KAPITEL 2

Schutz und Management von Data Assets

Nachdem Sie nun in Kapitel 1 Informationen dazu erhalten haben, wo die Verantwortung Ihres Cloud-Providers endet und wo Ihre beginnt, besteht Ihr erster Schritt beim Absichern Ihrer Cloud-Umgebung darin, herauszufinden, wo sich Ihre Daten befinden (werden) und wie Sie sie schützen können. Es herrscht oft Verwirrung bezüglich des Begriffs »Asset Management«. Was sind genau unsere Assets, und was müssen wir tun, um sie zu managen? Die offensichtliche (und nicht hilfreiche) Antwort ist, dass Assets das sind, was Sie an Wert haben. Beginnen wir also, uns die Details anzuschauen.

In diesem Buch habe ich das Asset Management in zwei Teile aufgeteilt: *Data Asset Management* und *Cloud Asset Management*. *Data Assets* sind die wichtigen Informationen, die Sie haben, wie zum Beispiel Namen und Adressen von Kunden, Kreditkarteninformationen, Bankkonteninformationen oder Credentials für den Zugriff auf solche Daten. *Cloud Assets* sind die Dinge, die Sie haben, um Ihre Daten abzuspeichern und zu verarbeiten – Rechenressourcen wie Server oder Container, Storage wie Object Stores oder Block Storage und Plattforminstanzen wie Datenbanken oder Queues. Das Managen dieser Assets wird im nächsten Kapitel behandelt. Es ist zwar prinzipiell egal, ob Sie mit Data Assets oder Cloud Assets beginnen, da Sie sowieso immer wieder hin- und herspringen müssen, um das ganze Bild sehen zu können, aber ich finde es einfacher, mit Data Assets zu beginnen.

Die Theorie des Managens von Data Assets in der Cloud unterscheidet sich nicht von der On-Premises-Welt, aber in der Praxis gibt es einige Cloud-Technologien, die helfen können.

Identifizieren und Klassifizieren von Daten

Haben Sie wie im vorherigen Kapitel ein Diagramm und ein Threat Model zumindest auf einem Schmierzettel erstellt, haben Sie eine grobe Vorstellung davon, was Ihre wichtigen Daten sind, um welche Threat Actors Sie sich sorgen müssen und worauf diese aus sein könnten. Schauen wir uns unterschiedliche Wege an, wie Threat Actors Ihre Daten angreifen könnten.

Eines der beliebteren Modelle zur Informationssicherheit ist die *CIA Triade*: Vertraulichkeit (*Confidentiality*), Integrität (*Integrity*) und Verfügbarkeit (*Availability*). Ein Threat Actor, der versucht, die Vertraulichkeit Ihrer Daten zu verletzen, will sie stehlen – meist, um sie zu verkaufen oder Sie zu blamieren. Ein Threat Actor, der versucht, Ihre Datenintegrität zu stören, will Ihre Daten beeinflussen, zum Beispiel einen Kontostand verändern. (Das kann sogar effektiv sein, wenn beim Angriff keine Daten *gelesen* werden können – ich würde mich beispielsweise freuen, den gleichen Kontostand von Bill Gates zu haben, auch wenn ich den Betrag gar nicht kenne.) Ein Threat Actor, der versucht, Ihre Datenverfügbarkeit zu beeinflussen, will die Daten aus Spaß an der Freude oder für den Profit offline nehmen oder sie mit Ransomware verschlüsseln.[1]

Die meisten von uns haben nur begrenzte Ressourcen und müssen bei ihrem Vorgehen Prioritäten setzen.[2] Ein System zur Klassifikation von Daten kann dabei helfen, aber widerstehen Sie dem Drang, es komplizierter als absolut notwendig zu machen.

Beispiele für Stufen der Datenklassifikation

Jede Organisation ist anders, aber die folgenden Regeln bilden einen guten und einfachen Ausgangspunkt für das Einschätzen des Werts Ihrer Daten und damit des Risikos, dass unbefugt auf sie zugegriffen wird:

Niedrig oder öffentlich
: Egal ob die Informationen in dieser Kategorie für eine Veröffentlichung gedacht sind – wenn sie öffentlich gemacht würden, wären die Auswirkungen auf die Organisation sehr gering oder vernachlässigbar. Hier ein paar Beispiele:

  - Die öffentlichen IP-Adressen Ihrer Server.
  - Anwendungs-Log-Daten ohne persönliche Daten, Secrets oder Daten von Wert für angreifende Personen.
  - Softwareinstallationsmaterialien ohne Secrets oder andere für Angreifer wertvollen Elemente.

Mittel oder privat
: Diese Informationen sollten außerhalb der Organisation nicht ohne ein passendes Nondisclosure Agreement verfügbar sein. In viele Fällen (insbesondere in größeren Organisationen) sollte diese Art von Daten auch innerhalb der Organisation nur auf Need-to-know-Basis nutzbar sein. In vielen Unternehmen fällt der Großteil der Daten in diese Kategorie. Hier ein paar Beispiele:

  - Detaillierte Informationen dazu, wie Ihre Informationssysteme entworfen sind – nützliche Informationen für einen Angriff.
  - Informationen zu Ihren Mitarbeiterinnen und Mitarbeitern, die sich für Phishing- oder Pretexting-Angriffe nutzen lassen.

1 Ransomware ist gleichzeitig eine Verletzung der Verfügbarkeit und der Integrität, weil sie unautorisierte Veränderungen an Ihren Daten vornimmt, um sie nicht mehr verfügbar zu machen.

2 Wenn Sie unbegrenzte Ressourcen zur Verfügung haben, treten Sie bitte in Kontakt mit mir!

- Normale Finanzinformationen, wie zum Beispiel Bestellungen oder Reisekostenerstattungen, die genutzt werden könnten, um eventuell zu schlussfolgern, dass eine Übernahme ansteht.

Hoch oder vertraulich

Diese Informationen sind für die Organisation von großer Wichtigkeit, und eine Veröffentlichung kann signifikanten Schaden anrichten. Der Zugriff auf diese Daten sollte sehr strikt gesteuert werden, und es sollte mehrere Schutzebenen geben. In manchen Organisationen wird diese Art von Daten als »Crown Juwels« bezeichnet. Hier ein paar Beispiele:

- Informationen über zukünftige Strategien oder Finanzinformationen, die Wettbewerbern einen deutlichen Vorteil verschaffen würden.
- Geschäftsgeheimnisse, wie zum Beispiel das Rezept für Ihren beliebten Softdrink oder für frittierte Hähnchenteile.
- Geheimnisse, die als »Schlüssel zum Himmelsreich« dienen, wie zum Beispiel die Credentials für den umfassenden Zugriff auf Ihre Cloud-Infrastruktur.
- Kritische Informationen, die Sie zum Aufbewahren bekommen haben, wie zum Beispiel finanzielle Daten Ihrer Kunden.
- Andere Informationen, bei denen eine Datenleck Nachrichtenwert hätte.

Beachten Sie, dass Gesetze und Branchenregeln vorgeben können, wie Sie bestimmte Informationen klassifizieren. So enthält beispielsweise die Datenschutz-Grundverordnung der Europäischen Union (DSGVO) viele verschiedene Anforderungen für den Umgang mit persönlichen Daten, sodass Sie sich möglicherweise dazu entscheiden, alle persönlichen Daten als »moderates Risiko« zu klassifizieren und sie entsprechend zu schützen. Anforderungen des Payment Card Industry Data Security Standard (PCI DSS) würden wahrscheinlich vorgeben, dass Sie Daten zu Kreditkarten als »hohes Risiko« klassifizieren, wenn Sie diese in Ihrer Umgebung nutzen.

Beachten Sie auch, dass es Cloud-Services gibt, die Ihnen beim Klassifizieren und Schützen helfen können. So unterstützt Sie beispielsweise Amazon Macie (*https://oreil.ly/znsxp*) dabei, vertrauliche Daten in Amazon-S3-Buckets zu finden, Google Cloud Sensitive Data Prevention (*https://oreil.ly/MSzAr*) kann dabei helfen, bestimmte Arten vertraulicher Daten zu klassifizieren oder zu maskieren, und Microsoft Pureview (*https://oreil.ly/av897*) kann Daten auf Azure Cloud Services klassifizieren.

Was auch immer Sie für ein System zur Datenklassifikation nutzen – schreiben Sie eine Definition jeder Klassifikationsstufe und ein paar Beispiele auf und stellen Sie sicher, dass jeder, der Daten erzeugt, sammelt oder schützt, das Klassifikationssystem versteht.

Relevante Anforderungen aus Gesetzen oder aus Branchenvorgaben

Wie schon in der Einleitung erwähnt, ist dies ein Buch zu Sicherheit, nicht zu Compliance. Sehr vereinfacht gesagt, geht es bei Compliance darum, Ihre Sicherheit gegenüber einem Dritten zu beweisen – und das lässt sich viel einfacher erreichen, wenn Sie Ihre Systeme und Daten auch tatsächlich abgesichert haben. Die Informationen in diesem Buch werden Ihnen dabei helfen, sicher zu sein, es müssen jedoch zusätzliche Compliance-Aufgaben und Dokumentationen durchgeführt werden, nachdem Sie Ihre Systeme sicher gestaltet haben.

Manche Compliance-Anforderungen können aber Einfluss auf Ihr Sicherheitsdesign haben. Daher ist es schon in diesem frühen Stadium wichtig, ein paar Anforderungen aus Gesetzen und Branchenvorgaben zu kennen:

DSGVO (EU)
: Diese Verordnung kann auf die persönlichen Daten jedes Mitbürgers der Europäischen Union oder des Europäischen Wirtschaftsraums angewandt werden – egal wo sich diese Daten auf der Welt befinden. Die *Datenschutz-Grundverordnung* (DSGVO) fordert von Ihnen, Zugriff auf »alle Informationen über eine bestimmte oder bestimmbare natürliche Person ... die direkt oder indirekt identifiziert werden kann, insbesondere durch Zuordnung zu einer Kennnummer« zu katalogisieren, zu schützen und zu auditieren. Die Techniken in diesem Kapitel können Ihnen dabei helfen, einige der DSGVO-Anforderungen zu erfüllen, aber Sie müssen sicherstellen, dass überhaupt relevante persönliche Daten als Teil der zu schützenden Daten enthalten sind.

FISMA oder FedRAMP (USA)
: Der *Federal Information Security Management Act* wird pro Behörde angewandt, während die Zertifizierung für das *Federal Risk and Authorization Management Program* für mehrere Behörden genutzt werden kann. Bei beiden müssen Sie aber Ihre Daten und Systeme anhand des FIPS 199 (*https://oreil.ly/2BQRBJc*) und anderer US-Regierungsstandards klassifizieren. Sind Sie in einem Bereich tätig, in dem Sie eventuell eine dieser Zertifizierungen benötigen, sollten Sie die Klassifikationsstufen nach FIPS 199 nutzen.

ITAR (USA)
: Unterliegen Sie den *Internation Traffic in Arms Regulations*, werden Sie neben Ihren eigenen Schutzmaßnahmen auch Cloud-Services wählen müssen, die ITAR unterstützen. Solche Services sind bei manchen Cloud-Providern verfügbar, und sie werden nur von US-Personal gemanagt.

PCI DSS (global)
: Arbeiten Sie mit Kreditkarteninformationen, gibt der *Payment Card Industry Data Security Standard* vor, dass es bestimmte Schutzmaßnahmen gibt, die Sie umsetzen müssen, und dass bestimmte Arten von Daten nicht gespeichert werden dürfen.

HIPAA (USA)
: Agieren Sie in den USA und arbeiten mit geschützten Gesundheitsinformationen (*Protected Health Information*, PHI), legt der *Health Insurance Portability and Accountability Act* fest, dass Sie diese Informationen in Ihre Liste aufnehmen und sie schützen, was oft eine Verschlüsselung beinhaltet.

Es gibt viele weitere Vorschriften und Branchenanforderungen weltweit, wie zum Beispiel MTCS (Singapur), G-Cloud (UK) oder IRAP (Australien). Denken Sie, dass Sie einer davon unterliegen, schauen Sie sich die Arten von Daten an, die schützenswert sind, sodass Sie sicherstellen können, dass Sie sie angemessen katalogisieren und schützen.

Data Asset Management in der Cloud

Ein Großteil der eben aufgeführten Informationen ist allgemein sinnvoll und nicht für Cloud-Umgebungen spezifisch. Aber Cloud-Provider befinden sich in der einmaligen Situation, Ihnen beim Identifizieren und Klassifizieren Ihrer Daten helfen zu können. So werden sie Ihnen beispielsweise genau sagen können, wo Sie Ihre Daten ablegen, weil sie schließlich Geld dafür haben wollen!

Zudem sorgt der Einsatz von Cloud-Services aufgrund des Designs für einen gewissen Grad an Standardisierung. In vielen Fällen werden sich Ihre persistierten Daten in der Cloud in einem der Cloud-Services befinden, der Daten speichert, wie zum Beispiel Object Storage, File Storage, Block Storage, einer Cloud-Datenbank oder einer Cloud Message Queue, und nicht auf Tausende verschiedener Festplatten verstreut sein, die mit vielen verschiedenen physischen Servern verbunden sind.

Ihr Cloud-Provider bietet Ihnen die Tools, um diese Storage-Standorte zu inventarisieren und auf sie zuzugreifen (in einer sorgfältig kontrollierten Art und Weise), um herauszufinden, was für Arten von Daten dort gespeichert sind. Es gibt zudem Cloud-Services, die sich all Ihre Storage-Standorte anschauen und versuchen, automatisch zu klassifizieren, wo sich Ihre wichtigen Daten befinden. Sie können diese Informationen dann nutzen, um Ihre Cloud Assets zu taggen, die Daten speichern.

Identifizieren Sie Ihre wichtigen Daten und vergessen Sie Passwörter, API-Schlüssel und andere Secrets nicht, die genutzt werden können, um diese Daten auszulesen oder sie zu verändern! Wir werden über den besten Weg zum Absichern von Secrets in Kapitel 4 sprechen, aber zuerst einmal müssen Sie wissen, wo genau sie sich befinden.

Mit Blick auf unsere Beispielanwendung, die wir in Kapitel 1 in Diagrammen dargestellt haben, stellen wir fest, dass es offensichtlich Kundendaten in der Datenbank gibt. Aber wo haben Sie noch wichtige Assets? Hier ein paar wichtige Dinge, die Sie berücksichtigen sollten:

- Die Webserver haben Log-Daten, die eventuell genutzt werden können, um Ihre Kunden zu identifizieren.

- Ihr Webserver besitzt einen privaten Schlüssel für ein Transport-Layer-Security-(TLS-)Zertifikat – mit diesem und ein wenig Domain-Name-System-(DNS-) oder Border-Gateway-Protocol-(BGP-)Hijacking kann jeder vorgeben, Ihre Site zu sein, und die Passwörter Ihrer Kundinnen und Kunden stehlen (und einige Arten von zweiten Faktoren), wenn diese versuchen, sich anzumelden.
- Halten Sie eine Liste mit Passwort-Hashes vor, um Ihre Kunden zu verifizieren? Hoffentlich nutzen Sie ein Federated Identity System (beschrieben in Kapitel 4), wenn aber nicht, sind die Passwort-Hashes ein schönes Angriffsziel.[3]
- Ihr Anwendungsserver braucht ein Passwort oder einen API-Schlüssel, um auf die Datenbank zuzugreifen. Mit diesem Passwort kann man bei einem Angriff alles in der Datenbank lesen oder verändern, was auch die Anwendung kann.

Selbst in dieser wirklich einfachen Anwendung gibt es viele nicht offensichtliche Dinge, die Sie schützen müssen. Abbildung 2-1 entspricht Abbildung 1-6 aus dem vorherigen Kapitel, wobei dieses Mal noch die Data Assets hinzugefügt wurden.

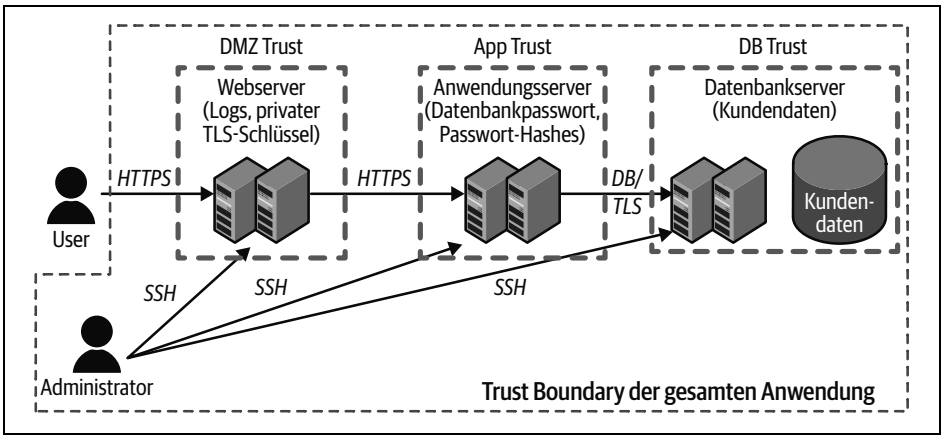

Abbildung 2-1: Diagramm der Beispielanwendung mit Data Assets

Cloud-Ressourcen taggen

Die meisten Cloud-Provider, aber auch Container Management Systems wie Kubernetes, kennen das Konzept von Tags. Ein *Tag* ist im Allgemeinen eine Kombination aus einem Namen (oder »Schlüssel«) und einem Wert. Diese Tags können für viele Zwecke genutzt werden, vom Kategorisieren von Ressourcen in einem Inventar über das Treffen von Zugriffsentscheidungen bis hin zum Auswählen, wer oder was alarmiert werden soll. So könnten Sie beispielsweise einen Schlüssel `PII-data` und ei-

3 Erinnern Sie sich an die 6,5 Millionen Passwort-Hashes von LinkedIn, die gestohlen, offline entschlüsselt und dann verwendet wurden, um auf andere Konten zuzugreifen, bei denen die User das gleiche Passwort eingesetzt haben? So etwas ist immer wieder passiert, und Sites wie »have i been pwned« (*https://haveibeenpwned.com*) können Ihnen all die Datendiebstähle aufführen, die eventuell Ihre E-Mail- und Passwortdaten enthalten.

nen Wert `yes` für alles nutzen, das identifizierbare persönliche Daten enthält, oder Sie verwenden einen Schlüssel `datatype` und einen Wert `PII`.

Das Problem ist klar: Wenn in Ihrer Organisation alle unterschiedliche Tags nutzen, werden diese nicht allzu hilfreich sein! Legen Sie eine Richtlinie für den Einsatz von Tags fest. Um diese Richtlinie zu unterstützen, erstellen Sie einen Tag-Standard mit einer Liste von Tags und Erläuterungen für ihren Einsatz. Dann nutzen Sie die gleichen Tags bei mehreren Cloud-Providern und lassen sie automatisch von Tools erzeugen, die auch die Cloud-Ressourcen anlegen.

In kleineren Organisationen wird ein einfacher Tag-Standard vermutlich reichen. In größeren Organisationen sollte dieser Tag-Standard eher als versionierter, abwärtskompatibler Standard behandelt werden – mit einem eindeutigen Verantwortlichen und regelmäßigen Reviews. Manche Tags werden sicherlich unternehmensweit zum Einsatz kommen, während andere nur für einzelne Bereiche der Firma relevant sind. Selbst wenn einer Ihrer Cloud-Provider nicht explizit den Einsatz von Tags unterstützt, gibt es oft andere Beschreibungsfelder, die Tags in einem leicht zu parsenden Format wie JSON aufnehmen können. Tags verursachen nur selten Probleme, nutzen Sie sie daher ausgiebig – wenn Sie sie nicht brauchen, lassen sie sich leicht ignorieren.

Tags können kostenlos genutzt werden, daher gibt es nur selten technisch begründete Bedenken, wenn sehr viele zum Einsatz kommen. Sie sollten aber darauf achten, den Tag-Standard nicht so kompliziert zu machen, dass er die Menschen verwirrt, die Regeln zum Anwenden und Konsumieren von Tags schreiben müssen. Zudem haben Cloud-Provider Grenzen für die Menge an Tags bei einer bestimmten Ressource (meist zwischen 15 und 64 Tags pro Ressource).

Manche Cloud-Provider bieten Automatismen an, um zu prüfen, ob Tags korrekt den Ressourcen zugewiesen wurden, sodass Sie ungetaggte oder falsch getaggte Ressourcen leicht finden und korrigieren können. Gibt es bei Ihnen beispielsweise die Regel, dass jedes Asset mit der maximalen Datenklassifikation getaggt sein muss, die dort erlaubt ist, können Sie automatische Scans laufen lassen, um Ressourcen zu finden, bei denen das Tag fehlt oder bei denen der Wert keiner der Klassifikationsstufen entspricht, auf die Sie sich geeinigt haben.

In Tabelle 2-1 sehen Sie die Namen, die Tags bei den verschiedenen Cloud-Providern tragen. Kubernetes, das on-premises oder auf einem beliebigen IaaS-Provider laufen kann, nutzt den Begriff *Label*.

Tabelle 2-1: Tagging-Features

Infrastruktur	Feature-Name
Amazon Web Services	Tags
Microsoft Azure	Tags
Google Cloud Platform	Label und Network Tags
IBM Cloud	Tags

Mehr über das Taggen von Ressourcen werden wir in Kapitel 3 erfahren, aber jetzt sollten Sie sich erst einmal ein paar Tags überlegen, die sich auf Daten beziehen und die sich auf Ihre verschiedenen Cloud-Ressourcen anwenden lassen, zum Beispiel `dataclass:low`, `dataclass:moderate`, `dataclass:high` und `regulatory:gdpr`.

Daten in der Cloud schützen

Diverse der in diesem Abschnitt vorgestellten Techniken zum Schützen von Daten lassen sich auch on-premises anwenden, aber viele Cloud-Provider bieten einfache, standardisierte und weniger teure Wege an, Ihre Daten zu schützen.

Tokenisieren

Warum sollten Sie die Daten speichern, wenn Sie etwas speichern können, das genauso funktioniert, für Angreifende aber nutzlos ist? Beim *Tokenisieren*, das am häufigsten bei Kreditkartennummern zum Einsatz kommt, wird ein Teil der kritischen Daten durch ein Token ersetzt (meist zufällig generiert). Das Token besitzt im Allgemeinen die gleichen Charakteristiken wie die ursprünglichen Daten (zum Beispiel 16 Ziffern lang zu sein), sodass Folgesysteme, die dafür eingerichtet sind, diese Daten zu nutzen, nicht angepasst werden müssen. Nur eine Stelle (ein »Token Service«) kennt die tatsächlichen kritischen Daten. Das Tokenisieren kann allein oder zusammen mit Verschlüsselung genutzt werden (siehe den folgenden Abschnitt).

Es gibt beispielsweise Cloud-Services, die mit Ihrem Browser zusammenarbeiten, um kritische Daten zu tokenisieren, bevor diese verschickt werden, oder welche, die zwischen dem Browser und der Anwendung sitzen, um vertrauliche Daten zu tokenisieren, bevor sie die Anwendung erreichen.

Verschlüsselung

Verschlüsselung ist das Allheilmittel in der Welt des Datenschutzes – wir wollen »alles verschlüsseln«. Leider ist das Ganze doch ein bisschen komplizierter. Es gibt drei Arten von Daten, die Sie eventuell verschlüsseln müssen:

- Daten, die transferiert werden (über ein Netzwerk)
- Confidential Computing oder Daten im Einsatz (aktuell in der CPU des Computers verarbeitet oder im Speicher vorgehalten)
- abgelegte Daten (auf persistentem Storage, etwa einer Festplatte)

Das Verschlüsseln von Daten, die transferiert werden, behandeln wir genauer in Kapitel 6. In diesem Abschnitt werden wir auf die beiden anderen Einsatzgebiete von Verschlüsselung eingehen.

Haben Sie einmal einen bestimmten Punkt erreicht, sind gar nicht immer mehr Bits notwendig oder nützlich. Verschlüsselung wird oft durch einen Fehler in der Implementierung geknackt und nicht durch Brute Force. Zudem gibt es häufig Performancenachteile, wenn ein Verschlüsselungsalgorithmus mit mehr Bits zum Einsatz kommt, insbesondere dann, wenn Sie keine Hardwarebeschleunigung verwenden können. Wollen Sie daraus keine Wissenschaft machen, ist es im Allgemeinen sicher genug, sich an den Verschlüsselungsanforderungen großer privater oder staatlicher Organisationen zu orientieren, die das Thema ausgiebig analysiert haben.

Confidential Computing

Das Verschlüsseln von aktuell genutzten Daten ist mittlerweile bei vielen Cloud-Providern möglich und wird typischerweise Organisationen mit sehr vertraulichen Daten als *Confidential Computing* angeboten. Weil es die Art und Weise des Zugriffs des Prozessors auf den Speicher verändert, muss es durch die Hardwareplattform unterstützt und dann auch vom Cloud-Provider angeboten werden.

Meist wird in der Cloud der Speicher des Prozesses oder der virtuellen Maschine verschlüsselt, sodass er selbst von einem privilegierten User (oder einer angreifenden Person oder Malware, die sich als solcher ausgibt) nicht gelesen werden kann. Der Prozessor selbst kann ihn nur lesen, wenn er Code für einen bestimmten Prozess oder eine bestimmte virtuelle Maschine ausführt.[4] Agieren Sie in einer Hochsicherheitsumgebung und beinhaltet Ihr Threat-Modell auch das Schützen von Daten im Speicher vor einem privilegierten User oder brauchen Sie eine zusätzliche Isolierung zwischen Ihnen und einem anderen in der Cloud, sollten Sie sich nach einer Plattform umschauen, die Speicherverschlüsselung unterstützt. Stichwörter sind hier häufig hardwarespezifische Markennamen wie Intel SGX/TGX, AMD SEV oder IBM Z Pervasive Encryption.

Verschlüsseln von gespeicherten Daten

Die Verschlüsselung von gespeicherten Daten kann am kompliziertesten zu implementieren sein. Das Problem ist nicht das Verschlüsseln der Daten – dafür gibt es viele Bibliotheken. Das Problem ist, dass Sie nach dem Verschlüsseln einen Schlüssel haben, mit dem Sie darauf zugreifen können. Wo legen viele Menschen diesen ab? Ganz klar – direkt bei den Daten! Stellen Sie sich vor, Sie schließen eine Tür ab und hängen den Schlüssel an einem Haken daneben auf – freundlicherweise auch noch mit einem »Schlüssel«-Schildchen daneben. Um wirkliche Sicherheit zu erreichen (und nicht nur einen Haken neben »Daten verschlüsseln« zu machen), müssen Sie sich um ein ordentliches Key Management kümmern. Zum Glück gibt es Cloud-Services, die dabei helfen können.

4 Beachten Sie, dass eine In-Memory-Verschlüsselung die Daten nur vor Angriffen von außerhalb des Prozesses schützt – schaffen Sie es, den Prozess selbst dazu zu bringen, etwas zu tun, das er nicht tun sollte, kann dieser den Speicher lesen und die Daten preisgeben.

Verschlüsselte Daten können nicht effektiv komprimiert oder dedupliziert werden. Wollen Sie Kompression oder Deduplikation nutzen, geschieht das vor dem Verschlüsseln.

In klassischen On-Premises-Umgebungen mit hohen Sicherheitsanforderungen würden Sie ein *Hardware Security Module* (HSM) kaufen, in dem Ihre Schlüssel aufbewahrt werden – meist in Form einer Erweiterungskarte oder eines Moduls, das über das Netzwerk angesprochen wird. Ein HSM besitzt signifikanten logischen und physischen Schutz gegen unautorisierten Zugriff. Bei den meisten Systemen kann jeder mit physischem Zugriff versuchen, an die Schlüssel zu kommen, aber ein HSM besitzt Sensoren, um die Daten zu löschen, sobald jemand versucht, es zu entfernen, mit Röntgenstrahlen zu durchleuchten, an der Stromversorgung herumzupfuschen oder auch nur bedrohlich in seine Richtung zu schauen.

HSMs sind teuer und daher für die meisten On-Premises-Umgebungen nicht praktikabel. In Cloud-Umgebungen hingegen sind ausgefeilte Technologien wie HSMs und Key Management Systems durchaus auch für Projekte mit kleinem Budget realistisch.

Manche Cloud-Provider bieten an, ein dediziertes HSM für Ihre Umgebung zu mieten. Das kann für Umgebungen sinnvoll sein, die höchste Sicherheit verlangen, aber es ist auch in einer Cloud-Umgebung teuer und oft nur schwierig automatisch einzurichten. Eine andere gute Option ist ein *Key Management Service* (KMS), der vom Cloud-Provider betrieben wird und im Allgemeinen im Backend ein HSM nutzt, um die Schlüssel sicher zu verwahren. Ein KMS ist meist ein Multitenant-Service, was die Angriffsoberfläche ein klein wenig vergrößert und erfordert, dass Sie sowohl dem HSM als auch dem KMS vertrauen müssen (statt nur dem HSM), sodass das Risiko ein bisschen größer wird. Aber verglichen mit dem Einrichten und Betreiben eines eigenen Key Management – was oft nicht ordentlich geschieht – bietet ein KMS ausgezeichnete Sicherheit bei sehr geringen Kosten. Sie können die Vorteile eines korrekten Key Management auch in Projekten nutzen, die nur ein geringes Sicherheitsbudget besitzen.

In Tabelle 2-2 finden Sie die aktuell verfügbaren Angebote der großen Cloud-Provider für das Key Management.

Tabelle 2-2: Angebote zum Key Management

Provider	Dedizierte HSM-Option	Key Management System
Amazon Web Services	Cloud HSM	Amazon KMS
Microsoft Azure	Azure Dedicated DSM	Key Vault
Google Compute Platform	Cloud HSM	Cloud KMS
IBM Cloud	Cloud HSM	Key Protect

Wie nutzen Sie aber ein KMS richtig? Nun, da wird es doch ein wenig kompliziert.

Key Management Der einfachste Ansatz beim Key Management ist, einen Schlüssel zu erzeugen, die Daten damit zu verschlüsseln, den Schlüssel in das KMS zu stecken und dann die verschlüsselten Daten auf die Festplatte zu schreiben – zusammen mit einem Hinweis, welcher Schlüssel genutzt wurde. Bei diesem Vorgehen gibt es aber zwei größere Probleme:

1. Das KMS bekommt viel Last ab. Es gibt gute Gründe dafür, für jede Datei einen anderen Schlüssel zu verwenden, daher müsste ein KMS mit vielen Kunden Milliarden oder Billionen von Schlüsseln speichern, auf die aber nahezu sofort zugegriffen werden können sollte.
2. Wollen Sie die Daten sicher löschen, müssen Sie dem KMS so weit trauen, dass es den Schlüssel ebenfalls unwiderruflich löscht, wenn Sie fertig sind, und nicht irgendwo Backup-Kopien herumliegen lässt. Alternativ müssen Sie alle verschlüsselten Daten überschreiben,[5] was eine Weile dauern kann.

Sie wollen vermutlich nicht Stunden oder Tage warten, bis Ihre Daten überschrieben sind. Es ist besser, die Möglichkeit zu haben, Datenobjekte schnell und sicher auf zwei verschiedenen Wegen zu löschen: entweder durch das Löschen eines Schlüssels im KMS, wodurch effektiv viele verschiedene Objekte auf einmal gelöscht werden können, oder durch das Löschen eines Schlüssels für die eigentlich gespeicherten Daten, um ein einzelnes Datenobjekt zu löschen. Aus diesem Grund haben Sie meist Schlüssel auf zwei Ebenen: einen *Key Encryption Key* (KEK) und einen *Data Encryption Key* (DEK). Wie die Namen schon andeuten, wird der Key Encryption Key genutzt, um die Data Encryption Keys zu verschlüsseln (oder zu »wrappen«), während die Data Encryption Keys direkt bei den Daten abgelegt werden. Der Key Encryption Key lebt im Allgemeinen im KMS und kommt dort aus Sicherheitsgründen nie heraus. Die gewrappten Data Encryption Keys werden bei Bedarf zum Entschlüsseln an das HSM gesendet, danach kann man mit ihnen die Daten verschlüsseln oder entschlüsseln. Sie schreiben die entwrappten Schlüssel nie auf. Sind Sie mit der aktuellen Ver- oder Entschlüsselungsoperation fertig, vergessen Sie sie wieder.[6]

Der Einsatz von Schlüsseln lässt sich mit einer Analogie aus der physischen Welt leichter verstehen. Stellen Sie sich vor, Sie verkaufen Ihr Haus (das all Ihre Daten enthält) und geben dem Makler einen Schlüssel, damit dieser die Tür aufschließen kann. Dieser Haustürschlüssel ist wie ein Data Encryption Key – er lässt sich nutzen, um direkt auf Ihr Haus (Ihre Daten) zuzugreifen. Der Makler wird diesen Schlüssel in einer Box an Ihrer Tür unterbringen und diese mit einem Code absichern, der vom Maklerservice bereitgestellt wird. Dieser Code ist vergleichbar mit dem Key Encryption Key, und der Maklerservice, der die Codes herausgibt, entspricht dem Key

5 Trotz der Erkenntnisse eines bekannten USENIX-Artikels (*https://oreil.ly/FSbkW*) von Peter Gutmann aus dem Jahr 1996, in dem die Möglichkeit untersucht wurde, Daten von einer Festplatte zu lesen, die überschrieben wurden, ist das heutzutage nicht praktikabel. Das Wiederherstellen überschriebener Daten von Solid State Drives (SSDs) ist aufgrund der Art und Weise des Schreibens von Daten ein bisschen leichter, aber die meisten SSDs besitzen ein »Secure Erase«-Feature, um das gesamte Laufwerk zu säubern – in einem USENIX-Artikel (*https://oreil.ly/ec5Hp*) von Michael Wei et al. aus dem Jahr 2011 finden Sie mehr dazu.

6 Das ist eine extrem vereinfachte Darstellung. Umfassend werden all diese kryptografischen Dinge in Bruce Schneiers Buch *Angewandte Kryptographie* (Pearson Studium 2006) behandelt.

Management Service. In dieser etwas überdehnten Analogie nehmen Sie die Schlüsselbox mit zum KMS und erhalten eine Kopie des darin befindlichen Schlüssels, wenn Sie versprechen, dass Sie keine Kopie davon machen (ihn auf die Festplatte schreiben) und ihn einschmelzen (vergessen), wenn Sie fertig sind. Sie sehen niemals den eigentlichen Code, der die Box öffnet.

Wenn Sie zum Haus (den Daten) gehen, wissen Sie im Endergebnis, dass sich der Data Key direkt vor Ihrer Nase befindet, sich aber nicht ohne einen anderen Schlüssel beziehungsweise ein Passwort öffnen lässt. In der realen Welt würden natürlich ein Hammer und ein wenig Zeit zum Schlüssel führen oder Ihnen dazu dienen, ein Fenster einzuschlagen und ohne Schlüssel direkt an die Daten zu gelangen. Das kryptografische Äquivalent zum Hammer ist das Raten des Schlüssels oder Passworts, mit dem der Data Key gesichert ist. Das geschieht üblicherweise durch Ausprobieren aller Möglichkeiten (Brute Force) oder – für Schlüssel, die auf Passwörtern basieren – durch Ausprobieren vieler gebräuchlicher Passwörter (ein Dictionary-Angriff). Sind der Verschlüsselungsalgorithmus und die Implementierung dieses Algorithmus korrekt, ist die zu erwartende Dauer für den »Hammer«, um in die Box zu gelangen, deutlich größer als die erwartete Restlebensdauer des Universums.

Serverseitige und clientseitige Verschlüsselung Ausgezeichnete Neuigkeiten: Sie müssen einen Großteil dieses Key Management gar nicht selbst übernehmen! Für die meisten Cloud-Provider gilt: Nutzen Sie deren Storage und deren KMS und schalten Sie KMS-Verschlüsselung für Ihre Storage-Instanzen ein, wird der Storage-Service automatisch Data Encryption Keys erzeugen, sie mit einem Key Encryption Key verpacken, den Sie im KMS managen können, und die gewrappten Schlüssel zusammen mit den Daten ablegen. Sie können die Schlüssel weiterhin im KMS managen, aber Sie müssen es nicht bitten, sie zu ver- oder entpacken, und Sie müssen die Verschlüsselung oder Entschlüsselung nicht selbst durchführen. Manche Provider nennen das *serverseitige Verschlüsselung*.

Weil der Multitenant-Storage-Service die Möglichkeit besitzt, Ihre Daten zu entschlüsseln, könnte ein Fehler in diesem Storage-Service einem unautorisierten User potenziell ermöglichen, den Storage-Service zu bitten, Ihre Daten zu entschlüsseln. Aus diesem Grund ist das Ver- und Entschlüsseln durch den Storage-Service nicht *ganz* so sicher wie ein Entschlüsseln in Ihrer eigenen Instanz – wenn Sie sie korrekt implementieren und dabei bekannte Bibliotheken und Prozesse nutzen. Das Verschlüsseln und Entschlüsseln in Ihrer eigenen Anwendung wird oft als *clientseitige Verschlüsselung* bezeichnet. Sofern Sie aber keine sehr geringe Risikotoleranz besitzen, empfehle ich Ihnen, die gut getesteten Cloud-Services zu nutzen und diese das Ver- und Entschlüsseln für Sie übernehmen zu lassen.

Beachten Sie, dass der Server beim Einsatz von clientseitiger Verschlüsselung keine Möglichkeit hat, die verschlüsselten Daten zu lesen, weil er die Schlüssel nicht kennt. Das bedeutet keine Suche auf Serverseite, keine Berechnungen, kein Indexieren, keine Malware-Scans oder andere hochwertige Aufgaben, die dort durchgeführt werden können. Homomorphe Verschlüsselung ermöglicht es, dass Operationen

wie z.B. Additionen mit verschlüsselten Daten korrekt durchgeführt werden können, ohne die Daten zu entschlüsseln, aber aktuell ist das Ganze noch zu langsam, um praktikabel zu sein.

Sofern Sie nicht einen Großteil Ihrer bisherigen Karriere der Kryptografie gewidmet haben, sollten Sie nicht versuchen, Ihr eigenes Kryptosystem zu erstellen oder zu implementieren. Selbst wenn Sie das Verschlüsseln und Entschlüsseln in Ihrer eigenen Anwendung realisieren, sollten Sie nur gut getestete und ausreichend supportete Bibliotheksimplementierungen sicherer Algorithmen einsetzen.

Gibt es für Ihre Organisation keine Liste genehmigter kryptografischer Algorithmen, ist eine gute Quelle für empfohlene Algorithmen die NIST SP 800-131A (*https://oreil.ly/ABAjn*).

Kryptografisches Löschen Es ist tatsächlich schwierig, große Datenmengen zuverlässig zu löschen.[7] Es dauert lange, die Daten vollständig zu überschreiben, und selbst dann kann es sein, dass noch irgendwo an anderer Stelle Kopien existieren. Wir können das durch *kryptografisches Löschen* lösen. Dabei speichern wir keine Daten im Klartext auf der Festplatte, sondern nur eine verschlüsselte Version. Wollen wir dann, dass die Daten nicht wiederherstellbar sind, können wir den Zugriff auf den Key Encryption Key im KMS löschen oder zurückziehen, sodass alle Data Encryption Keys, die mit diesem Schlüssel »verpackt« wurden, nutzlos werden – egal wo sie sich in der Welt befinden. Wir können auch nur bestimmte Datenelemente löschen, indem wir den verpackten Data Encryption Key beseitigen, sodass eine Datei mit mehreren Terabytes sehr effektiv durch das Überschreiben eines 256-Bit-Schlüssels nicht mehr wiederherstellbar ist.

Wie Verschlüsselung unterschiedliche Arten von Angriffen vereitelt

Eine Verschlüsselung von gespeicherten Daten kann also für einen gewissen Schutz sorgen, weil bei Angriffen die Optionen reduziert werden. Die Daten sind nur an wenigen Orten im Klartext lesbar – abhängig davon, wo die Verschlüsselung im System vorgenommen wird. Schauen wir uns eine einfache Beispielanwendung mit einer Datenbank an, um zu sehen, wie unsere Verschlüsselungsentscheidungen uns schützen. Die relevanten Layer der Anwendung sind:

1. Das Storage-System, zu dem die Festplatten gehören – dabei werden die Daten eventuell schon vor dem Speichern auf den Platten verschlüsselt.
2. Das Angebot der »Database-as-a-Service«-Plattform, das die Daten eventuell verschlüsselt, bevor sie an das Storage-System geschickt werden.
3. Die Anwendung, die eventuell schon Daten verschlüsselt, bevor diese an die Datenbank geschickt werden.

Schauen wir uns die Vor- und Nachteile sowie die verbleibenden Risiken des Verschlüsselns auf diesen unterschiedlichen Layern an.

7 Auch wenn es paradoxerweise ganz einfach geht, wenn man es unabsichtlich macht!

Verschlüsselung auf den Festplatten Bei einem Angriff könnte versucht werden, Festplatten aus dem Data Center oder aus einem Müllcontainer zu stehlen oder Bänder auf dem Transport zu entwenden. Verschlüsselt das Storage-Subsystem Daten, bevor es sie auf die Festplatte speichert, können die Daten bei diesen Angriffen nicht genutzt werden, selbst wenn ein physischer Zugriff auf die Platten oder Bänder möglich ist. Dies ist dann lediglich ein Zugriff auf einen unverständlichen Haufen Bits, und die Schlüssel zum Entschlüsseln der Daten sind sicher an einem anderen Ort gespeichert – im Storage-Subsystem!

Früher gab es Performancenachteile durch das Verschlüsseln von Daten, die auf die Festplatte geschickt werden, aber dank der Beschleunigung durch kryptografische Hardware ist das meist kein Problem mehr – Cloud-Provider verschlüsseln routinemäßig so gut wie alle Daten, die auf Festplatten gespeichert werden, außer in wenigen Bare-Metal-Situationen, in denen Sie sich selbst um die Festplatten kümmern. Es gibt also nur sehr wenige Nachteile bei der Verschlüsselung auf Festplattenebene in einer Cloud-Umgebung, und vermutlich wird das sowieso schon für Sie erledigt.

Das ist großartig, aber gestohlene oder verloren gegangene Medien sind in Cloud-Umgebungen angesichts der Zugangskontrollen und der Behandlung von ausgebautem Equipment meist sowieso kein großes Problem. (Verschlüsselung auf Festplattenebene ist für portable Geräte wie Smartphones und Laptops viel wichtiger, weil hier Geräte viel eher gestohlen werden oder abhandenkommen und die Dekommissionierungsprozesse oft nicht so ausgereift sind.)

Verschlüsselung lediglich um des Erfüllens einer Checkliste willen hilft oft nur dabei, die Bedrohung durch physischen Diebstahl zu verringern – und manchmal nicht einmal das, weil dieser Schutz keiner ist, wenn Sie die unverpackten Schlüssel auf denselben Medien wie die Daten unterbringen.

Aber was, wenn es Angreifende schaffen, sich als Administrator des Storage-Systems auszugeben, in dem die Festplatten arbeiten? Da das Storage-Subsystem die Entschlüsselung ausführt, werden Angriffe auf dieser Ebene (oder noch weiter oben) Zugriff auf die unverschlüsselten Daten erhalten können.

Verschlüsselung auf der Plattform Wie wäre es stattdessen, die Datenbank (oder einen anderen Service) die Daten verschlüsseln zu lassen, bevor sie an das Storage-Subsystem geschickt werden? In diesem Fall wird jeder, der Zugriff auf das Storage-Subsystem hat, nur einen Haufen Bits zu Gesicht bekommen – diese Daten könnten zerstört werden oder eine Zeit lang nicht verfügbar sein, sie dürften aber nicht lesbar oder gar veränderbar sein.

Der Nachteil beim Verschlüsseln auf dieser Ebene ist oft etwas größer. Weil die Datenbank die Daten vor dem Ablegen im Storage verschlüsseln, können sie vom Storage-Subsystem nicht komprimiert oder dedupliziert werden, was wiederum bedeutet, dass die Storage-Kosten höher sein können. Abhängig von der Datenbank-Engine kann es auch Nachteile bei der Performance geben.

Die Verschlüsselung auf Datenbankebene kann Sie zwar davor schützen, dass auf den Ebenen darunter etwas falsch läuft, aber es gibt immer noch ein gewisses Risiko.

Die Datenbank hat User, denen es erlaubt ist, die Daten zu sehen, und der Datenbankservice hat Administratoren, die eventuell dazu in der Lage sind, auf jede Datenbank zuzugreifen. Cloud-Provider besitzen oft mehrere Schutzebenen, um zu verhindern, dass deren Administratoren Kundendaten lesen oder darin herumpfuschen können, daher ist das ein recht geringes Risiko. Das größere Risiko ist, dass ein Angreifer, der Zugriff auf den API-Schlüssel erhält, mit dem Ihre Anwendung mit der Datenbank spricht, alles in der Datenbank lesen und schreiben kann, was auch Ihre Anwendung kann!

Verschlüsselung auf Anwendungsebene Jetzt sind wir im Stack ganz oben angekommen. Verschlüsselt Ihre Anwendung Daten, bevor sie sie an die Datenbank schickt, können Menschen mit Zugriff auf die Datenbank nichts damit anfangen – es sei denn, sie erhalten Zugriff auf die Anwendung oder können deren Schlüssel abgreifen.

Sie verlieren vielleicht ein bisschen Performance oder Kosteneffektivität, wenn Sie auf Datenbankebene verschlüsseln, aber wenn Sie das auf Anwendungsebene machen, ist dies deutlich mehr spürbar. Weil die Datenbank die unverschlüsselten Daten nicht zu sehen bekommt, kann sie nicht nach bestimmten Datenobjekten suchen, sie sortieren, Berichte erstellen oder anderes damit tun. Die Anwendung muss diese Dinge selbst umsetzen (oder ohne sie zurechtkommen), was deutliche Auswirkungen auf Performance oder Funktionalität haben kann. Ich empfehle die Implementierung einer Verschlüsselung auf Anwendungsebene nur bei besonders kritischen Daten, die Ihre Anwendung verarbeitet – lassen Sie die Ebenen darunter sich um alles andere kümmern.

Erhält ein Angreifer bei einem Angriff unautorisierten Zugriff auf die Anwendung, ist aus Sicht der Verschlüsselung alles verloren, weil die Anwendung die Daten lesen können muss, damit sie funktioniert. Defense-in-Depth-Techniken können dagegen helfen. So dienen beispielsweise getrennte Data Stores, die durch *Access Control Lists* (ACLs) geschützt sind, und getrennte Kryptografieschlüssel für unterschiedliche Anwendungen dazu, die Auswirkungen solch eines Einbruchs zu reduzieren, weil dann der Angreifer nur auf das zugreifen kann, auf das auch die kompromittierte Applikation Zugriff hat – aber auf nichts anderes.

Quantensichere Kryptografie

Es wird davon ausgegangen, dass Quantencomputer bei manchen Aufgaben viel besser als klassische Computer sind, und manche dieser Aufgaben haben Auswirkungen auf die Sicherheit von Verschlüsselungen. Ein bekanntes Beispiel: Wenn Sie große Zahlen schnell in ihre Faktoren zerlegen können, können Sie einen wichtigen kryptografischen Algorithmus knacken.

Auch wenn Quantencomputer diese Angriffe noch nicht durchführen können, besteht eines der Risiken darin, dass bei Angriffen verschlüsselte Daten gesammelt werden, um sie dann später zu entschlüsseln. Aus diesem Grund gibt es einen branchenweiten Schub, jetzt schon zu quantensicheren Algorithmen zu wechseln, bevor diese Angriffe Wirklichkeit werden können.

Algorithmen, die zum Verschlüsseln von Daten auf dem Transportweg zum Einsatz kommen, sind am meisten gefährdet, und zukünftige Versionen von TLS werden vermutlich quantensichere Algorithmen nutzen. Während das Verschlüsseln von gespeicherten Daten mit AES-256 auf absehbare Zukunft als ausreichend angesehen wird, sei darauf hingewiesen, dass viele Schemata einen nicht quantensicheren Algorithmus nutzen, um den symmetrischen AES-Schlüssel zu verschlüsseln. Das geschieht, um den Schlüssel vielen verschiedenen Individuen zur Verfügung stellen zu können, ohne mehrere Kopien der Daten mit unterschiedlichen AES-Schlüsseln verschlüsseln zu müssen. Aber jedes Produkt, das dies nutzt, wird angepasst werden müssen, um die AES-Schlüssel mit einem quantensicheren Algorithmus neu zu verschlüsseln – ansonsten werden die gespeicherten Daten ebenfalls kompromittiert werden können.

Wenn Sie sich weitergehend mit diesem Thema befassen wollen, empfehle ich als (noch in der Entwicklung befindliche) Ressource NIST SP 1800-38 (*https://oreil.ly/z3KTj*).

Das ist ein weiteres Beispiel, bei dem Defense in Depth helfen kann. Designen Sie das System so, dass Sie verschlüsselte Versionen Ihrer Daten ohne die Schlüssel veröffentlichen könnten, ohne dass sie jemand theoretisch lesen kann. Schützen Sie die verschlüsselten Daten zudem wo immer möglich, sodass sie auch dann noch sicher sind, wenn diese Annahme nicht mehr zutrifft.

Zusammenfassung

Wenn Sie Ihre Cloud-Strategie planen, müssen Sie sich überlegen, was für Daten Sie haben. Klassifizieren Sie jeden Datentyp anhand der Auswirkungen, die durch das Lesen, Verändern oder Löschen durch Angreifer entstehen könnten. Finden Sie in einem »Tag-Standard« organisationsweit eine Übereinkunft darüber, welche Tags verwendet werden sollen, und nutzen Sie die Tagging-Features Ihres Cloud-Providers, um Ressourcen auszuzeichnen, die Daten enthalten.

Sie sollten sich wenn möglich für eine Verschlüsselungsstrategie entscheiden, bevor Sie Cloud-Ressourcen erstellen, in denen Daten gespeichert werden, weil es schwierig sein kann, sie später noch zu ändern. In den meisten Fällen sollten Sie das Key Management System Ihres Cloud-Providers einsetzen, um die kryptografischen Schlüssel zu verwalten, und auf die in den Datenbank- und Storage-Services vorhandenen Verschlüsselungsmöglichkeiten zurückgreifen. Haben Sie wirklich kritische Informationen, denken Sie darüber nach, sie selbst in Ihrer Anwendung zu verschlüsseln, bevor Sie sie speichern, und dabei nur gut getestete Implementierungen sicherer Algorithmen zu nutzen.

Kontrollieren Sie die User und Systeme mit Zugriff auf die Schlüssel sorgfältig und richten Sie Benachrichtigungen ein, wenn die Schlüssel auf ungewöhnliche Art und Weise genutzt werden. Verwenden Sie ein Key Management System, erhalten Sie ne-

ben der Zugriffskontrolle der Storage-Instanzen zusätzlichen Schutz, und Sie haben so eine einfache Möglichkeit, die Informationen kryptografisch zu löschen, wenn Sie fertig sind.

Bei Verschlüsselung gibt es oft Bedenken, dass sie Performance kostet, weil zusätzliche Rechenzeit zum Verschlüsseln und Entschlüsseln der Daten erforderlich ist. Zum Glück ist das mittlerweile kein so großes Problem mehr wie früher – Hardware ist billig, und alle großen Chiphersteller haben in ihre CPUs eine Hardwarebeschleunigung eingebaut. Performancebedenken sind nur selten ein guter Grund, Daten nicht zu verschlüsseln, aber es gibt Nachteile, und Sie können nur dann sicher sein, wenn Sie Ihr Szenario mit realen Lasten austesten.

Ein wichtigerer Aspekt ist bei der Verschlüsselung die Verfügbarkeit Ihrer Daten. Wenn Sie nicht auf die kryptografischen Schlüssel zugreifen können, können Sie auch nicht auf Ihre Daten zugreifen. Stellen Sie sicher, dass Sie einen Notfallprozess haben, um Zugriff auf die Schlüssel zu erhalten, und sorgen Sie dort dafür, dass das nicht geräuschlos über die Bühne geht, um solch einen Zugriff nicht unbemerkt erfolgen zu lassen.

Wenn wir mit dem »Boden« des Stacks die physische Hardware und mit der »Spitze« die Anwendung bezeichnen, dann schützen Sie sich vor mehr Arten von Angriffen, wenn die Verschlüsselung so weit oben wie möglich geschieht. Die Nachteile sind weniger Funktionalität, Performance und Kosteneffektivität gegenüber einer Verschlüsselung auf tieferer Ebene, außerdem müssen Sie selbst mehr Arbeit übernehmen.

In einer Cloud-Umgebung werden Ihre Data Assets von unterschiedlichen Arten von Cloud Assets gespeichert und verarbeitet. Im nächsten Kapitel werden Sie diese verschiedenen Typen kennenlernen und erfahren, wie Sie sie verfolgen und schützen können.

Übungen

1. Wie viele Klassifikationsstufen sind für die meisten Organisationen sinnvoll?
   a. 3
   b. 30
   c. 300
2. Was sind gute Beispiele für Data Assets, die Sie in der Cloud schützen müssen? Wählen Sie alle passenden aus.
   a. Passwort-Hashes
   b. API-Schlüssel
   c. Dokumente und Bilder, die Sie für Ihre Kunden speichern
   d. IP-Adressen Ihrer Endanwender

3. In welchen Zuständen können Daten verschlüsselt und entschlüsselt werden? Wählen Sie alle passenden aus.
    a. Wenn Daten auf persistenten Storage gespeichert werden.
    b. Wenn Daten im Speicher genutzt werden.
    c. Wenn Daten von einem Ort zum anderen transportiert werden.
    d. Wenn Daten gelöscht werden und nicht mehr genutzt werden sollen.
4. Welche der folgenden Aussagen über Key Management ist wahr?
    a. Für ein sauberes Key Management ist ein Hardware Security Module notwendig.
    b. Sie sollten niemals kryptografische Schlüssel zusammen mit den Daten abspeichern, auch wenn die Schlüssel selbst mit einem anderen Schlüssel verschlüsselt sind, der nicht bei den Daten abgespeichert ist.
    c. Cloud-Provider haben Services, die Ihnen einen Teil des Key Management abnehmen.
5. Wenn Sie den Festplattencontroller die Daten beim Schreiben auf die physische Festplatte verschlüsseln lassen – was für Angriffsarten werden dadurch abgewehrt?
    a. Angriffe, die unautorisierten Zugriff auf eine Anwendung erhalten, die die Festplatte nutzt.
    b. Angriffe, die unautorisierten Zugriff auf eine Datenbank erhalten, die von der Anwendung genutzt wird.
    c. Angriffe, die Zugriff auf die physische Festplatte erhalten.

Inhalt