfrage ist, verdient die Angriffstaktik des Social Engineerings (im Folgenden SE) besondere Erwähnung im Zusammenhang mit Cloud-Infrastrukturen. Social Engineering wird definiert als: „Social engineering is any act that influences a person to take an action that may or may not be in his or her best interests.“ [3, S. 7], frei übersetzt als „Social Engineering ist eine Technik, eine Person dazu zu beeinflussen, eine Handlung auszuführen, die möglicherweise nicht in ihrem Interesse liegt.“ SE, alleinstehend oder als Teil einer Angriffstechnik, stellt eine signifikante Bedrohung für sämtliche Arten von Infrastrukturen dar, aber besonders für Cloud-Umgebungen wird diese Bedrohung verstärkt. Durch die globale Erreichbarkeit und den homogenen Aufbau einer Cloud können einem Angreifer bereits wenige Informationen genügen, um eine Cloud-Infrastruktur teilweise oder ganzheitlich zu kompromittieren. Ein gestohlenes Nutzerpasswort stellt in einer klassischen Infrastruktur eine grobe Sicherheitsverletzung dar, aber es bestehen gute Chancen, dass der Angreifer auf seinem ersten Erfolg nicht aufbauen kann, da ihm der Zugriff auf die eigentliche Infrastruktur verwehrt bleibt. Durch die starke Abschottung sind die möglichen Zugänge zur Infrastruktur beschränkt und oft stark überwacht. Hingegen kann ein Angreifer in einer Cloud-Umgebung oft sofort und ohne Umwege gestohlene Anmeldeinformationen verwenden, ohne weitere Sicherheitsmaßnahmen überwinden zu müssen.

Aufgrund der dauerhaften Verfügbarkeit bedürfen eingesetzte Clouddienste aktiver Überwachung, was technisch und personell sehr aufwendig ist. Daher kann davon ausgegangen werden, dass nicht alle Cloud-Infrastrukturen aktiv gepflegt werden. Die Erreichbarkeit der Cloud bringt bedingt jedoch noch weitere rein technische Aufklärungsmöglichkeiten für einen Angreifer. Da ein Großteil der Dienste in der Azure Cloud weltweit erreichbar ist und die Option bietet, dass sie ohne Einschränkungen aufgerufen werden können, bedarf es einer Vielzahl an einzigartigen Namen. Um die Zuordnung einer Ressource zu ermöglichen, bedarf es in so gut wie allen technischen Systemen eines eindeutigen Identifikationsmerkmals. Azure ist insofern besonders, da diese Identifikationsmerkmale nicht nur, wie oft üblich, numerisch sind, sondern abhängig von der Ressource auch aus Buchstaben bestehen können. Diese Eigenheit könnte es erlauben, gezielt nach Ressourcen zu suchen und diese leichter einem Ziel zuzuordnen. Auch lassen sich mittels sogenannter Wörterbuchattacken gezielt nach Begriffen suchen, unter der Annahme, dass Ressourcen möglichst aussagekräftige Namen bekommen. Unter Beachtung dieser Eigenschaften wurden verschiedene Dienste von Azure untersucht, ob es sich bei ihnen um günstige Ziele handeln könnte. Ein Service, der dabei besonders hervorstach, waren die Azure Storage Accounts (zu deutsch Speicherkonten).

2.3 Aufbau Azure Storage

Ein Azure Speicherkonto fungiert als Container für verschiedene Arten von Speicherdatenobjekten und ermöglicht den Zugriff auf diese Daten über HTTP oder HTTPS. Die Erreichbarkeit erfolgt über einen eindeutigen Namen, und es stehen verschiedene Optionen zur Verfügung, um Verfügbarkeit und Änderungsresilienz anzupassen [5]. Innerhalb eines Speicherkontos werden verschiedene Arten von Datenobjekten gespeichert, die in 4 Kategorien unterteilt werden können: Blob Storage, Azure Files, Queue Storage und Table Storage, die gemeinsam auch als Storage Container bezeichnet werden. Diese Datenobjekte unterscheiden sich in ihrer Funktion und der Art und Weise, wie Daten abgelegt werden, aber der Zugriff erfolgt bei allen über den eindeutigen Namen des Speicherkontos. Von besonderem Interesse sind Blob Storage und Azure Files, die als Datenablage konzipiert sind. Blob Storage ist für unstrukturierte Daten gedacht, während Azure Files eine Cloud-Variante des Windows-Dateisystems darstellt. Blob Storage wird von Microsoft als generelle Datenablage empfohlen und ist kostengünstig im Verhältnis zur bereitgestellten Speicherkapazität. Zusätzlich wird dieser Dienst von vielen anderen Azure-Services als Speicherablage verwendet und ist somit ein integraler Bestandteil der meisten Azure-Cloud-Infrastrukturen. Die Erreichbarkeit der verschiedenen Datenobjekte erfolgt über den eindeutigen Namen des Speicherkontos, der weltweit eindeutig ist und einem festen Schema von 3-24 Kleinbuchstaben und Zahlen folgen muss [5, Speicherkontoname]. Diese Eigenschaften machen Azure Speicherkonten zu einem vielversprechenden Ziel bei der

Suche nach potenziellen Konfigurationsfehlern. Der Zugriff auf die Datenobjekte erfolgt über den Namen des Speicherkontos, gefolgt vom Pfad zur jeweiligen Ressource, wobei der Name des Speichercontainers keinen Einschränkungen unterliegt. Es ist möglich, diesen Zugriff mittels verschiedener Methoden abzusichern, dies ist jedoch nicht verpflichtend. Eine Standardoption bei der Erstellung eines Speicherkontos ist es, anonymen Zugriff zu erlauben, wodurch nur die URL zum Datenobjekt für den Abruf bekannt sein muss. Des Weiteren besteht die Möglichkeit, die Verbindung über Zugangsschlüssel zu erlauben oder das Azure-eigene rollenbasierte Berechtigungssystem namens Azure Entra ID zu verwenden. Die bei der Erstellung des Speicherkontos gewählte Freigabemethode wird auf alle Datenobjekte in diesem Container vererbt und kann im Nachhinein angepasst werden.

3 Entwicklung

3.1 Entwicklung des Scanners

Durch die identifizierten Eigenschaften von Azure Storage Accounts wurden diese als günstiges Ziel identifiziert, um zu überprüfen, ob mögliche Konfigurationsfehler systematisch ausgenutzt werden können. Da der anonyme Zugriff auf Datenobjekte bei der Erstellung eines Storage Accounts als Standardwert hinterlegt wird, bestand die Annahme, dass einige Nutzer diese Option unbeabsichtigt aktiviert haben und somit Daten für den Abruf bereitstellen, die nicht für den öffentlichen Zugang bestimmt sind. Auch ist das Szenario denkbar, dass bei mehreren Storage Accounts Daten fälschlicherweise in einen öffentlich zugänglichen Account abgelegt wurden. Um diese Annahme zu überprüfen, sollte ein Scanner entwickelt werden, der nach öffentlich verfügbaren Storage Containern sucht und deren Inhalte als URL-Liste ausgibt. Anschließend könnten diese überprüft und ausgewertet werden, um zu überprüfen, inwieweit nicht öffentliche Daten über diese Container abrufbar sind. Es wurde entschieden, den Scanner mittels PowerShell zu entwickeln, da Microsoft dedizierte Plugins für den Zugriff auf Azure-Ressourcen mittels PowerShell bereitstellt.
Im ersten Schritt muss der Scanner nach möglichen Storage Accounts suchen, um diese anschließend auf mögliche offene Storage Container zu überprüfen. Da Storage Accounts einen weltweit eindeutigen Namen haben müssen, muss nur überprüft werden, ob dieser bereits vergeben ist, um anschließend den Account nach Containern zu durchsuchen. Hier entsteht jedoch die erste Schwierigkeit, denn die Namen der Storage Container sind frei wählbar, müssen keinem Schema entsprechen und können sich wiederholen. Somit ist für das Überprüfen von Storage Containern ein Brute-Force-Angriff notwendig, um alle möglichen Container zu überprüfen. Dies wäre jedoch kein zielführendes Vorgehen, daher wird jeder Storage Account nur auf eine Handvoll ausgewählter Container überprüft. Bei diesen handelt es sich um industrietypische Namen für Container, in der Annahme, mit diesen die höchste Trefferquote zu erzielen. Ebenso werden nur ausgewählte Storage Accounts untersucht, da die Anzahl der möglichen Kombinationen $\sum_{i=3}^{24} 34^i$ beträgt und damit zu umfangreich für eine vollständige Überprüfung ist. Somit bleibt die Ausführungszeit des Scanners auch in einem angemessenen Maß.
Der Scanner besteht somit aus zwei separaten Schleifen, die nacheinander erst die Liste ausgewählter Storage Accounts auf ihre Existenz und anschließend für die vorhandenen Accounts auf die Verfügbarkeit ausgewählter Storage Container prüfen. Zum Überprüfen der Accounts kann ein HTTP-Aufruf an den jeweiligen Accountnamen gesendet werden, welcher einen Statuscode von 200 enthält, wenn der Storage Account existiert. Dasselbe Prinzip kann für die Container angewandt werden, wobei der HTTP-Aufruf zusätzlich als Rückgabewert die Ordnerstruktur des Containers enthält, falls dieser existiert. Dadurch

können sämtliche Dateien innerhalb eines Containers abgefragt werden, indem nur das Wurzelverzeichnis korrekt geraten wird. Folgende Grafik zeigt den Aufbau der URLs für Storage Accounts und Container, wie diese durch den Scanner aufgerufen werden.

Abbildung 3: Aufbau Storage Account URL

Auffallend hierbei ist der Teil „blob.core.windows.net“. Dieser ist ein fester Bestandteil der URL, der unabhängig vom Ziel gleich bleibt und durch Azures Infrastruktur vorgegeben wird. Der Storage Container wird gefolgt vom Pfad zur Datei, wobei die Wurzel des Pfades der Containername ist. Der Quellcode für den Scanner sowie die Liste für die Storage Container Namen werden auf GitHub in einem öffentlichen Repository zur Verfügung gestellt [1].

3.2 Auswahl der Ziele

Für den entwickelten Scanner mussten nun Ziele für die zwei Ziellisten bestimmt werden. Die Liste der zu prüfenden Storage-Accounts musste umfangreich sein, um ein möglichst großes Suchspektrum zu haben, während gleichzeitig die Ausführungszeit auf einen praktikablen Rahmen beschränkt werden sollte. Es wurde angenommen, dass Nutzer tendenziell sinnvolle Namen für ihre Storage-Accounts wählen. Daher boten sich bereits existierende Abkürzungen als Ziele an. Es wurde sich hierfür auf deutsche Aktienkennzeichen entschieden, die drei bis vier Buchstaben lang sind und bei denen davon ausgegangen werden kann, dass die zugehörigen Unternehmen Cloud-Infrastrukturen einsetzen. Für diese Menge an potenziellen Account-Namen bestand somit eine erhöhte Chance, dass diese in Verwendung sind. Diese Ziele wurden in Form einer Textdatei für den Scanner bereitgestellt. Die Liste der zu überprüfenden Container wurde ebenfalls in Form einer Textdatei übergeben, die einige typische Ordnernamen enthielt. Insgesamt wurden somit 5540 Accounts auf 83 Container überprüft.

3.3 Kategorisierung der Ergebnisse

Mit der Anwendung des Scanners auf das gestellte Suchset mussten die gefunden Ergebnisse ausgewertet werden. Von den 5540 überprüften Storage Accounts sind insgesamt 1727 tatsächlich vorhanden und somit von Organisationen in Benutzung. Diese bestätigten Storage Accounts wurden anschließend auf öffentlich verfügbare Storage Container überprüft. Dabei konnten rund 30800 öffentlich verfügbare Dateien gefunden werden, welche auf ungefähr 40 Storage Accounts verteilt sind. Diese Fülle an gefunden Informationen musste Kategorisiert werden um eine sinnvolle Auswertung zu ermöglichen. Zu diesem Zweck wurde zunächst eine Liste aller Dateiendungen mit Ihrer Häufigkeit erstellt. Da jede gefundene URL auf genau eine Datei verweist konnte so die Zusammensetzung der Datenmenge erkannt werden, gleichzeitig konnten Dateien identifiziert werden, welche unüblich für den öffentlichen Zugriff sind. Bei dieser ersten Betrachtung wurden 54 verschiedene Dateitypen gefunden, wobei der Großteil der Dateien Bilder im jpg oder png Format waren. Folgende Grafik zeigt die Verteilung der häufigsten Dateitypen, wobei nur die ersten 6 gelistet sind, da viele Typen nur einmalig auftreten.

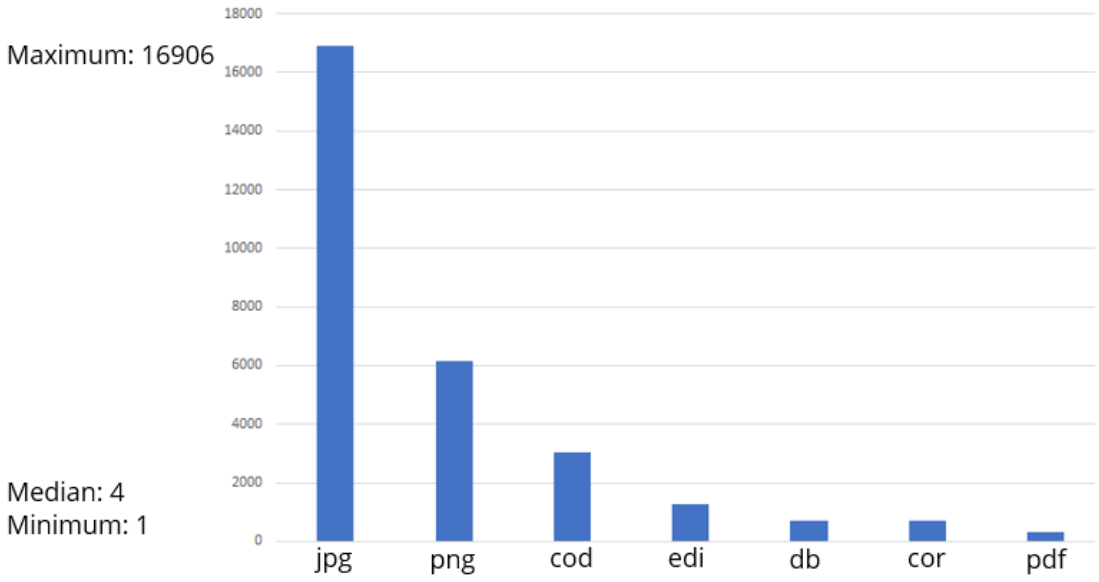

Abbildung 4: Verteilung der 6 am häufigsten gefunden Dateitypen

Diese Verteilung gibt einen guten Eindruck von der allgemeinen Struktur der gefunden Daten, aus sicherheitstechnischer Sicht sind jedoch seltener auftretende Dateitypen potenziell interessanter. Wie beispielsweise ein paar bacpac Dateien, bei denen es sich um Backupdateien für Microsoft SQL Datenbanken handelt [2]. Solche Dateien können für einen potenziellen Angreifer sehr viele Informationen bieten, nicht nur durch das offenlegen von interner Daten, sondern auch weil es einen Rückschluss auf die eingesetzte Datenbankarchitektur und Prozeduren ermöglicht. Es gibt wenige Szenarien, worin solche Sicherungsdateien absichtlich öffentlich verfügbar gemacht werden.

Eine weitere Möglichkeit die gefunden Daten zu kategorisieren ist die Verteilung auf die Speicherkonten. Es kann angenommen werden das, solche Konten mit vielen Datei eher absichtlich veröffentlicht sind als solche mit wenigen. Folgende Grafik analog zur vorangegangen zeigt die Verteilung der Dateien auf die 6 größten Storage Accounts.

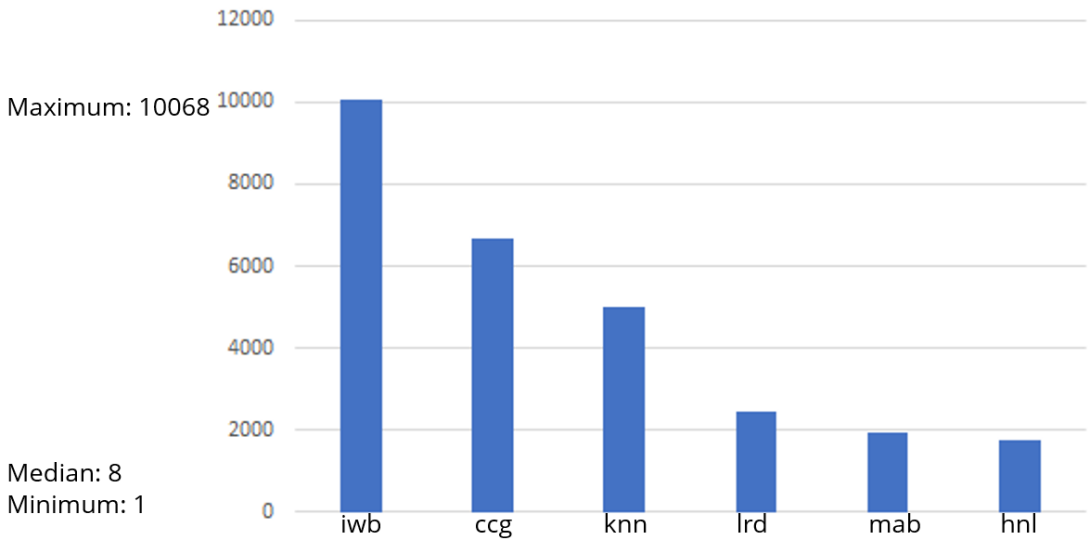

Abbildung 5: Verteilung von Dateien auf die 6 größten Speicherkonten

Auch hier lässt sich feststellen, dass ein Großteil der Dateien auf einige wenige Speicherkonten verteilt sind. In vielen gefunden Speicherkonten ist nur ein Container mit ein- bis zweistelliger Dateieinanzahl.
Mit diesen Metriken konnte ein guter Überblick über die Struktur der gefundenen Daten geschaffen werden, wobei im nächsten Schritt einzelne Dateien gezielt auf ihren Inhalt geprüft werden sollten um zu bewerten ob sie fälschlicherweise öffentlich zugänglich sind.

4 Diskussion

4.1 Bewertung der Ergebnisse

Zur Bewertung der gefunden Daten wurden zunächst Stichprobenartig einzelne Dateien des Typs png und jpg aufgerufen, welche auf einigen wenigen Speicherkonten verteilt waren. Viele Dateipfaden entsprachen dabei folgendem Schema „brx.blob.core.windows .net/images/logo.png“, wobei „images“ der Name des Storage Containers war, welcher überprüft wurde. In diesem Fall handelt es sich um ein Logo, welches wahrscheinlich als Teil einer Webseite verwendet wird. Somit kann davon ausgegangen werden das dieses und ähnliche Elemente mit Absicht öffentlich zugänglich gemacht wurden. Für große Teile der Bilddateien finden sich ähnliche Pfade und Dateinamen, was auf eine Verwendung als Webseitenkomponente schließen lässt, so gibt es abgesehen von Logos auch Mitarbeiterfotos, Produktbilder, Banner, Icons und ähnliche Bilddaten. Ebenso weißen die meisten json und js Dateien Eigenschaften und Pfadnamen auf, welche sie als Webseitenkomponenten identifizieren. Verschiedene Unternehmen scheinen Azure Storage Accounts für die Ablage von Webseitenressourcen zu verwenden, womit auch das Wissen einhergeht, dass diese Dateien öffentlich abrufbar sind.
Schwieriger ist die Auswertung bei Dokumenten wie pdf Dateien, da sich über den Titel oft nicht eindeutig bestimmen lässt ob es ein für intern oder extern bestimmtes Dokument ist. Oft lässt sich vom Titel und Ordnerpfad eine Bestimmung für die Veröffentlichung erahnen, beispielsweise bei Flyern oder Produktbeschreibungen. Für eine genaue Kategorisierung müssten jedoch alle Dokumente einzeln gesichtet werden, was für ungefähr 340 pdf-Dokumenten den Zeitrahmen dieser Arbeit jedoch überstiegen hätte. Weitere als Webseiten Komponenten identifizierte Objekte beinhalteten xml, csv und svg Dateien.
Ein Dokumententyp welcher eindeutig nicht öffentlich zugänglich sein sollte sind die zuvor erwähnten Datenbankbackupdateien vom Typ bacpac. Mit der Wiederherstellung dieser Datenbankdatei auf einem Microsoft SQL-Express Server konnte die Datenbankstruktur, sowie der Inhalt uneingeschränkt eingesehen werden. Es stellte sich heraus das es sich bei zwei der gefunden Sicherungsdateien um Datenbanken mit Patientendaten handelte. Dadurch wurde Einsicht in die persönlichen Daten der Patienten, Ihre Behandlungsgeschichte, sowie durchgeführte Untersuchungen möglich. Es lässt sich mit großer Sicherheit feststellen, dass diese Daten unter keinen Umständen unbefugten Dritten zum direkten Abruf bereitgestellt werden sollten. Es kann vermutet werden, dass diese Daten Aufgrund eines Konfigurations- oder Zuordnungsfehler unwissentlich öffentlich zugänglich gemacht wurden. Leider war kein Versuch erfolgreich den Besitzer des Storage Accounts zu identifizieren um auf den Fehler aufmerksam zu machen.

4.2 Einschränkungen

Wie in jeder Arbeit bestanden auch in dieser einige Einschränkungen, die es zu erwähnen gilt. Aufgrund des begrenzten Zeitrahmens konnten nur ausgewählte Funktionen der Azure Cloud untersucht werden, und diese Untersuchung musste auf oberflächliche Konfigurationsfehler beschränkt bleiben. Eine technisch intensivere Detailuntersuchung der angebotenen Dienste war nicht möglich. Zudem stellen die untersuchten Dienste nur einen kleinen Teil des angebotenen Spektrums dar.
Die Auswahl von Azure erfolgte aufgrund vorhandener Vorkenntnisse sowie der persönlichen Vorliebe des Autors für dieses Produkt. Die Betrachtung eines einzelnen Marktanbieters ist jedoch nicht repräsentativ für die Gesamtheit von Cloud-Infrastrukturen, daher können keine allgemeingültigen Schlussfolgerungen aus den Untersuchungen gezogen werden. Aufgrund der schnellen Veränderungen und häufigen Anpassungen von Azure-Infrastrukturen kann es sein, dass die ausgenutzten Konfigurationsfehler seitens des Anbieters behoben werden. Wie zuvor bereits erwähnt, können viele der gefundenen Daten bewusst veröffentlicht sein, und die Unterscheidung zwischen versehentlich verfügbaren internen Daten und solchen, die absichtlich öffentlich verfügbar sein sollen, ist nicht trivial. Außerdem wurden nur eine kleine Anzahl potenzieller Speicherkonten auf eine ausgewählte Liste von Containern überprüft. Der Suchraum könnte dementsprechend noch erheblich erweitert werden, insbesondere die Liste der zu prüfenden Container stellt einen kritischen Punkt dar. Je umfassender diese Liste ist, desto mehr potenzielle Ergebnisse können gefunden werden. Die Abwägung zwischen Ausführungszeit und Ergebnismenge war jedoch schwierig.
Zur Korrektur von Rechtschreibung und Grammatik wurde ein generatives KI-Modell eingesetzt, wobei sämtliche Texte zunächst in Eigenarbeit erstellt und nachträglich überarbeitet wurden. Das eingesetzte Modell war Chat-GPT 3.5 entwickelt, trainiert und bereitgestellt von OpenAI.

5 Abschluss

5.1 Fazit und Weiterentwicklung

Die zu beginn dieser Arbeit gestellte Forschungsfrage, „Welche Möglichkeiten zur Aufklärung von Azure Cloudinfrastrukturen ermöglichen Konfigurationsfehler, ohne dabei Sicherheitsmaßnahmen zu verletzen?“, lässt sich mit den durchgeführten Untersuchungen wie folgt beantworten. Es ist möglich interne Informationen aus einer Azure Cloudinfrastruktur zu gewinnen wenn Konfigurationsfehler bei der Einrichtung oder dem Betrieb von Azure Storage Accounts vorkommen. Diese Fehler können es einem unbefugten dritten ermöglichen lesenden Zugriff auf interne Dokumente zu erhalten. Außerdem ist es möglich auch über die Struktur der öffentlichen Dateipfade auf interne Strukturen zu schließen. Ein Angreifer muss hierfür keine Sicherheitsmaßnahmen umgehen, sondern nur die Eigenheiten der Azure Cloud bezüglich Ressourcenbenennung kennen. Es ist jedoch kaum möglich über die von der Azure Cloud offenbarten Informationen auf den Besitzer der gefunden Daten zu schließen. Abhängig von den öffentlichen Dokumenten kann eine Zuordnung zu einer Organisation teils nur mit erheblichen Aufwand möglich sein. Der gefundene Fehler lässt sich leicht systematisch ausnutzen und bietet großes Schadenspotenzial, dennoch ist er ebenso leicht zu umgehen. Durch sorgfältige Konfiguration kann eine Organisation verhindern das Daten unabsichtlich Dritten zur Verfügung gestellt werden. Auch kann die hier verwendete Vorgehensweise gegen die eigene Infrastruktur eingesetzt werden, um diese auf eventuell unabsichtlich verfügbare Dateien hin zu überprüfen. Aus organisatorischer Sicht kann sich ein Unternehmen gegen diesen Fehler mittels fester Ablaufpläne zur Erstellung von Storage Accounts und Richtlinien zur Dateiablage schützen.
Der Scanner könnte verwendet werden um einen größeren Suchraum abzudecken, woraus aller Wahrscheinlichkeit auch eine größere Ergebnisdatenmenge resultieren würde. Die bereits erwähnten Schwierigkeiten der Datenzuordnung zum Besitzer würden jedoch weiter bestehen, diese Informationen müsste ein Angreifer mittels andere Aufklärungstechniken in Erfahrung bringen. Es wurde im Laufe des Seminars keine inhaltlich sinnvoll Erweiterung für den Scanner erkannt, auch konnten keine weiteren Azure Dienste mit dergleichen Informationspotenzial identifiziert werden. Aufgrund der zentralisierten Anmeldung für alle Services gibt es kaum Informationen die man über eine gegebene Azure Infrastruktur gewinnen kann ohne das Schutzmaßnahmen umgangen werden müssen. Gleichzeitig stellt diese zentralisierte Anmeldung eine Bedrohung da, denn sollte ein Angreifer sie überwinden kann er anschließend jeden verwendeten Azure Service als Ziel für Folgeattacken wählen.
Aus dem Konzept des Scanners konnte ein Programm entwickelt werden, mit welchem Organisationen ihre eigenen Azure Infrastrukturen auf diesen Fehler überprüfen können. Indem mittels der entsprechenden Berechtigungen die Struktur der Azure Storage Account

exportiert wird kann sie als Input für den Scanner verwendet werden. Sollten unerwünschte Dateien öffentlichen zugänglich sein sind diese in der Ausgabe des Scanners enthalten.

5.2 Ausblick Folgesemester

Abschließend soll noch auf die Fortführung dieses Themas im Forschungsseminar des folgenden Semesters eingegangen werden. Auch wenn das Thema Azure Storage Accounts aus meiner persönlichen Sicht technisch geschlossen ist, was bedeuten soll das keine weiteren Erkenntnisse aus der Untersuchung dieses Dienstes gewonnen werden können, möchte ich das Thema Cloudsicherheit fortführen. Es ließen sich durch eine Erweiterung des Suchraums mit hoher Wahrscheinlichkeit noch deutlich größere Mengen an Daten finden und unter diesen würden sich wahrscheinlich auch wieder solche befinden, welche nicht für den öffentlichen Zugang bestimmt sind. Diese Informationen würden jedoch nur durch das Ausnutzen eines bekannten Fehlers entstehen und keine neuen Erkenntnisse bezüglich Cloud-Sicherheit bieten. Aufgrund des Grundlegend unterschiedlichen Aufbaus von Cloud Infrastrukturen ist es schwer allgemeine Schutzmaßnahmen und Angriffsszenarien zu definieren, in diesem Bereich gibt es jedoch interessante Entwicklungen wie [6] demonstriert. Ein umfassender Blick auf Schutzziele und deren Umsetzbarkeit in Cloud Umgebungen wäre denkbar um dieses Thema sinnvoll fortzuführen. Auch sollten in diesem Zusammenhang andere Cloudanbieter betrachtet werden, vor allem Amazons AWS um eine bessere Verallgemeinerung der gewonnen Erkenntnisse zu ermöglichen. Zusammenfassend lässt sich feststellen das Cloud Umgebungen neue Herausforderungen im Bereich der Informationssicherheit darstellen und durch ihre steigende Beliebtheit als Forschungsziel zunehmend an Interesse gewinnen.

Literatur

[1] *Azure Storage Account Scanner GitHub Repository.* URL: https://github.com/TomMarinovic/AzureStorageCrawler.

[2] *Export a Data-tier Application.* URL: https://learn.microsoft.com/en-us/sql/relational-databases/data-tier-applications/export-a-data-tier-application?view=sql-server-ver16.

[3] Christopher Hadnagy. *Social engineering : the science of human hacking.* ISBN: 9781119433385.

[4] Huanruo Li u. a. „Defensive deception framework against reconnaissance attacks in the cloud with deep reinforcement learning“. In: *Science China Information Sciences* 65 (7 Juli 2022). Li, H., Guo, Y., Huo, S. et al. Defensive deception framework against reconnaissance attacks in the cloud with deep reinforcement learning. Sci. China Inf. Sci. 65, 170305 (2022). ISSN: 18691919. DOI: 10.1007/s11432-021-3462-4.

[5] *Storage account overview.* URL: https://learn.microsoft.com/en-us/azure/storage/common/storage-account-overview.

[6] Lizhe Wang u. a. „Cloud Computing: a Perspective Study“. In: *New Generation Computing* 28 (2010). Wang, L., von Laszewski, G., Younge, A. et al. Cloud Computing: a Perspective Study. New Gener. Comput. 28, 137–146 (2010)., S. 137–146. DOI: https://doi.org/10.1007/s00354-008-0081-5. URL: https://link.springer.com/article/10.1007/s00354-008-0081-5.