Überprüfen von Cloud-Umgebungen

Das Nutzen von Cloud-Diensten mag Unternehmen einige Ausgaben ersparen, sicher ist es indes nicht zwingend. Mit den richtigen Tools können Sicherheitsverantwortliche die eingesetzten Dienste auditieren und die oft unsicheren Standardeinstellungen ändern.

Von Frank Ully

In den vergangenen Jahren haben Unternehmen Milliarden investiert, um Anwendungen und Daten von on Premises in öffentliche Clouds zu verlagern, vor allem zu Amazon Web Services (AWS), Microsoft Azure und Google Cloud Platform (GCP). Neben der Flexibilität und anderen operativen Vorteilen mag höhere Sicherheit ein vermeintlich gutes Argument sein – schließlich haben die Anbieter deutlich mehr Expertise, Budget und Mitarbeiter für Informationssicherheit.

Allerdings gerät die Datenwolke zunehmend ins Visier von Cyberkriminellen: Die Zahl der Angreifer mit Cloud-Expertise hat sich verdreifacht, schreibt CrowdStrike in einem Bedrohungsbericht (alle im Text zitierten Quellen, Werkzeuge und Artikel sind über ix.de/z3r7 zu finden). Auch deswegen erlitten über 80 Prozent der Organisationen im vergangenen Jahr einen kleineren oder größeren Sicherheitsvorfall mit Cloud-Bezug, wie eine Umfrage von Venafi ergab.

Denn ein Großteil der Zuständigkeit für die Sicherheit lastet im Modell der gemeinsamen Verantwortung weiterhin auf den Schultern des Cloud-Nutzers (siehe Abbildung 1). Problematisch kann es dann werden, wenn ein Unternehmen sich auf unzureichende Standardeinstellungen etwa von Microsofts Cloud-Verzeichnisdienst Entra ID – früher Azure Active Directory – verlässt oder seine Pflichten vernachlässigt und nicht die eingebauten Sicherheitsmechanismen oder kostenfreie Auditwerkzeuge von Drittanbietern nutzt.

Das führt zu kompromittierten Cloud-Umgebungen wegen fehlerhafter Konfigurationen, auf die fast alle Datenverluste aus Clouds zurückzuführen sind. Nur weil man Daten in einer Cloud speichert, hat man sich nicht der Verantwortung entledigt [1]. Für Sicherheit und Sicherung der Identitäten und Daten ist weiterhin das Unternehmen zuständig.

Die größten Cloud-Service-Provider (CSP) sind AWS und Microsoft Azure: Sie teilen sich knapp die Hälfte der weltweiten Marktanteile. Amazon dominiert, ihm gehört ein Drittel des Marktes. Besonders bei KMU beliebt ist Microsoft – wegen der engen Verzahnung mit den On-Premises-Produkten der Redmonder und dem SaaS-Angebot Microsoft 365. GCP folgt abgeschlagen mit knapp zehn Prozent Marktanteilen. Andere Anbieter wie IBM, Salesforce und Oracle belegen relativ nur Nischen.

Die in diesem Tutorial vorgestellten Auditwerkzeuge legen den Schwerpunkt auf Azure und AWS; für GCP geeignete Tools kommen vor. Mit ihnen prüfen Sicherheitsverantwortliche und Admins ihre Umgebung, ohne vorher eine Genehmigung beim Provider einzuholen. Die Tools nutzen von den Anbietern bereitgestellte APIs, um Konfigurationsdaten zu sammeln, Sicherheitslücken zu erkennen und potenzielle Risiken aufzuzeigen. Um sicherzugehen, sollte man vor weiterführenden Audits, die Richtung Penetrationstest gehen, die Kundensupportrichtlinien des jeweiligen Betreibers studieren (siehe ix.de/z3r7).

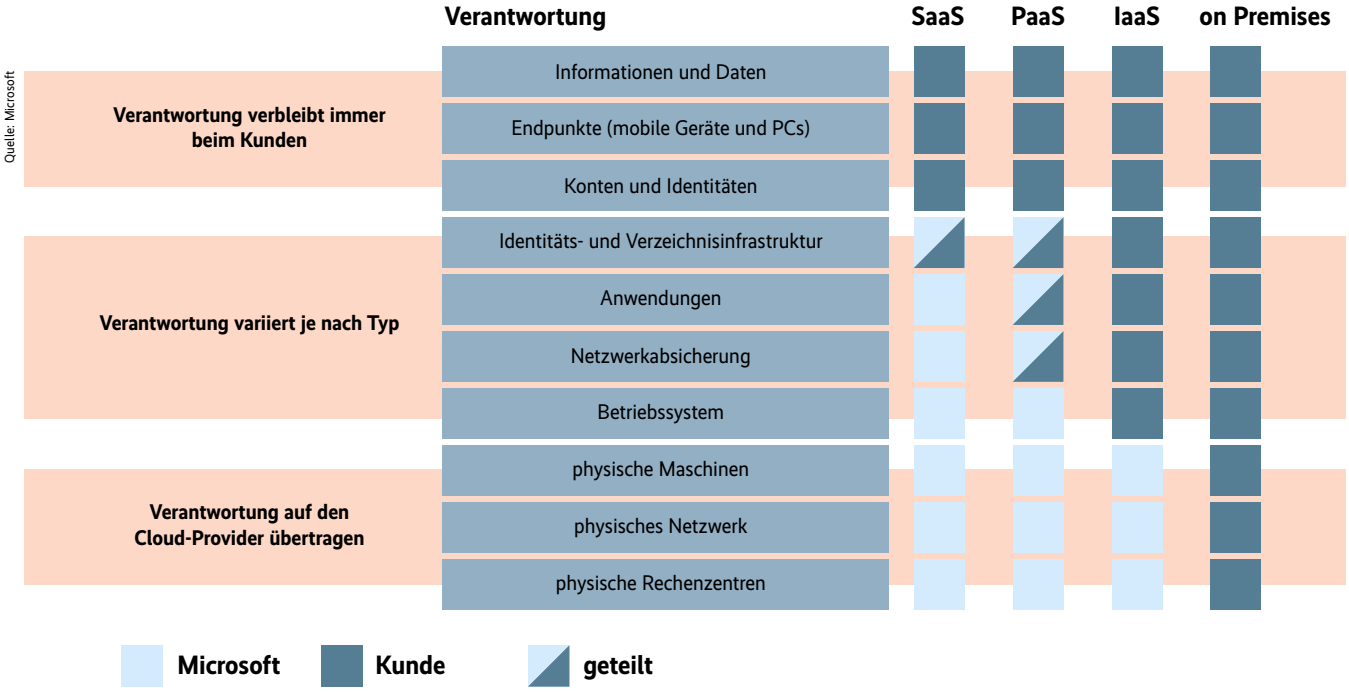
Die Cloud – auch nur Computer von anderen

Wenn Sie diesen Artikel lesen, stecken Sie schon in der Zwickmühle – Ihre Organisation setzt eine oder mehrere Clouds ein.

Einige sehenswerte Listen sammeln Belege, die zur Vorsicht beim Wechsel in

iX-TRACT

- Immer mehr Unternehmen setzen Cloud-Dienste von Amazon, Microsoft oder Google ein. Securityverantwortliche und Admins müssen ihre Sicherheitsvorkehrungen darauf ausrichten.
- Fast alle Datenverluste aus öffentlichen Clouds resultieren aus deren unsicheren Standardeinstellungen; weitere Fehlkonfigurationen entstehen beim Nutzen der Dienste.
- Kostenfreie Open-Source-Tools helfen kleinen wie großen Organisationen beim Selbst-Audit unterschiedlicher Aspekte von Datenwolken.



Microsofts Modell der gemeinsamen Verantwortung kann für Cloud-Nutzer gefährlich sein, wenn diese sich nur auf den Provider verlassen und die eigenen Pflichten vernachlässigen (Abb. 1).

die Datenwolke mahnen. Der Blogartikel „You Can't Control Your Data in the Cloud" von Karl Voit widerlegt quellensatt das viel zitierte Argument, Anbieter wie Google, Amazon oder Facebook könnten persönliche Daten besser schützen als eine Privatperson. Er beschreibt: Egal, wie sicher der CSP Daten speichert, zu einem gewissen Grad verliert man die Kontrolle; sie können nicht mehr wirklich gelöscht werden oder der Zugriff oder die Daten selbst können verloren gehen. Eine Lektüre lohnt ebenfalls, wenn man Unternehmensdaten in die Cloud packen will oder es schon getan hat.

Einen Vorteil bieten öffentliche Clouds: In Unternehmensnetzen gelingt regelmäßiges und flächendeckendes Einspielen von Patches oft mehr schlecht als recht. Diese Herausforderung hat die Wolke nicht, weil die Betreiber Bugs für alle Nutzer beheben. Die „Open Cloud Vulnerability & Security Issue Database" (kurz CloudVulnDB) katalogisiert kritische Schwachstellen, die Sicherheitsforscher gefunden haben. Diese Datenbank ist in erster Linie auf die Seite des Anbieters im Modell der gemeinsamen Verantwortung ausgerichtet.

Im Gegensatz dazu konzentriert sich der „Cloud Security Atlas" auf umsetzbare Praktiken, die im Verantwortungsmodell auf der Seite des Kunden liegen. Der Atlas verzeichnet ein breites Spektrum an realen Angriffen. Das hilft, ein tieferes Verständnis des jeweiligen Angriffs zu erlangen: wie er funktioniert und wie man ihn in der eigenen Umgebung verhindert und erkennt. Zusätzlich deckt der Atlas Schwachstellen und Fehlkonfigurationen ab, die häufig in Clouds zu finden sind. Die durchsuchbare Datenbank lässt sich nach Provider und Art des Risikos filtern sowie nach Auswirkung, Ausnutzbarkeit und Aktualität sortieren. Ein Blogbeitrag beschreibt die Hintergründe.

Schließlich verzeichnet das Kompendium „Public Cloud Security Breaches" Sicherheitsvorfälle bei Nutzern der wichtigsten Anbieter. Es hilft Sicherheitsexperten, die Risiken bestimmter Fehler zu formulieren und ihre Teams oder Vorgesetzten darüber zu informieren. Als Anlaufstelle dokumentiert das Verzeichnis greifbare Beispiele dafür, wie sich Fehlkonfigurationen auswirken.

Ressourcen in Clouds unauthentisiert aufspüren

Der öffentliche Zugriff auf Ressourcen ist zu einfach zu konfigurieren, weswegen man rasch ungewollt Dienste und Daten aus dem gesamten Internet zugänglich macht – besonders gefährlich in allen Clouds. Ein Blogartikel bei Ermetic beschreibt dieses Risiko bezogen auf Azure (siehe ix.de/z3r7), es besteht aber bei allen Anbietern. Selbst sicherheitsbewusste Organisationen sind betroffen: Ein ausgelagerter Mailserver des US-Verteidigungsministeriums machte ohne Passwort militärische Nachrichten lesbar. Sogar Provider erwischt es; so veröffentlichten KI-Forscher von Microsoft unbeabsichtigt einen Zugriffsschlüssel zu einem Azure-Speicherkonto, in dem Terabyte an Daten lagen: darunter Passwörter, private Schlüssel und interne Teams-Kommunikation.

Wichtig für Verteidiger ist zunächst, aus der Perspektive eines Gelegenheitsangreifers womöglich bereitgestellte Dienste zu ermitteln. Dazu dient Open Source Intelligence (OSINT) – das Sammeln öffentlicher Informationen –, mit dem sich der vorige Artikel dieser Reihe befasste.

cloud_enum ist ein Multi-Cloud-OSINT-Werkzeug, das allgemein zugängliche Ressourcen in den drei großen Clouds enumeriert; neben Speicherkonten (Buckets) in allen Plattformen zum Beispiel awsapps bei Amazon, virtuelle Maschinen in Azure und Firebase-Datenbanken bei Google. Am besten installieren Verteidiger die in Python geschriebene Anwendung in Kali Linux. Der erste Tutorialteil zeigte, wie es einzurichten ist.

Listing 1 packt das Tool samt notwendigen Bibliotheken in eine eigene virtuelle Python-Umgebung, um Python-Bibliotheken für das Betriebssystem und andere Programme nicht durcheinanderzubringen.

Tutorialinhalt

Teil 1: Scannen und Verifizieren der eigenen Systeme

Teil 2: Webapplikationen angreifen

Teil 3: Auditieren interner Netzwerke, Domänen und Systeme

Teil 4: Sammeln öffentlich verfügbarer Informationen und Analysieren der Angriffsfläche

Teil 5: Überprüfen von Cloud-Umgebungen

Workshop

iX-Workshop „Wie Angreifer vorgehen – Pentesting mit Open-Source-Werkzeugen“

In diesem von Tim Mittermeier abgehaltenen iX-Workshop lernen die Teilnehmenden, wie sie Angriffsvektoren in ihrer eigenen Unternehmens-IT durch die Anwendung von Hacking-Techniken aufdecken und beseitigen können. Im Mittelpunkt stehen Themen wie das Sammeln und Auswerten öffentlich verfügbarer Informationen (OSINT), die Untersuchung auf Netzwerkebene und die Überprüfung von Webanwendungen mit Open-Source- und Auditing-Tools sowie die Privilegieneskalation unter Windows und Linux. Ein besonderer Schwerpunkt liegt auf dem zentralen Verzeichnisdienst Active Directory.

Der Workshop findet online statt, weitere Informationen gibt es unter https://heise.de/s/g1E2.

Abbildung 2 zeigt den Aufruf von cloud_enum aus der aktivierten Python-Umgebung heraus und die Suche nach einem Stichwort für die untersuchte Organisation, etwa Unternehmens- oder Produktnamen. Den k-Parameter kann man mehrfach mit unterschiedlichen Stichwörtern angeben.

Das Enumerationswerkzeug bildet aus dem jeweiligen Begriff und der optional veränderbaren Wörterliste mögliche Ressourcennamen wie `admin-heise.s3.amazonaws.com` oder `www.googleapis.com/storage/v1/b/heise-secret/`, prüft, ob es diese standardisierten DNS- oder API-Einträge gibt, und, wenn ja, ob man unauthentisiert auf die Daten zugreifen kann. Verteidiger verifizieren die Ergebnisse – schließlich kann jedermann Ressourcen mit beliebigen Namen anlegen – und entfernen Unternehmensdaten, die unabsichtlich zugänglich gemacht wurden.

Ein OSINT-Webdienst, der Daten in öffentlichen Buckets durchsuchbar macht, ist GrayhatWarfare. Die ersten Ergebnisse sind ohne Registrierung sichtbar. Will man alle Ergebnisse anzeigen, kauft man einen relativ preiswerten Premium-Zugang.

Authentisiert Ressourcen inventarisieren

Einen noch besseren Überblick über Ressourcen – neudeutsch Assets –, die eine Organisation in öffentlichen Clouds bereitstellt, erhalten Verteidiger, wenn sie sich mit passenden Zugangsdaten anmelden und die Assets inventarisieren.

Der vierte Tutorialteil stellte Project Discovery vor: Das Projekt veröffentlicht Open-Source-Tools für aktive und passive Informationssammlung. Das Kommandozeilenwerkzeug cloudlist inventarisiert Ressourcen bei den großen Betreibern, aber auch kleineren CSP wie Heroku oder Hetzner – und aus herstellerunabhängigen Technologien wie Kubernetes.

Hat man den Open-Source-Tool-Manager pdtm wie im vorigen Artikel beschrieben in Kali Linux installiert, steht cloudlist bereit. Mit `pdtm --update-all` aktualisiert man die Werkzeugsammlung.

Nun konfiguriert man in einer neu angelegten Datei `$HOME/.config/cloudlist/provider-config.yaml` Zugangsdaten für die verwendeten CSP, auf dem cloudlist-GitHub-Repo in einer beispielhaften Konfiguration dargestellt. Welche Berechtigungen notwendig sind, dokumentiert die Liste der unterstützten Provider (siehe ix.de/z3r7).

Mit passenden Werkzeugen überprüfen

Anschließend rufen Verteidiger die Anwendung ohne Parameter auf. Die ausgegebenen IP-Adressen und Domänennamen prüfen sie mit Internet-of-Things-Suchmaschinen wie Shodan oder Censys auf erreichbare Dienste oder übergeben sie direkt an andere Werkzeuge wie Port- oder Schwachstellenscanner.

Folgender Befehl untersucht die ausgelesenen und erreichbaren Cloud-Dienste der Organisation mit dem anpassbaren Schwachstellenscanner nuclei, ebenfalls von Project Discovery:

```
$ cloudlist -silent | httpx -silent ↵
                                | nuclei
```

Weitere Geheimtipps aus dem gut ausgestatteten Werkzeugkasten des Projekts verrät ein Blogartikel.

Der umgekehrte Weg: Von der IP-Adresse zum Cloud-Anbieter

Das Python-Skript ip2provider geht den umgekehrten Weg: Füttert man es mit IP-Adressen, zeigt es an, ob ein Cloud-Betreiber diese verwaltet.

Das Tool arbeitet mit zehn Providern zusammen; die Liste steht im GitHub-Repository. Installiert wird das Skript in einer virtuellen Python-Umgebung wie in Listing 2.

Ist die Umgebung aktiviert, lässt sich für eine Liste von IP-Adressen wie folgt prüfen, welche davon sich in welcher Cloud befinden. Die Adressen müssen jeweils in einer eigenen Zeile stehen.

```
$ cat ~/ipaddresses.txt | ↵
                    ./ip2provider.py
18.173.154.113 aws AMAZON GLOBAL
52.218.100.202 aws AMAZON eu-west-1
```

Hat man wie im vierten Tutorialteil beschrieben (Abschnitt „Arbeitserleichternde Werkzeuge“) per OSINT die Subdomänen einer Organisation ermittelt und nach IP-Adressen aufgelöst, sieht man jetzt schnell, welche Clouds sie nutzt.

Ein alternatives Werkzeug ist edge. Es unterstützt nur die drei großen CSP. Das Projekt ipranges sammelt IP-Adressbereiche von diesen und weiteren Providern in einzeln oder kombiniert herunterladbaren Textdateien.

Auf Herz und Nieren prüfen mit dem Multi-Cloud-Auditor

Das quelloffene Audit-Tool ScoutSuite der NCC Group untersucht AWS, Azure,

Listing 1: Installieren von cloud_enum in einer virtuellen Python-Umgebung

```
$ git clone --depth=1 https://github.com/initstring/cloud_enum.git
$ sudo apt update && sudo apt install -y python3-venv
$ cd cloud_enum && python3 -m venv venv
$ source ./venv/bin/activate
$ pip3 install -r requirements.txt
```

Listing 2: Installieren von ip2provider, das anzeigt, ob eine IP-Adresse in der Cloud gehostet wird

```
$ git clone --depth=1 https://github.com/oldrho/ip2provider.git
$ sudo apt update && sudo apt install -y python3-venv
$ cd ip2provider && python3 -m venv venv
$ source ./venv/bin/activate
$ pip3 install -r requirements.txt && pip3 install requests
```

GCP sowie exotischere Clouds wie Alibaba und Oracle auf unsichere Standardeinstellungen und typische Fehlkonfigurationen. Anstatt Dutzende Seiten auf Webkonsolen zu durchforsten und etwa nach den CIS-Benchmark-Dokumenten für den jeweiligen Anbieter durchzusehen, präsentiert ScoutSuite Admins und Sicherheitsverantwortlichen eine klare Sicht auf die Angriffsfläche.

Bei der Installation in Kali Linux vereinfacht das Helferlein Pipx das Verwalten mehrerer Python-Umgebungen: Damit installierte Kommandozeilentools lassen sich direkt aufrufen. Pipx funktioniert allerdings nicht für jedes Python-Projekt.

```
$ sudo apt update && sudo apt ↵
                    install -y pipx
$ pipx ensurepath && source ~/↵
                              .profile
$ pipx install scoutsuite
```

Um Ressourcen in Azure zu prüfen, legt man sich am besten einen Audit-Benutzer an. In Microsoft Entra ID bekommt er die Rollen Globaler Leser und Sicherheitsleseberechtigter. Zusätzlich wird er mit den Azure-Rollen Leser und Sicherheitsleseberechtigter auf der Stammverwaltungsgruppe versehen oder manuell in jedem Azure-Abonnement, das geprüft werden soll. Alles zu den Rollensystemen in der Microsoft-Cloud gibt es unter [2] zu lesen.

Mit folgenden Zeilen installieren Admins die benötigte Azure-Befehlszeilenschnittstelle, melden sich daran mit dem Audit-Benutzer an und starten die Prüfung. Beim Anmelden wird ein eventueller zweiter Faktor abgefragt, der durch Sicherheitsstandards oder Richtlinien für bedingten Zugriff hoffentlich für alle Benutzer erzwungen wird [3].

```
$ pipx install azure-cli
$ az login
$ scout azure --cli
```

Nach erfolgreicher Ausführung öffnet ScoutSuite automatisch einen Bericht im HTML-Format im Standardbrowser (siehe Abbildung 3).

Der Bericht deckt zahlreiche Risiken auf, etwa Gastbenutzer und öffentliche Speicherkonten. Im Haupt-Dashboard sieht man alle bereitgestellten Dienste, die Anzahl der gescannten und von Befunden betroffenen Ressourcen sowie die Zahl der durchgeführten Tests. Anhand der roten und gelben Symbole erkennt man auf einen Blick, bei welchem Ressourcentyp Handlungsbedarf herrscht. Ein Klick auf eine Tabellenzeile öffnet die Ergebnisse zum Typ. Im jeweiligen Dashboard (siehe Abbildung 4) klappen Details zum Beheben unter dem Pluszeichen auf; nach Klicken auf den Befundtitel sieht man, welche Ressourcen davon betroffen sind.

```
┌──(kali㉿kali)-[~]
└─$ cd cloud_enum

┌──(kali㉿kali)-[~/cloud_enum]
└─$ source ./venv/bin/activate

┌──(venv)─(kali㉿kali)-[~/cloud_enum]
└─$ ./cloud_enum.py -k heise -m ~/cloud_enum/enum_tools/fuzz.txt

############################
        cloud_enum
   github.com/initstring
############################


Keywords:    heise
Mutations:   /home/kali/cloud_enum/enum_tools/fuzz.txt
Brute-list:  /home/kali/cloud_enum/enum_tools/fuzz.txt

[+] Mutations list imported: 242 items
[+] Mutated results: 1453 items

+++++++++++++++++++++++++++
      amazon checks
+++++++++++++++++++++++++++

[+] Checking for S3 buckets
  Protected S3 Bucket: http://heise.s3.amazonaws.com/
  Protected S3 Bucket: http://heisearchive.s3.amazonaws.com/
  Protected S3 Bucket: http://heisebackup.s3.amazonaws.com/
  Protected S3 Bucket: http://heisestorage.s3.amazonaws.com/

 Elapsed time: 00:02:06

[+] Checking for AWS Apps
[*] Brute-forcing a list of 1453 possible DNS names

 Elapsed time: 00:00:22


+++++++++++++++++++++++++++
       azure checks
+++++++++++++++++++++++++++

[+] Checking for Azure Storage Accounts
[*] Brute-forcing a list of 471 possible DNS names
  HTTPS-Only Storage Account: http://storageheise.blob.core.windows.net/

 Elapsed time: 00:00:07
```

Das OSINT-Tool cloud_enum, aufgerufen aus der aktivierten Python-Umgebung, sucht nach Ressourcen, die das Stichwort enthalten (Abb. 2).

Audit-Benutzer anlegen, Zugriffe sicher gestalten

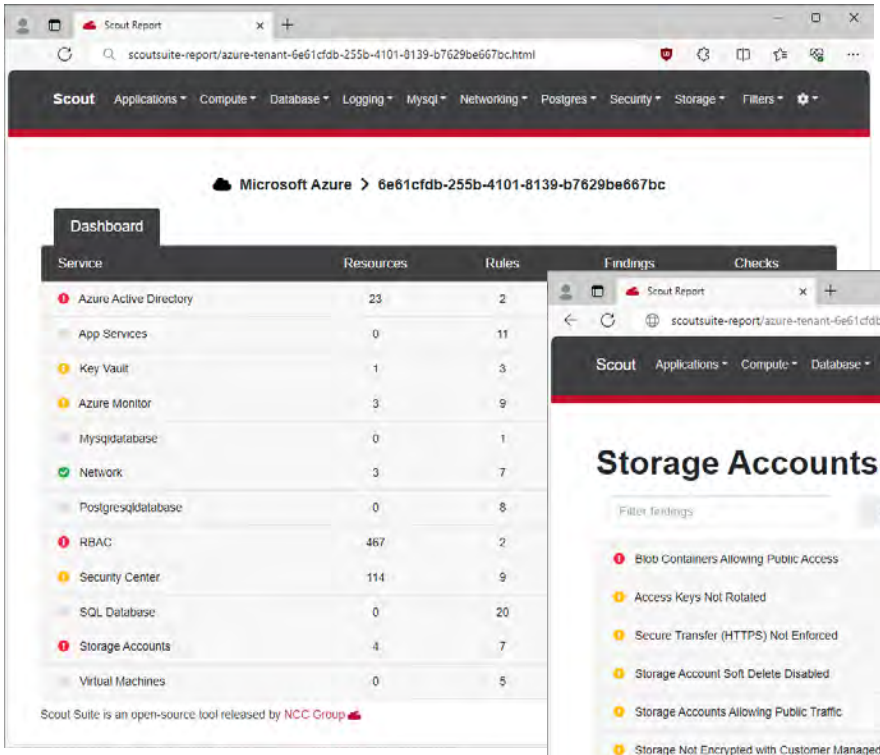
Um AWS zu prüfen, legen Admins in dessen Webkonsole im Identity and Access Management (IAM) einen neuen Benutzer „audit“ an und wählen im folgenden Schritt das direkte Anfügen von Richtlinien. Bei den Berechtigungsrichtlinien filtern sie nach „AWS-verwaltet – Aufgabenfunktion“, suchen und aktivieren die Richtlinien ReadOnlyAccess sowie SecurityAudit. Ist der Benutzer angelegt, klickt man auf ihn und wechselt auf die Registerkarte für Sicherheitsanmeldeinformationen. Dort erstellt man einen neuen Zugriffsschlüssel für den Anwendungsfall Befehlszeilenschnittstelle, bestätigt Nachfragen und legt die angezeigten Schlüsseldaten sicher ab, beispielsweise in einem Passwortmanager [1].

Listing 3 zeigt, wie nach dem Installieren der AWS-Befehlszeile und dem Konfigurieren der eben angelegten (unkenntlich gemachten) Zugriffsschlüssel ScoutSuite für Amazon Web Services startet.

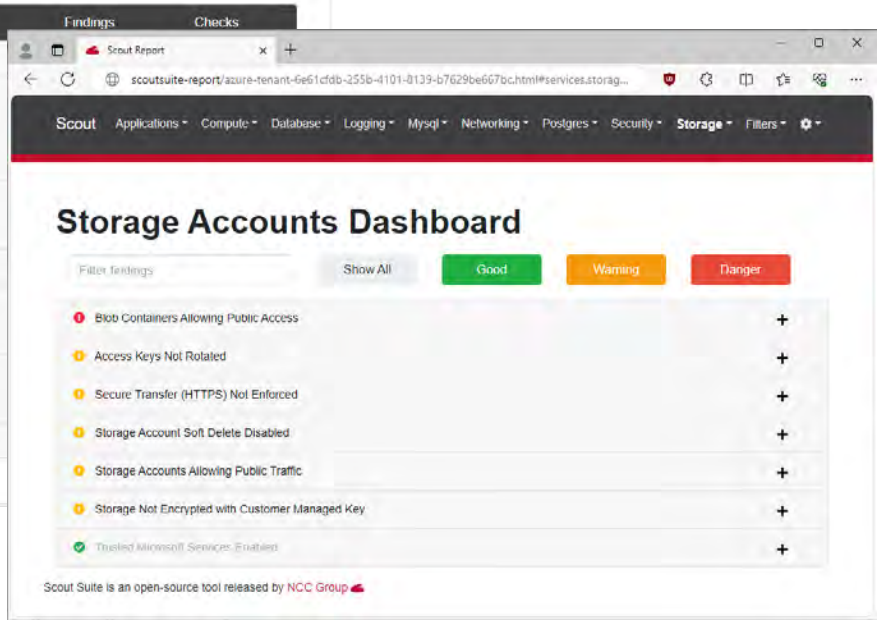
Das umfangreiche ScoutSuite-Wiki (siehe ix.de/z3r7) beschreibt, wie man eingebaute Regeln aus- und einschaltet, Regelsätze anpasst, Berichtsdaten exportiert und weitere Clouds wie GCP prüft.

Spezialist für AWS

Ein echter Spezialist für Amazon Web Services und das populärste Open-



Der HTML-Bericht des Multi-Cloud-Auditwerkzeugs ScoutSuite für Azure offenbart diverse Risiken und weitere Details (Abb. 3).



Befunde für einen bestimmten Ressourcentyp im Multi-Cloud-Auditwerkzeug ScoutSuite (Abb. 4).

Source-Auditwerkzeug dafür ist Prowler. Der, wörtlich übersetzt, Herumtreiber testet AWS auf fast dreihundert Arten. Um alle Prüfungen durchführen zu können, fügt man beim angelegten Audit-Benutzer eine neue Inline-Berechtigung hinzu, in die man die Richtliniendefinition `prowler-additions-policy.json` aus dem GitHub-Repo kopiert.

Die folgenden Befehle installieren Prowler und untersuchen die eigene Amazon-Umgebung. Der Aufruf geht davon aus, dass man bereits ScoutSuite für AWS wie beschrieben ausgeführt und damit die Voraussetzungen für die Nutzung von Prowler geschaffen hat.

```
$ pipx install prowler
$ ulimit -n 4096 && aws logout
$ prowler aws
```

Die Prüfung dauert einige Zeit. Die Ergebnisse erscheinen in einer Übersicht auf der Kommandozeile und werden detailliert im Verzeichnis output als HTML-, CSV- und JSON-Dateien abgelegt. Im grafischen Bericht (Abbildung 5) zeigen die Bedienelemente über der Tabelle auf Wunsch alle Einträge an, die Schaltfläche filtert nach Kritikalität oder Dienst – und das Eingabefeld sucht im Volltext.

Prowler treibt sich ebenso in anderen Datenwolken herum: Für GCP unterstützt es derzeit etwa achtzig Tests und für Azure knapp über zwanzig. Seine Dokumentation (siehe ix.de/z3r7) hilft beim Einsetzen.

Schattenadministratoren aufspüren

„Identität ist der neue Perimeter“ ist eine Devise, die man im Zusammenhang mit der Cloud immer wieder hört. Beim Auffinden offensichtlicher Administratoren und – unerwartet – hoch privilegierter Benutzer, auch als Schattenadministratoren bezeichnet, in den beiden größten Clouds helfen die PowerShell-Skripte AWSStealth und AzureStealth, Teil der SkyArk-Sammlung.

Um die notwendigen Standardmodule der Anbieter aus dem PowerShell-Katalog im Web zu installieren, aktualisieren Systemverwalter zunächst in einer administrativen Sitzung der klassischen Windows PowerShell die dafür notwendige Komponente:

```
PS C:\Windows\system32> Install-↵
          Module PowerShellGet -Force
```

Anschließend installiert man in einer normalen Sitzung von Windows PowerShell wie in Listing 4 die benötigten PowerShell-Module für Azure AD (das Modul heißt immer noch so) und Azure sowie SkyArk selbst. Besonders das Einspielen des Az-Moduls kann einige Zeit dauern.

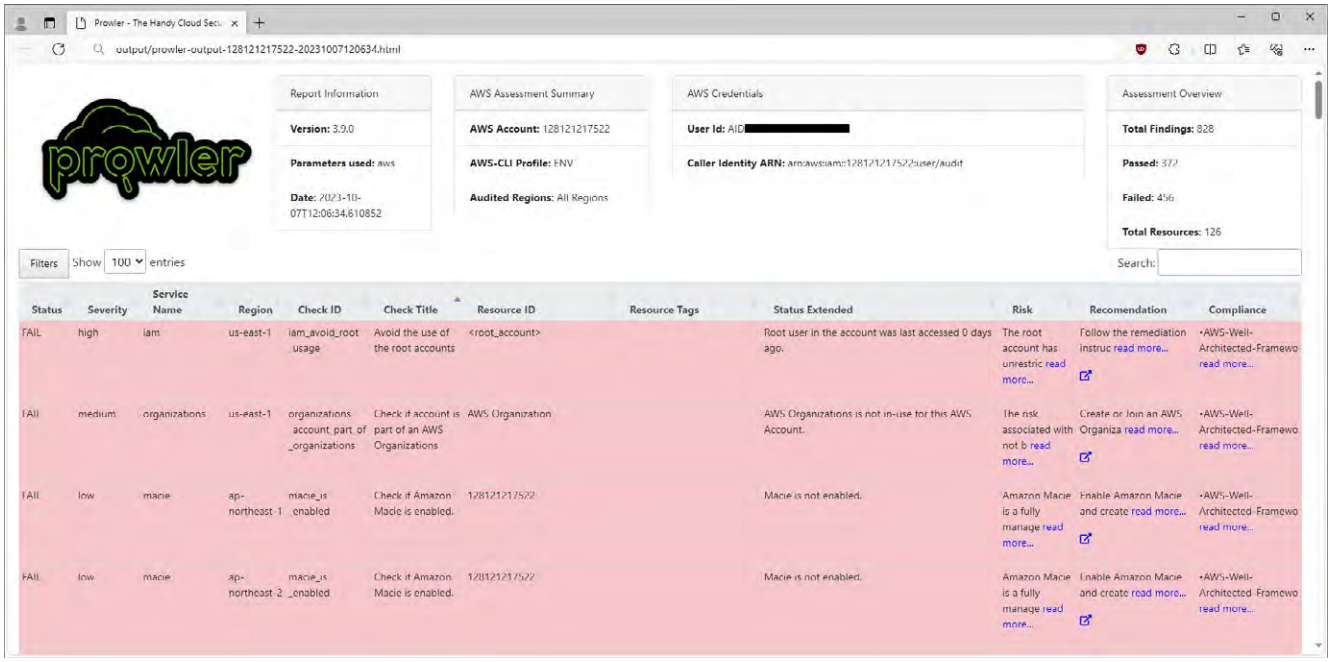
Ist man noch in derselben PowerShell-Sitzung wie im Listing 4, sucht folgender Befehl nach Schattenadmins in Azure:

```
PS > Start-AzureStealth
```

Die Nachfrage, ob das US-basierte Azure verwendet werden soll, bestätigt man und meldet sich mit dem zuvor angelegten Audit-Nutzer an. Das Skript gibt die

Listing 3: Vorbereitung und Start des ScoutSuite-Audits für AWS

```
$ sudo apt install -y awscli
$ aws configure
AWS Access Key ID [None]: AKI****************
AWS Secret Access Key [None]: 4Lt*************************************
Default region name [None]:
Default output format [None]:
$ scout aws
```



Der HTML-Bericht des auf AWS spezialisierten Auditwerkzeugs Prowler ist im Volltext durchsuchbar und die Ergebnisse lassen sich nach diversen Kriterien filtern (Abb. 5).

Ergebnisse in unterschiedlichen Formaten aus, darunter Text- und CSV-Dateien.

Auch die Amazon-Cloud lässt sich prüfen. Dabei wird automatisch das AWS-PowerShell-Modul installiert. Bei der Frage nach Zugangsdaten gibt man die ID und den Zugriffsschlüssel des für Audits angelegten IAM-Benutzers (Identity and Access Management) an.

```
PS > Start-AWStealth
```

Anhand der Scanergebnisse ermitteln Unternehmen die Sicherheitsprinzipale – Benutzer, Gruppen und Rollen – mit den sensibelsten und riskantesten Berechtigungen. Angreifer haben es auf diese Identitäten abgesehen.

Verteidiger sollten sicherstellen, dass es möglichst wenige dieser Sicherheitsprinzipale gibt und dass die privilegierten Benutzer gut geschützt sind: mit starken und sicher gespeicherten Anmeldeinformationen, mit aktivierter Mehr-Faktor-Authentifizierung und im Idealfall einem automatisierten Monitoring, wenn sich damit jemand anmeldet. Haben Sicherheitsverantwortliche die notwendigen Premium-Lizenzen für Entra ID übrig, sollten sie über die Einführung des PIM-Dienstes (Privileged Identity Management) nachdenken [3]. Darüber hinaus sollten die SkyArk-Skripte von Zeit zu Zeit laufen und so verdächtige Abweichungen in der Liste der privilegierten Identitäten zeigen.

Microsoft 365 sicher konfigurieren

Da Microsoft 365 (M365) das beliebteste Software-as-a-Service-Angebot ist – auch bei US-Bundesbehörden –, hat die amerikanische Cybersecurity-Agentur CISA dafür SCuBAGear entwickelt. Administratoren aller Organisationen mit M365-Mandanten nutzen das Tool, um Konfigurationslücken schnell zu identifizieren und zu beheben.

Das in PowerShell geschriebene Programm prüft, ob die Konfiguration eines Mandanten die Mindestanforderungen der CISA aus ihren Richtlinien zu Secure Cloud Business Applications erfüllt (daher die Abkürzung SCuBA und das Wortspiel im Toolnamen, übersetzt „Tauchausrüstung“). Die Konfigurationsleitfäden beschreiben die sichere Konfiguration von Azure Active Directory (die Umbenennung ist derzeit dort noch nicht angekommen), Exchange Online, Power BI und Power Plattform, dem eng verwobenen Anwendungstrio OneDrive for Business, SharePoint Online und MS Teams sowie Microsoft Defender für Office 365. Die Richtlinien kann man sich im GitHub-Repository des Projekts ansehen.

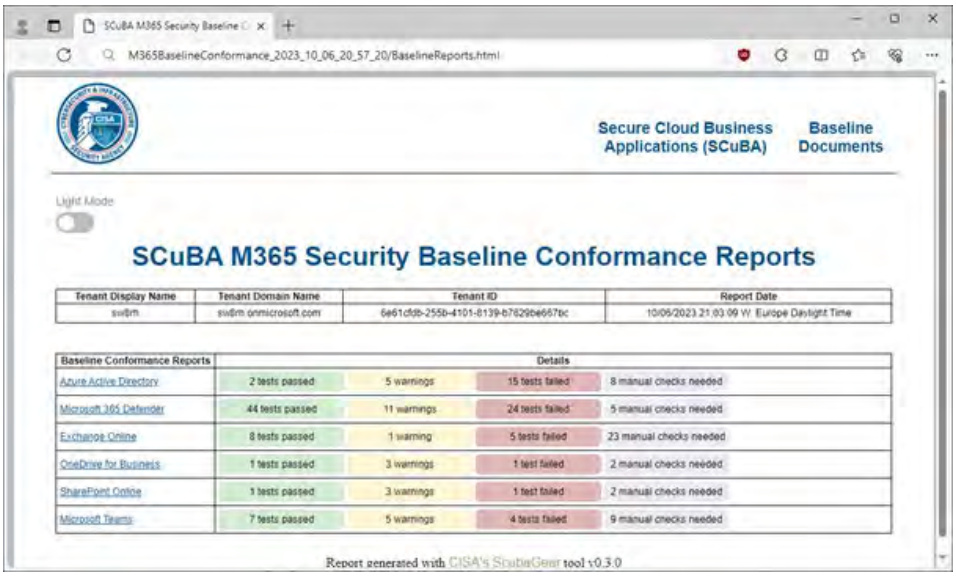
Das Auditwerkzeug ist in einem frühen Entwicklungsstadium und wird aktiv erweitert – es läuft in Tests verlässlich. Damit es alle Richtlinien prüft und nicht pauschal als nicht konform bewertet, sind höherwertige Lizenzen notwendig,

Listing 4: Installieren von Modulen und SkyArk in einer unprivilegierten Windows PowerShell

```
PS > Set-ExecutionPolicy Unrestricted -Scope Process -Force
PS > Install-Module AzureAD -Scope CurrentUser -Force
PS > Install-Module Az -Scope CurrentUser -Force
PS > wget https://github.com/cyberark/SkyArk/archive/refs/heads/master.zip -O SkyArk.zip
PS > Expand-Archive .\SkyArk.zip
PS > cd .\SkyArk\SkyArk-master\
PS > Import-Module .\SkyArk.ps1 -Force
```

Listing 5: Einrichten und Aufrufen von SCuBAGear

```
PS > Set-ExecutionPolicy Unrestricted -Scope Process -Force
PS > .\SetUp.ps1
PS > Import-Module -Name .\PowerShell\ScubaGear
PS > Invoke-SCuBA
```



HTML-Bericht des auf Microsoft 365 und Entra ID spezialisierten Auditwerkzeugs SCuBAGear. Die mit den Befunden verbundenen Empfehlungen wirken sich auf die Endbenutzer aus (Abb. 6).

wie auf der GitHub-Seite dokumentiert (siehe ix.de/z3r7).

Dort findet sich immer die aktuelle Anleitung zum Installieren und Verwenden des Tools. Beim Verfassen des Artikels ist 0.3 die neueste Version, die man dort aus dem Release-Bereich herunterlädt und entpackt. Listing 5 zeigt, wie SCuBAGear aus einer normalen Windows PowerShell startet, nachdem man in sein Verzeichnis gewechselt ist.

Bei der Frage nach Anmeldeinformationen verwendet man am einfachsten einen globalen Administrator, er kann alle notwendigen Berechtigungen erteilen. Administratorrechte sind nach dem Prinzip des Least Privilege grundsätzlich nur dann einzuräumen, wenn es zwingend erforderlich ist, so in diesem Fall [1]. Die Rollen mit den geringsten Rechten finden sich auf der GitHub-Seite. Im Dialog „Angeforderte Berechtigungen" aktiviert man das Kontrollkästchen „Zustimmung im Namen Ihrer Organisation". Der Anmeldebildschirm ploppt mehrmals auf.

Einige der Empfehlungen im Bericht, im HTML- (Abbildung 6) oder JSON-Format, haben größere Auswirkungen auf die Endbenutzer. Admins sollten sicherstellen, dass sie verstehen, was diese Richtlinien bewirken.

Ebenfalls einen Blick wert ist der Microsoft Defender for Office 365 Recommended Configuration Analyzer (ORCA), selbst ohne Defender-Lizenzen. Das Tool Monkey365 prüft sowohl Azure, M365 wie auch Entra ID, ist allerdings sperrig zu bedienen. In einem Datenblatt stellt die CISA weitere ihrer kostenfreien Werkzeuge für die Microsoft-Cloud vor (siehe ix.de/z3r7).

Der purpurne Ritter für ein sicheres Entra ID

Purple Knight ist ein Tool, das Fehlkonfigurationen und Verdachtsmomente in On-Premises-Active-Directories aufspürt. Es wurde im dritten Tutorialartikel vorgestellt. Seit Kurzem prüft es auch Azure AD/Entra ID. Das Werkzeug ist proprietär, gegen Abgabe von Kontaktdaten lädt man es beim Entwickler Semperis in einer Community-Edition kostenfrei herunter.

Damit der purpurne Ritter die Daten aus dem Cloud-Verzeichnisdienst abfragen kann, muss man einen Dienstprinzipal einrichten. Listing 6 zeigt, wie es automatisiert mit PowerShell geht. Das setzt voraus, dass die Module AzureAD und Az wie in Listing 4 bereits installiert sind. Die erzeugten Zugangsdaten kopiert man und speichert sie sicher, etwa in einem Passwortmanager. Sie sind nur eine Stunde gültig. Falls das automatisierte Anlegen des Dienstprinzipals scheitert, steht eine manuelle Anleitung im Getting Started Guide im Purple-Knight-Ordner.

Zum Audit startet ein Systemverwalter die ausführbare Datei, stimmt den Lizenzbedingungen zu, aktiviert das Kontrollkästchen Azure Active Directory und fügt aus der Ausgabe des PowerShell-Skripts die Felder Tenant ID, Application App ID und Client Secret ein. Der Bericht wird als HTML-Datei (Abbildung 7), PDF und Excel-Tabelle erzeugt. Die Konfigurationsempfehlungen für den Cloud-Verzeichnisdienst überschneiden sich naturgemäß mit jenen aus SCuBAGear.

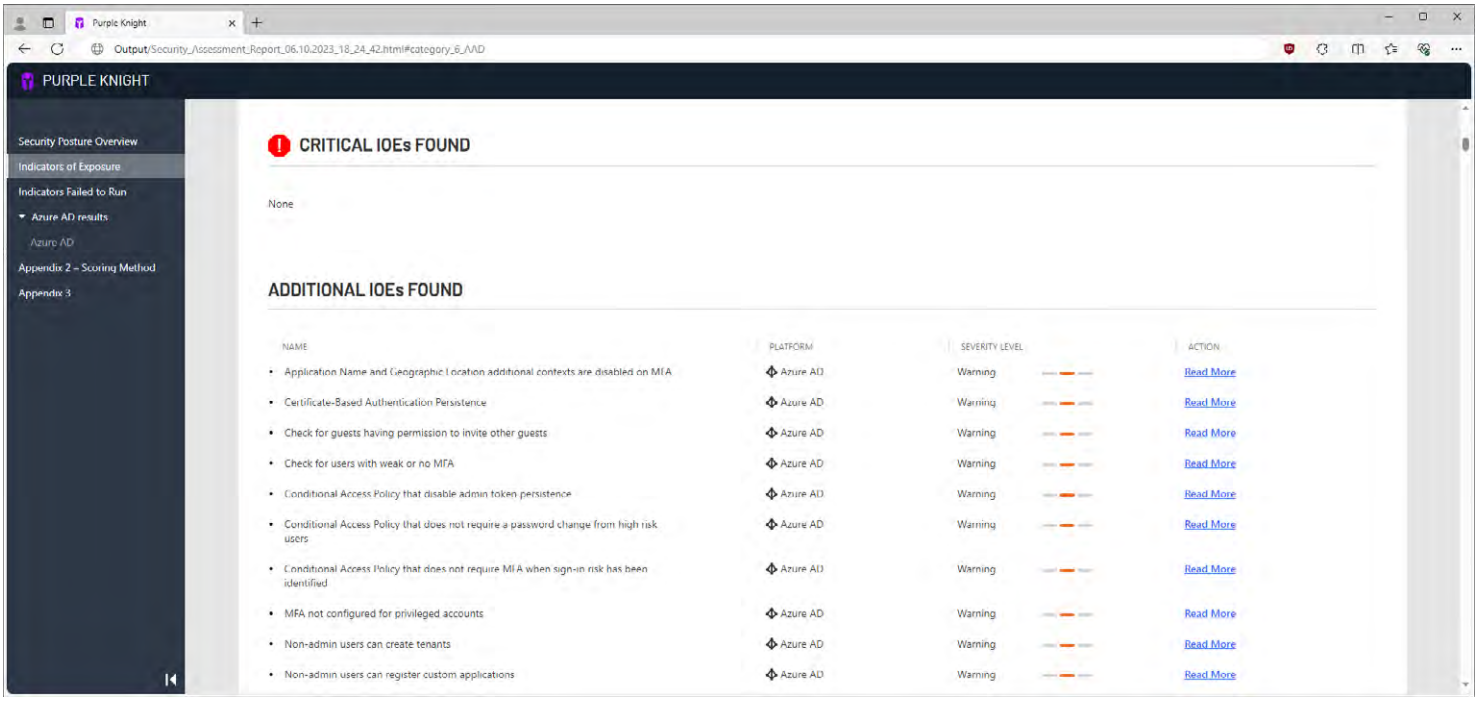
Das AD-Audittool PingCastle kann ebenfalls den Cloud-Verzeichnisdienst prüfen. Es spielt interessante Daten aus, etwa über Anwendungen und Dienstprinzipale mit kritischen Berechtigungen, hat aber (noch) wenig eingebaute Regeln für Entra ID/Azure AD, die direkt auf Verbesserungspotenziale aufmerksam machen.

Geheimnisse in Quellcode finden mit dem Trüffelschwein

Entwickler speichern Geheimnisse wie Passwörter, Zugriffstoken oder private Schlüssel leider häufig in Quelltext- oder Konfigurationsdateien, ob bewusst oder unabsichtlich. Selbst wenn sie ihren Fehler bemerken und die sensiblen Daten löschen, verbleiben diese in vorherigen Dateiständen in einer Versionsverwaltung wie Git. Gerade wenn das Software-Repository öffentlich ist, sind solche Geheimnisse für Angreifer willkommene Beute.

Listing 6: Einrichten des Dienstprinzipals für Purple Knight

```
PS > Set-ExecutionPolicy Unrestricted -Scope Process -Force
PS > Install-Module Microsoft.Graph -Scope CurrentUser -Force
PS > $cred = Get-Credential
PS > Connect-AzureAD -Credential $cred
PS > Connect-AzAccount -Credential $cred
PS > wget https://raw.githubusercontent.com/Semperis/PK-AADAppReg/main/Create-Update-Delete-AAD-PK-Vulnerability-Scanning-
App.ps1 -OutFile Create-Update-Delete-AAD-PK-Vulnerability-Scanning-App.ps1
PS > .\Create-Update-Delete-AAD-PK-Vulnerability-Scanning-App.ps1 -aadTenantFQDN (Get-AzureADDomain).Name ↵
                                -appRegDisplayName "Semperis Purple Knight Vulnerability Scanning App" -createOrUpdateApp ↵
                                                                                                -updateAPIPerms -createClientSecret
```



Der HTML-Bericht des auf Entra ID (und eigentlich On-Prem-AD) spezialisierten Auditwerkzeugs Purple Knight sortiert seine Funde nach Kritikalität (Abb. 7).

Zahlreiche Geheimnisscanner helfen Programmierern und Admins dabei, sich dagegen zur Wehr zu setzen. Der wohl umfangreichste und bekannteste ist Trufflehog. Weil diese Werkzeuge viele Abhängigkeiten mitbringen, installiert man sie am einfachsten aus einem Docker-Image (Listing 7).

Hat man das Trüffelschwein wie im Listing angeschirrt, bringt man es mit folgendem Aufruf dazu, in einem GitHub-Repository samt komplettem Commitverlauf nach wertvollen Zugangsdaten zu schnüffeln.

```
$ trufflehog github --repo https://↵
 github.com/trufflesecurity/test_keys
```

Dasselbe funktioniert für Dateien oder Verzeichnisse, die nur lokal liegen:

```
$ cd && git clone https://github.↵
        com/trufflesecurity/test_keys
$ cd test_keys && trufflehog ↵
                    filesystem /pwd
```

Sogar für alle Ebenen eines Docker-Images:

```
$ trufflehog docker --image ↵
              trufflesecurity/secrets
```

Das Trüffelschwein ist intelligent und vielseitig: Die GitHub-Seite des Projekts dokumentiert, wie man nur nach verifizierten Schlüsseln sucht und Geheimnisse in S3- oder GCS-Buckets aufspürt. Geheimnisscanner lassen sich gut in eine CI/CD-Pipeline einbinden [4]. Weitere Vertreter solcher Scanwerkzeuge sind gitleaks, detect-secrets und Nosey Parker. Verwendet eine Organisation GitHub, lassen sich Funktionen zum automatischen Geheimnisscan kostenfrei dort direkt aktivieren – wie, verraten zwei Blogbeiträge (siehe ix.de/z3r7).

Ressourcen im Zaum halten mit Richtlinien

Cloud Custodian geht einen Schritt weiter, als nur Umgebungen danach zu bewerten, wie sicher sie konfiguriert sind – es setzt Richtlinien für Ressourcen durch. Damit lassen sich Vorschriften für Compliance, Sicherheit, Inventarisierung und Kosteneinsparung vorgeben. Sie werden in YAML definiert. Der Cloud-Verwalter unterstützt die drei großen Provider. Regeln für Kubernetes und OpenShift sind derzeit in Beta.

Listing 7: Installieren von Trufflehog

```
$ sudo apt update && sudo apt install -y docker.io
$ sudo systemctl enable docker –now
$ cat << 'EOF' >> ~/.zshrc
alias trufflehog='sudo docker run --rm -it -v "$PWD:/pwd" trufflesecurity/trufflehog:latest'
EOF
$ source ~/.zshrc
```

Übung macht den Meister. Will ein Admin neue Funktionen einer Cloud ausprobieren, Testdaten anlegen oder eines der vorgestellten Auditwerkzeuge erproben, verwendet er am besten eine eigene Testumgebung, denn Experimente in der Produktion sind generell eine schlechte Idee.

Mit dem Microsoft-365-Entwicklerprogramm spielt man kostenfrei in einer mit Beispieldaten vorkonfigurierten Sandbox der Microsoft-Anwendungscloud samt Verzeichnisdienst Entra ID. Mit dabei sind hochwertige E5-Lizenzen; optional schließt man ein Azure-Testabonnement ab, mit dem man den Infrastruktur- und Plattformteil der Microsoft-Wolke ausprobiert. AWS stellt ebenso ein kostenloses Nutzungskontingent bereit, gleichfalls GCP.

Noch realistischer wird eine solche Testumgebung, bringt man darin typische Fehlkonfigurationen ein. Das GitHub-Projekt „Awesome Cloud Security Labs“ listet absichtlich verwundbare Anwendungen für die drei großen CSP (sowie für verwandte Technologien wie Kubernetes und Terraform).

Weitere Quellen: OWASP und mehr

Das Open Web Application Security Project (OWASP) ist eine Non-Profit-Organisation, die sich auf die Fahnen geschrieben hat, die Sicherheit im Web zu verbessern. Sie stellt knapp neunzig Cheatsheets bereit: Eine unschätzbare Sammlung von kompakt aufgeschriebenem Wissen. An dieser Stelle relevant ist der Spickzettel zu „Cloud Architecture Security“, der gängige und notwendige Sicherheitsmuster erörtert, die beim Erstellen und Prüfen von Cloud-Architekturen zu beachten sind. Das Cloud Security Wiki sowie HackTricks Cloud enthalten nützliches Wissen für Sicherheitsverantwortliche und Pentester (alle siehe ix.de/z3r7).

Die Cloud Pentest Cheatsheets von Beau Bullock auf GitHub verzeichnen kompakt, wie man die großen Wolken aus offensiver Perspektive prüft. Auf dem Spickzettel zu anderen nützlichen Tools stehen Befehle, wie man die JSON-Berichte von ScoutSuite automatisiert durchsucht. Wer solche Programme nicht selbst installieren möchte, findet in RedCloud OS eine virtuelle Maschine für VMware auf Basis von Debian, die Werkzeuge für AWS, Azure, GCP und cloudübergreifend mitbringt.

Beau Bullock hält auch die Schulung „Breaching the Cloud“, die Pentester und andere Interessierte lehrt, wie man Unternehmensinfrastrukturen bei den drei großen CSP auditiert. Sie steht auf Abruf bereit bei Antisyphon Training, das preiswerte englischsprachige Lehrgänge für Informationssicherheitsexperten anbietet.

Das GitHub-Projekt Cloud OSINT enthält unter anderem Dorks (Suchabfragen) für Google und Shodan, mit denen man Ressourcen bei den Cloud-Service-Providern aufspürt.

Da besonders Datenspeicher wie S3-Buckets und Speicherkonten vor Fehlkonfigurationen strotzen und im Visier von Cyberkriminellen sind, hat Microsoft die herstellerneutrale Threat Matrix for Storage Services entwickelt. Sie hilft, strukturiert Bedrohungen für in Cloud-Speichern abgelegte Daten zu identifizieren und zu analysieren. Die Azure Threat Research Matrix und die AWS Threat Research Matrix beschreiben Details zu Taktiken und Techniken, mit denen Angreifer Unternehmen in diesen Clouds kompromittieren.

Mit Newslettern auf dem Laufenden bleiben

Die IT ist ohnehin schnelllebig und über den Wolken bläst der Wind noch schneller. Der wöchentliche Newsletter Cloud-SecList ist spezialisiert auf Cloud-Sicherheit und hilft, auf dem Laufenden zu bleiben, ohne überwältigt zu werden. Er informiert über herstellerspezifische Neuerungen, Blogartikel, aktuelle Angriffe und Werkzeuge. Zwar hat er einen Fokus auf AWS, deckt aber auch Azure, GCP und Technologien wie Kubernetes ab. Will man kein Abo, liest man über den Link „Past issues“ alle bisherigen über zweihundert Ausgaben. Auf der Übersicht der Ausgaben findet sich ein RSS-Feed.

Weil die Microsoft-Cloud in Organisationen weit verbreitet und Entra ID dabei so zentral ist, lohnt ebenfalls ein Abonnement der Entra.News. Wöchentlich stellt ein Microsoft-Mitarbeiter im Newsletter Nachrichten, Blogbeiträge und Videos vor, die mit der Entra-Produktfamilie zu tun haben.

Fazit

Da die IT-Ausgaben steigen, müssen Organisationen, die Geld sparen und die Cloud nutzen wollen, dafür sorgen, dass sie ihre Sicherheitsvorkehrungen darauf ausrichten. Der Einsatz kostenfreier Werkzeuge zum Selbstaudit öffentlicher Datenwolken hilft dabei.

Dieser Artikel beschließt die Tutorialserie. Sie hat gezeigt, wie man ohne externe Kosten und mit überschaubarem internen Aufwand die Angriffsfläche von extern erreichbaren Systemen und Webanwendungen, dem Unternehmensnetz und der öffentlichen Cloud verringert: Wie man sich im besten Sinne selbst hackt, bevor es andere tun. (ur@ix.de)

Quellen

[1] Frank Ully; Gefühlt sicher: Sicherheitsmythen und -irrtümer; iX 6/2023, S. 44

[2] Frank Ully; Ins Netz gezogen; Grundlagen von Azure Active Directory und Azure-Diensten; iX 4/2022, S. 44

[3] Frank Ully; Das Netz verstärkt; Azure Active Directory und Azure-Dienste absichern; iX 4/2022, S. 60

[4] Armin Berberovic; Schwachstellenscanner für Terraform-Skripte; iX 8/2023, S. 60

[5] Alle im Text erwähnten Werkzeuge, Dienste, Projekte und Artikel sind über ix.de/z3r7 zu finden.

FRANK ULLY

ist Head of Research der Oneconsult Deutschland AG in München. Er beschäftigt sich mit aktuellen Themen der offensiven IT-Sicherheit.