Bezpieczeństwo w chmurze. Zasady i pojęcia

Chris Dotson

Tak, ta książka została napisana jako praktyczny przewodnik, jednakże w pierwszej kolejności konieczne jest omówienie kilku istotnych dla chmury zasad wysokiego poziomu bezpieczeństwa przed skupieniem się na praktycznych elementach. W przypadku, gdy czytelnik uzna, że jest doświadczonym specjalistą do spraw bezpieczeństwa, ale początkującym w środowisku w chmurze, możliwe jest przejście od razu do podrozdziału „Model współodpowiedzialności w chmurze".

Najmniejsze uprzywilejowanie

Zasada najmniejszego uprzywilejowania stanowi po prostu, że użytkownicy bądź też zautomatyzowane narzędzia powinny mieć dostęp ograniczony tylko i wyłącznie do tego, co jest im potrzebne do wykonywania pracy. Bardzo łatwo jest jednak pominąć prawa dostępu narzędzi zautomatyzowanych. Na przykład komponent uzyskujący dostęp do bazy danych nie powinien używać danych uwierzytelniających umożliwiających zapis do bazy danych, jeśli dostęp do zapisu nie jest mu potrzebny.

Praktyczne zastosowanie zasady najmniejszego uprzywilejowania często oznacza, że z zasady dostęp jest domyślnie zabroniony. Oznacza to, że użytkownicy nie mają domyślnie żadnych lub mają niewiele uprawnień i muszą przejść proces żądania i następnie zatwierdzenia wymaganych uprawnień.

W środowisku chmury niektórzy administratorzy muszą mieć dostęp do konsoli chmury, czyli strony internetowej, która umożliwia tworzenie, modyfikowanie i kasowanie zasobów w chmurze, takich jak na przykład maszyny wirtualne. W przypadku środowisk w chmurze różnych dostawców każdy posiadacz dostępu do konsoli chmury ma jednocześnie także domyślne „boskie" uprawnienia do wszystkiego, czym zarządza ten dostawca chmury. Może to obejmować możliwość odczytywania, modyfikowania lub kasowania danych w dowolnej części środowiska w chmurze, niezależnie od tego, jakie mechanizmy kontrolne obowiązują w udostępnianych systemach operacyjnych. Z tego powodu konieczne jest zapewnienie ścisłej kontroli dostępu i uprawnień do konsoli w chmurze, podobnie jak ściśle kontrolowany jest dostęp do fizycznego centrum danych w środowiskach lokalnych oraz rejestrowane jest to, co robią użytkownicy.

Bezpieczeństwo od podstaw

W sytuacji, gdy wiele z przedstawionych w tej książce elementów kontrolnych zostało idealnie zaimplementowanych, nie istniałaby potrzeba stosowania innych elementów tego typu. Bezpieczeństwo od podstaw jest założeniem, że prawie każda kontrola bezpieczeństwa może zawieść, ponieważ osoba atakująca może być wystarczająco zdeterminowana lub też istnieje problem ze sposobem, w jaki kontrola bezpieczeństwa jest realizowana.

W przypadku bezpieczeństwa od podstaw tworzonych jest wiele nakładających się na siebie warstw mechanizmów kontroli bezpieczeństwa, tak aby w razie niepowodzenia jednej kolejna z nich mogła wychwycić atakujących.

W przypadku bezpieczeństwa od podstaw istnieje pewna możliwość popadnięcia w nierozsądne skrajności, dlatego ważne jest zrozumienie zagrożenia, z jakimi prawdopodobnie będzie trzeba się zmierzyć, a które zostały opisane w dalszej części tej książki. Zasadniczo jednak powinno się być w stanie wskazać dowolną kontrolę bezpieczeństwa i powiedzieć: „A co, jeśli to się nie powiedzie?" Jeśli odpowiedź to kompletne niepowodzenie, najprawdopodobniej bezpieczeństwo od podstaw nie zostało zapewnione w wystarczającym stopniu.

Potencjalni atakujący, diagramy i granice zaufania

Istnieją różne sposoby myślenia o ryzyku, ale zazwyczaj preferowane jest podejście zorientowane na zasoby. Oznacza to, że najpierw należy się skoncentrować na tym, co musi być chronione.

Warto również pamiętać, kim najprawdopodobniej może być osoba powodująca problemy. W mowie cyberbezpieczeństwa są to „potencjalni atakujący". Na przykład osoba odpowiedzialna za bezpieczeństwo niekoniecznie może zostać zmuszona do obrony przed dobrze finansowanym podmiotem państwowym, a jedynie przed przestępcą, który jest nastawiony na czerpanie zysków z kradzieży danych lub też jest „haktywistą" mającym na celu zniszczenie strony internetowej. Należy pamiętać o tych osobach podczas projektowania wszystkich zabezpieczeń.

Mimo że dostępnych jest wiele informacji oraz dyskusji na temat potencjalnych atakujących, ich motywacji oraz metod, jakie są przez nich wykorzystywane[1], to w książce rozważono cztery główne typy potencjalnych atakujących, które warte są uwzględnienia:

- przestępczość zorganizowana lub niezależni przestępcy, zainteresowani przede wszystkim zarabianiem pieniędzy;
- „haktywiści", zainteresowani przede wszystkim dyskredytowaniem przez rozpowszechnianie skradzionych danych, popełnianiem aktów wandalizmu lub zakłócaniem działalności firmy;
- wewnętrzni napastnicy, zwykle zainteresowani dyskredytowaniem lub zarabianiem pieniędzy;
- podmioty państwowe, które mogą być zainteresowane kradzieżą tajemnic lub zakłóceniem działalności firmy.

W celu zapożyczenia metod z realnych doświadczeń użytkowników korzystne jest wyobrażenie sobie członka każdej z wymienionych grup, nadanie mu nazwy, zanotowanie wybranych „cech osobowości" na kartach, które później mogą zostać wykorzystane podczas projektowania sposobów obrony.

Drugą czynnością, jaką należy zrobić, jest zrozumienie sposobów i kierunków

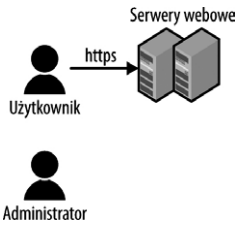
komunikacji w projektowanej aplikacji, a najłatwiejszą metodą do wykonania tego jest narysowanie całości i przeanalizowanie, gdzie mogą być zlokalizowane podatności na zagrożenia. Dostępne są całe książki o tym, jak to zrobić[2], ale nie trzeba być ekspertem, aby narysować coś na tyle przydatnego, aby było to pomocne w podejmowaniu decyzji.

W przypadku, gdy sprawa dotyczy środowiska narażonego na wysokie ryzyko, należy jednak stworzyć formalne diagramy za pomocą odpowiednich narzędzi, a nie bazować na prostych schematach.

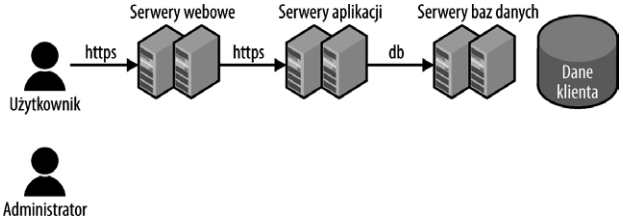
Mimo że istnieje wiele różnych możliwych architektur oprogramowania omówionej aplikacji, poniżej zaprezentowano prosty trójpoziomowy sposób projektowania:

1. W pierwszym kroku należy narysować ikonkę i oznaczyć ją jako „użytkownik". Następnie kolejną figurkę i oznaczyć ją jako „administrator" (rys. 1). Jak opisano później, może występować wiele typów użytkownika, administratora i innych ról, jest to jednak dobry punkt wyjściowy.
2. Następnie należy dodać pole dla pierwszego komponentu, z którym komunikuje się użytkownik (na przykład serwery sieciowe). Kolejnym krokiem jest narysowanie linii od użytkownika do tego pierwszego komponentu i opisanie, w jaki sposób użytkownik komunikuje się z tym komponentem (rys. 2). Należy zauważyć, że komponent ten może być usługą *serverless*, kontenerem, maszyną wirtualną lub czymś innym. Połączenie takie umożliwia każdemu komunikację z tym komponentem, tak więc najprawdopodobniej jest to pierwsza rzecz do zrobienia. Naprawdę nie jest konieczne, aby inne komponenty zaufały temu komponentowi bardziej, niż jest to konieczne.
3. Za polami pierwszych komponentów należy narysować dodatkowe pola dla wszystkich innych komponentów, z którymi musi komunikować się pierwszy komponent. Należy połączyć je liniami (rys. 3). Ilekroć zostanie osiągnięta granica systemu, który faktycznie przechowuje dane, należy oznaczyć go małym symbolem (na przykład cylindrem) i dodać opis, jakiego typu dane są tam przechowywane. Należy kontynuować tworzenie schematu, dopóki nie jest już możliwe określenie dodatkowych komponentów w projektowanej aplikacji.
4. W kolejnym kroku należy narysować, w jaki sposób administrator oraz wszelkie inne zdefiniowane role uzyskują dostęp do aplikacji. Należy pamiętać, że administrator może mieć kilka różnych sposobów komunikowania się z tą aplikacją: na przykład za pośrednictwem portalu dostawcy usługi chmury, interfejsów API, dostępu do systemu operacyjnego lub przez komunikowanie się z aplikacją w sposób podobny do tego, w jaki robi to użytkownik (rys. 4).
5. Następnie należy zaznaczyć wybrane granice zaufania kreskowanymi liniami (rys. 5). Granica zaufania oznacza, że wszystko w tej granicy może być przynajmniej w jakimś

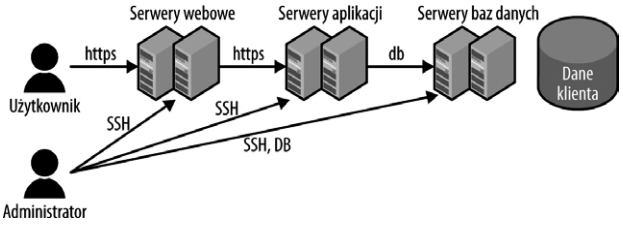
Rys. 1. Role użytkownika i administratora



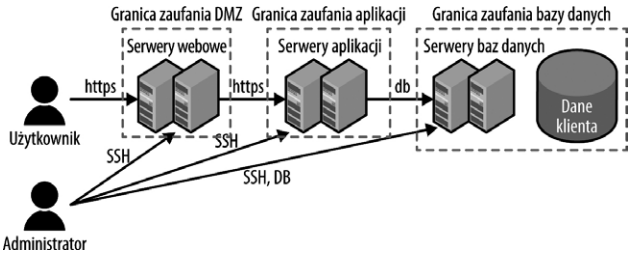
Rys. 2. Pierwszy komponent



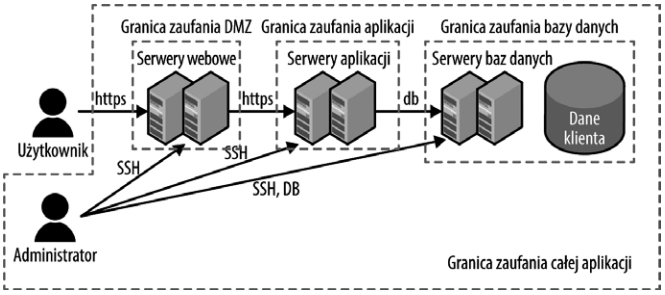
Rys. 3. Dodatkowe komponenty



Rys. 4. Dostęp administratora



Rys. 5. Granice zaufania dla komponentów



Rys. 6. Granice zaufania dla przykładowej aplikacji

stopniu pewne motywów działania czegokolwiek mieszczącego się w tej granicy, aczkolwiek wymagana jest weryfikacja przed zaufaniem czemukolwiek spoza granicy zaufania. Należy założyć, że jeśli osoba atakująca dostanie się w obręb pewnej granicy zaufania, ostatecznie uzyska pełną kontrolę nad wszystkim, co się w niej znajduje, tak więc przejście przez każdą kolejną granicę zaufania powinno wymagać wysiłku. Należy zauważyć, że na rysunku umieszczonych jest wiele serwerów internetowych w tej samej granicy zaufania. Oznacza to, że te serwery sieciowe mogą sobie całkowicie ufać, a jeśli ktoś ma dostęp do jednego, tak naprawdę ma dostęp do pozostałych. Innymi słowy, jeśli ktoś zdobędzie dostęp do jednego z tych serwerów sieciowych, nie zostaną wyrządzone dalsze szkody, jeśli zdobędzie dostęp do pozostałych.

6. Do pewnego stopnia skomponowany system jest obdarzony większym zaufaniem niż wszystko, co znajduje się poza tym systemem. Należy więc narysować kreskowaną linię wokół wszystkich komponentów, w tym administratora, ale pomijając użytkownika (rys. 6). Trzeba pamiętać, że jeśli jest wielu administratorów, takich jak administrator serwera www i administrator bazy danych, mogą się oni znajdować w różnych granicach zaufania. Fakt, że istnieją granice zaufania wewnątrz innych granic zaufania, obrazuje różne poziomy zaufania. Na przykład serwery mogą akceptować połączenia sieciowe z serwerów znajdujących się w innych granicach zaufania aplikacji, ale nadal weryfikować ich tożsamość. Mogą nawet nie przyjmować połączeń z systemów znajdujących się poza całą granicą zaufania aplikacji.

Stworzony schemat przykładowej aplikacji jest wykorzystywany w całej książce do omawiania modelu współodpowiedzialności, spisu zasobów, kontroli i monitorowania. W tej chwili na schemacie nie ma żadnych elementów sterujących specyficznych dla chmury, zostały one jednak dodane w kolejnych rozdziałach. Należy zwrócić szczególną uwagę na dowolne miejsca, w których linia oznaczająca komunikację przekracza granicę zaufania. Są to miejsca, na których należy się skoncentrować w pierwszej kolejności!

Model świadczenia usługi w chmurze

Istnieje niepisane prawo, że żadna książka na temat przetwarzania w chmurze nie jest kompletna bez omówienia modeli świadczenia usług, takich jak *Infrastruktura jako usługa* (IaaS), *Platforma jako usługa* (PaaS) i *Oprogramowanie jako usługa* (SaaS). Zamiast standardowego przeglądu zwrócono uwagę no to, że modele tych usług są użyteczne tylko do ogólnego zrozumienia pojęć. W szczególności różnica między IaaS i PaaS jest coraz mniej wyraźna. Czy usługa systemu dostarczania treści (CDN, ang. *Content Delivery Network*) buforująca informacje w Internecie tak, aby były blisko użytkownika, jest usługą PaaS czy IaaS? To naprawdę nie ma znaczenia. Ważne jest, aby zrozumieć, co oferuje (i czego nie oferuje!) ta usługa, a nie czy pasuje do konkretnej kategorii.

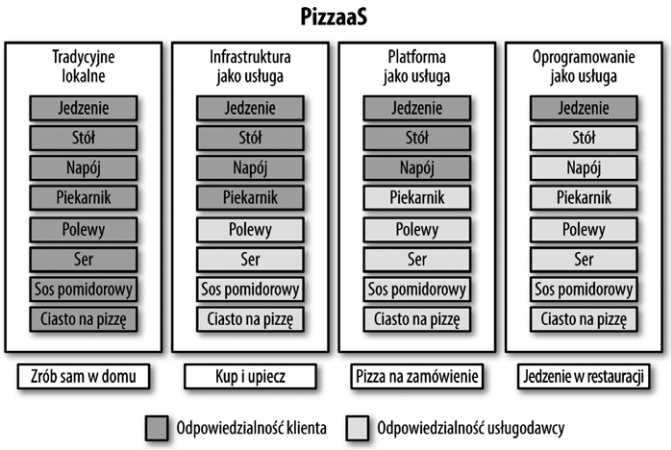
Model współodpowiedzialności w chmurze

Najbardziej podstawowe pytanie z zakresu bezpieczeństwa, na które trzeba odpowiedzieć, brzmi: „Za jakie aspekty bezpieczeństwa jesteśmy odpowiedzialni?”. W środowiskach lokalnych odpowiedź jest często udzielana pośrednio. Dział programistyczny jest odpowiedzialny za błędy w kodzie, natomiast dział operacyjny IT jest odpowiedzialny za pozostałe komponenty. Wiele organizacji stosuje obecnie model DevOps, w którym obowiązki są dzielone, a granice dzielące zespoły programistyczne i operacyjne są rozmyte lub nie istnieją. Niezależnie jednak od sposobu organizacji praktycznie cała odpowiedzialność za bezpieczeństwo zlokalizowana jest w obrębie firmy.

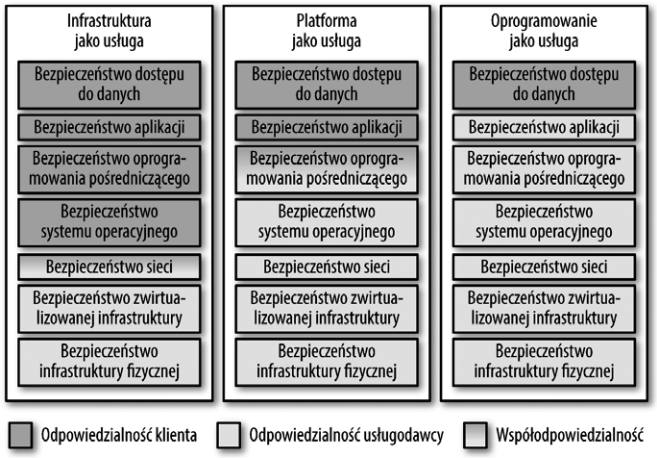
Być może jedną z najbardziej niepokojących zmian podczas przechodzenia ze środowiska lokalnego do środowiska w chmurze jest bardziej skomplikowany model współodpowiedzialności za bezpieczeństwo. W środowisku lokalnym może to być wewnętrzny dokument porozumienia, umowy z działem IT lub innym działem, który zajmuje się utrzymaniem serwerów. Jednak w wielu przypadkach biznesowi użytkownicy IT są przyzwyczajeni do przekazywania wymagań lub kodu wewnętrznemu dostawcy, który jest odpowiedzialny za wdrożenie wszystkiego, szczególnie w dziedzinie bezpieczeństwa.

Nawet osoby działające od dłuższego czasu w środowisku chmury mogą się zastanawiać, gdzie kończy się odpowiedzialność dostawcy chmury, a gdzie zaczyna się odpowiedzialność klienta. Ta linia rozgraniczająca różni się w zależności od rodzaju usługi w chmurze. Prawie wszyscy dostawcy usług w chmurze omawiają to w jakiś sposób w dokumentacji i materiałach szkoleniowych, ale najlepszym sposobem wyjaśnienia tego jest analogia do jedzenia pizzy.

Usługa Pizza-as-a-Service[3] odnosi się do chęci zjedzenia pizzy. Istnieje jednak wiele możliwości wyboru! Można po prostu zrobić pizzę w domu, chociaż potrzebne są wtedy różne składniki oraz trochę czasu. Można podbiec do sklepu spożywczego i kupić mrożoną pizzę – wymaga to wtedy tylko



Rys. 7. Pizza jako usługa



Rys. 8. Model współodpowiedzialności w chmurze

posiadania piekarnika i miejsca, w którym można ją zjeść. Można także zadzwonić do ulubionego dostawcy pizzy. Ewentualnie można po prostu usiąść w restauracji i zamówić pizzę. Wszystkie wymienione możliwości zostały umieszczone na schemacie składającym się z komponentów i osób za nie odpowiedzialnych na rysunku 7.

Tradycyjne środowisko lokalne przypomina robienie pizzy w domu. Konieczne jest zakupienie wielu różnych składników i potem samodzielne ich zmieszanie, ale zyskiwana jest pełna elastyczność. Sardele i cynamon na pszennym cieście? Jeśli ktoś da radę to zjeść, można tak zrobić.

Jednak w przypadku korzystania z usługi *Infrastruktura jako usługa* czynności podstawowe związane z przygotowaniem pizzy są już wykonane. Odpowiedzialność dotyczy jedynie odpowiedniego upieczenia, dodania sałatki i napojów. W przypadku trybu *Platforma jako usługa* jeszcze więcej decyzji jest już wykonanych i po prostu ta usługa jest wykorzystywana w ramach opracowywania ogólnego rozwiązania. Jak wspomniano wcześniej, czasem może być trudno sklasyfikować usługę jako IaaS lub PaaS, a w wielu przypadkach mieści się ona w obu kategoriach. Dokładna klasyfikacja nie jest jednak ważna, ważne jest, aby zrozumieć, co zapewnia usługa i jaką niesie odpowiedzialność.

W przypadku *Oprogramowanie jako usługa*, co zostało porównane na rysunku 7 z jedzeniem w restauracji, wydawać się może, że wszystko zostało już zrobione. Ale tak nie jest. Nadal konieczne jest jedzenie w sposób bezpieczny, a restauracja nie ponosi odpowiedzialności, jeśli klient zadławi się jedzeniem. W przypadku SaaS sprowadza się to głównie do właściwego zarządzania kontrolą dostępu.

Rzeczywistość przetwarzania w chmurze jest niestety nieco bardziej skomplikowana niż jedzenie pizzy, występują

więc pewne szare obszary. Elementy wymienione na dole schematu są namacalne, często dosłownie. Dostawca chmury ponosi pełną odpowiedzialność za bezpieczeństwo infrastruktury fizycznej. Są to często działania wykraczające poza to, co firmy mogą racjonalnie zrobić na miejscu, takie jak dostęp biometryczny ze środkami zapobiegającymi konfrontacji, strażnicy, bariery płytowe i podobne sposoby trzymania nieupoważnionego personelu z dala od urządzeń fizycznych.

Podobnie, jeśli dostawca oferuje środowiska wirtualne, kontrola bezpieczeństwa zwirtualizowanej infrastruktury oddzielającej środowisko wirtualne klienta od innych środowisk wirtualnych jest odpowiedzialnością dostawcy. Kiedy na początku 2018 r. ujawniły się luki Spectre i Meltdown, jednym z potencjalnych efektów było to, że użytkownicy jednej maszyny wirtualnej mogli odczytać pamięć innej maszyny wirtualnej na tym samym komputerze fizycznym. W przypadku klientów IaaS usunięcie tej części luki było obowiązkiem dostawcy usługi w chmurze, ale naprawienie luk w systemie operacyjnym należało już do klienta.

Na rysunku 8 bezpieczeństwo sieci zostało pokazane jako współodpowiedzialność w sekcji IaaS. Dlaczego? Trudno to pokazać na schemacie, ale istnieje kilka warstw sieci, a odpowiedzialność za każdą z nich spoczywa na innej grupie. Dostawca usług w chmurze ma własną sieć, za którą odpowiada, ale zwykle istnieje jeszcze nad nimi sieć wirtualna (na przykład niektórzy dostawcy usług w chmurze oferują wirtualną chmurę prywatną) i to klient odpowiada za przeniesienie jej do rozsądnych stref bezpieczeństwa i wprowadzenie właściwych zasad dostępu między nimi. W wielu implementacjach wykorzystuje się również sieci nakładek, zapory sieciowe oraz szyfrowanie podczas przesyłania, za które odpowiedzialność ponosi klient.

W przypadku systemu operacyjnego podział bezpieczeństwa jest zwykle prosty. W przypadku wykorzystania modelu IaaS odpowiedzialność spoczywa na użytkowniku. W przypadku zakupu platformy lub oprogramowania to dostawca ponosi odpowiedzialność za sprawy bezpieczeństwa. Ogólnie rzecz biorąc, w przypadku zakupu tego typu usługi nie ma dostępu do bazowego systemu operacyjnego. Za ogólną zasadę można przyjąć, że jeśli istnieje możliwość złamania zabezpieczenia systemu operacyjnego, to zwykle istnieje obowiązek jego zabezpieczenia!

Oprogramowanie pośredniczące, w tym kontekście, to ogólna nazwa oprogramowania, takiego jak bazy danych, serwery aplikacji lub systemy kolejkowania. Znajdują się one między systemem operacyjnym a aplikacją i nie są używane bezpośrednio przez użytkowników końcowych. Służą natomiast do opracowywania rozwiązań dla użytkowników końcowych. Jeśli wykorzystywany jest model PaaS, bezpieczeństwo oprogramowania pośredniczącego jest często wspólną odpowiedzialnością. Dostawca może aktualizować oprogramowanie lub łatwo udostępniać aktualizacje, ale to użytkownik ponosi odpowiedzialność za ustawienia związane z bezpieczeństwem, takie jak szyfrowanie.

Warstwa aplikacji jest tym, czego faktycznie używa użytkownik końcowy. W przypadku modelu SaaS za luki w tej warstwie, takie jak *Cross-site scripting* lub wstrzykiwanie kodu SQL, odpowiada dostawca, aczkolwiek czytelnik tej książki najprawdopodobniej nie jest tylko użytkownikiem czyjejś usługi SaaS. Nawet jeśli wszystkie pozostałe warstwy mają doskonałe zabezpieczenia, podatność na zagrożenia w warstwie zabezpieczeń aplikacji może łatwo zostać wykorzystana do przechwycenia wszystkich chronionych informacji. Ostatecznie bezpieczeństwo dostępu do danych jest prawie zawsze obowiązkiem klienta. W przypadku, gdy dostawca usług w chmurze zostanie błędnie poinformowany o tym, że może udzielać dostępu do określonych danych, takich jak udzielanie niepoprawnych uprawnień do pamięci, oprogramowania pośredniczącego lub SaaS, to tak naprawdę nie może nic zrobić w sprawie bezpieczeństwa.

Podstawową przyczyną wielu incydentów związanych z bezpieczeństwem jest założenie, że dostawca usług w chmurze obsługuje pewne elementy w momencie, gdy okazuje się, że nie są one w ogóle

obsługiwane. Wiele rzeczywistych przykładów incydentów bezpieczeństwa wynikających z niedostatecznego zrozumienia modelu współodpowiedzialności pochodzi z otwartych kubełków (*buckets*) Amazon Web Services Simple Storage Service (AWS S3). Oczywiście pamięć AWS S3 jest bezpieczna i szyfrowana, nie ma to jednak znaczenia, jeśli kontrola dostępu nie jest ustawiona poprawnie. Tego typu nieporozumienie spowodowało wyciek:

- danych dotyczących 198 milionów wyborców w USA;
- danych dotyczących śledzenia samochodów firmowych;
- danych klientów bezprzewodowych;
- ponad 3 milionów danych badań demograficznych;
- ponad 50 000 raportów kredytowych obywateli Indii.

Jeśli ktoś uważa, że dyskusja na temat współodpowiedzialności jest zbyt prosta, to należy mu pogratulować, gdyż znajduje się w najwyższym kwartylu. Według badania przeprowadzonego przez Barracuda Networks w 2017 r. (http://bit.ly/2EcgeQG) model współodpowiedzialności jest nadal bardzo źle rozumiany przez przedsiębiorstwa. Około 77% decydentów IT stwierdziło, że wierzy w to, że dostawcy chmury publicznej są odpowiedzialni za zabezpieczenie danych klientów w chmurze, a 68% uważa, że ci dostawcy są również odpowiedzialni za zabezpieczenie aplikacji klientów. Jeśli przeczytamy umowę z dostawcą chmury, przekonamy się, że to po prostu nieprawda!

Zarządzanie ryzykiem

Zarządzanie ryzykiem to szeroka tematyka, na którą składa się obszerna literatura. Jeśli czytelnik jest zainteresowany poważnym podejściem do zarządzania ryzykiem, zalecana jest następująca lektura *The Failure of Risk Management: Why It's Broken and How to Fix It* autorstwa Douglasa W. Hubbarda (Wiley) oraz publikacja NIST Special Publication 800-30 Rev 1 (http://bit.ly/2VmsLrV). Można to ująć w skrócie: ludzie naprawdę źle oceniają ryzyko oraz to, co z nim zrobić. W tej części książki przedstawiono najistotniejsze elementy niezbędne do zarządzania ryzykiem wystąpienia incydentów bezpieczeństwa i naruszenia danych.

Mówiąc wprost, ryzyko jest czymś złym, co może się przydarzyć. W większości systemów zarządzania ryzykiem poziom ryzyka opiera się na kombinacji tego, jak prawdopodobne jest, że zdarzy się coś złego (prawdopodobieństwo), oraz tego, jak złe będą skutki tego wydarzenia (wpływ). Na przykład, jeśli jest coś, co najprawdopodobniej się wydarzy (np. odgadnięcie hasła „1234") i w efekcie jest to złe (utrata wszystkich plików klientów i opłacenie wysokich grzywien), to jest to wysokie ryzyko. Coś, co jest bardzo mało prawdopodobne (asteroida niszcząca jednocześnie dwa różne regionalne centra danych), ale byłoby bardzo złe w skutkach (wycofanie się z działalności), może stanowić jedynie niewielkie ryzyko, oczywiście w zależności od używanego systemu do decydowania o poziomie ryzyka[4].

W niniejszej książce został poruszony problem nieznanego ryzyka, dla którego nie ma wystarczającej ilości informacji dotyczącej prawdopodobieństwa i skutków oraz znanego ryzyka, w przypadku którego można określić, czego dotyczy. Jeśli zagrożenia zostały sprecyzowane, to można z nimi zrobić jedną z czterech rzeczy:

1. Unikanie ryzyka. W zakresie bezpieczeństwa informacji zwykle oznacza to wyłączenie systemu – nie ma już ryzyka, ale także nie ma żadnych korzyści, jakie wynikają z korzystania z systemu.
2. Ograniczanie ryzyka. Ryzyko nadal występuje, ale robiono dodatkowe rzeczy w celu zmniejszenia prawdopodobieństwa wystąpienia złego wydarzenia lub negatywnych rezultatów, jeśli takie zdarzenie się wydarzy. Na przykład możliwe jest wybranie mniej wrażliwych danych do przechowywania, tak aby w przypadku ich naruszenia skutek nie był taki negatywny.
3. Przenoszenie ryzyka. Możliwe jest zapłacenie komuś innemu za zarządzanie, tak aby ryzyko stanowiło problem kogoś innego. Robi się tak często w środowiskach w chmurze, gdzie przenosi się wiele zagrożeń związanych z zarządzaniem niższymi poziomami systemu na dostawcę chmury.
4. Zaakceptowanie ryzyka. Po przyjrzeniu się ogólnemu poziomowi ryzyka i korzyściom z kontynuowania działalności można zdecydować się na zaakceptowanie ryzyka. Należy poprosić wszystkich interesariuszy, aby zgodzili się na istnienie pewnego ryzyka i następnie przejść do dalszych działań.

Każde z tych działań może być uzasadnione. Niedopuszczalny jest jednak brak pojęcia, jakie jest ryzyko albo, mimo świadomości o istnieniu ryzyka, jego akceptacja bez rozważenia konsekwencji lub uzyskania akceptacji od interesariuszy. Jako minimum powinno się utworzyć listę w arkuszu kalkulacyjnym lub dokument zawierający szczegółowe informacje na temat znanych zagrożeń, podjętych działań i wymaganych zatwierdzeń.

Przypisy

1 Verizon Data Breach Investigations Report (https://vz.to/2LoBfyq) jest doskonałym darmowym zasobem do zrozumienia różnych rodzajów udanych ataków, posortowanych ze względu na rodzaj przemysłu i użyte metody, a jego streszczenie jest bardzo czytelne.

2 Autor poleca: Adam Shostack Threat Modeling: Designing for Security (Wiley).

3 Oryginalny pomysł z artykułu Alberta Barrona.

4 Ryzyko może również oddziaływać lub agregować. Mogą istnieć dwa rodzaje ryzyka, z których każde ma stosunkowo małe prawdopodobieństwo i wpływ, ale mogą wystąpić wspólnie, a skutki mogą być większe. Na przykład wpływ awarii jednej z dwóch linii zasilających może być nieistotny, ale już awaria obu może mieć katastrofalne skutki. Jest to często trudne do wykrycia, a awaria zasilania lotniska w Atlancie w 2017 r. jest tego dobrym przykładem.

Fragment pochodzi z książki: *Bezpieczeństwo w chmurze. Przewodnik po projektowaniu i wdrażaniu zabezpieczeń*,
Chris Dotson, Wydawnictwo Naukowe PWN, Warszawa 2020