**Jak chronić aplikacje w chmurze przed zagrożeniami?**

**Natywne środki bezpieczeństwa w chmurze koncentrują się przede wszystkim na ochronie przed możliwymi do zidentyfikowania zagrożeniami, wykorzystując innowacyjne technologie, takie jak Big Data i sztuczna inteligencja, do monitorowania i zapobiegania potencjalnym atakom. Jednak pomimo tych wysiłków żaden system nie będzie całkowicie odporny na zagrożenia. Nawet przy zastosowaniu najnowocześniejszych zabezpieczeń bezpieczeństwo chmury nigdy nie jest wystarczająco dobre, jeśli nadal występują niezidentyfikowane luki w zabezpieczeniach – wyjaśniają eksperci Check Point Software.**

**Eksperci ds. bezpieczeństwa skupiają się na znanych wskaźnikach ryzyka.**

Branża cyberbezpieczeństwa zazwyczaj koncentruje się na badaniu możliwości wykorzystania przez atakujących i identyfikowaniu metod unikania błędów w kodach twórców rozwiązań zabezpieczających. Jednak najbardziej szkodliwe ataki to te, które zaskakują nas całkowicie nieświadomie, a ich obecność uświadamiamy sobie dopiero wtedy, gdy już są realizowane.

**Krótka historia chmury**

Około 12 lat temu Check Point uruchomił znacznie silniejsze algorytmy, które pozwoliły na:

l Zbieranie i analizowanie danych na szeroką skalę… czyli w zasadzie było to utworzenie **big data.**

l Potrzebny był jednak sposób na ekonomiczne przechowywanie tych danych, dlatego narodziły się **sieci wirtualne.**

l Tymczasem wraz z rozwojem aplikacji mobilnych na całym świecie dane nagle musiały się przemieszczać, być wszędzie… i nawiązano współpracę z **dostawcami usług internetowych w chmurze**. Wyzwania związane z przetwarzaniem w chmurze rozwiązano, uruchamiając wszystko jako **skalowalny kod**, podzielony na **usługi** działające na **maszynach wirtualnych**, które stają się aktywne po ich wywołaniu.

l Wprowadzono **uczenie maszynowe w celu optymalizacji danych i generowania głębokich spostrzeżeń.**

l **Kubernetes** został wprowadzony, aby zapewnić większą szczegółowość kodowania i uruchamiania usług.

l W tym czasie krypto przyczyniło się do rozwoju **przetwarzania GPU, a generatywna sztuczna inteligencja** stała się czymś powszechnym.

l I wreszcie to **pandemia** dała chmurze największy impuls.

Zdalne punkty końcowe, dostęp do danych z dowolnego miejsca i szybkość technologii wymagają usług w chmurze. Obecnie małe i duże firmy już korzystają z chmury lub do niej docierają.

**Chmura to nie problem – to rozwiązanie.**

Chmura stała się integralną częścią naszego codziennego życia, dojrzewając do punktu, w którym jej obecność jest wszechobecna. Jako profesjonaliści zajmujący się chmurą często jesteśmy pochłonięci rozwiązywaniem związanych z nią wyzwań i łatwo zapomnieć, jak wspaniała jest naprawdę chmura. Chmura dała nam wiele korzyści i możliwości, które kiedyś były nie do wyobrażenia. Trzeba przyznać, że przedstawił także sporo problemów. Jednak koniecznie musimy zdać sobie sprawę, że chmura nie jest problemem; to jest rozwiązanie. Wykorzystując zalety chmury, możemy w nadchodzących latach ulepszyć środki bezpieczeństwa i rozszerzyć nasze działania na wyższy poziom.

**Chodzi o wymianę danych:**

Aplikacje opierają się na dwóch głównych kanałach wymiany danych: Front Door w chmurze, gdzie użytkownicy przesyłają żądania dotyczące różnych usług z aplikacji w chmurze za pośrednictwem publicznego Internetu, sieci komórkowych i [VPN](https://www.checkpoint.com/quantum/remote-access-vpn/) . Następnie są drzwi serwisowe, przez które w sposób ciągły przesyłany jest kod i dane, aby aplikacja mogła działać. Zasadniczo chroniony jest przesył. Firmy muszą zrozumieć każde zgłaszane żądanie i posiadać zestaw narzędzi pozwalających potwierdzić, że wymiana jest bezpieczna.

**Każde żądanie użytkownika jest potencjalnym zagrożeniem….**

Bogactwo wymiany danych różnymi kanałami i chmurami hybrydowymi może oznaczać również kłopoty. Jeśli spojrzymy na uproszczony diagram łańcucha dostaw, od razu zobaczymy wiele komponentów, które wymieniają dane i mogą potencjalnie służyć jako tylne drzwi dla hakerów, którzy chcą przedostać się do naszego środowiska.

Za każdym razem, gdy dane są wymieniane pomiędzy komponentami, istnieje ryzyko, że wymiana zostanie naruszona lub zmanipulowana. Dodajmy do tego fakt, że polegamy na wielu kanałach i usługach stron trzecich, również wymieniającychdane, które mogą mieć własny zestaw luk w zabezpieczeniach.

**Przygotowujemy nasze wojska do bitwy i robimy wszystko, co w naszej mocy, aby chronić nasze zasoby w terenie, ale naszym głównym celem powinno być przede wszystkim uniknięcie wojny!**

Aby zabezpieczyć naszą działalność, stosujemy różne środki bezpieczeństwa, takie jak WAF dla drzwi wejściowych, [CSPM](https://www.checkpoint.com/cyber-hub/cloud-security/what-is-cspm-cloud-security-posture-management/pci-dss-compliance/) i ochronę obciążenia dla zawartości chmury oraz skanowanie kodów i bezpieczeństwo sieci dla punktu końcowego. Dzięki zastosowaniu tych środków mamy pewność, że poradzimy sobie ze wszystkimi znanymi zagrożeniami. Nie możemy jednak lekceważyć możliwości wystąpienia nieznanych zagrożeń, dlatego zachowujemy czujność i potrafimy dostosowywać się do kwestii bezpieczeństwa.

**Spróbujmy zrozumieć, przed czym się chronimy, zaczynając od kilku podstawowych definicji:**

**Znane luki w zabezpieczeniach** oznaczają rzeczy, które sami robimy, takie jak:

l błędne konfiguracje lub ukryte dane uwierzytelniające,

l znane luki w zabezpieczeniach – które mogą pozwolić na złośliwą aktywność do czasu załatania i…

l dynamiczne wskaźniki ryzyka oparte na szerokiej bazie danych wskaźników ataków, takich jak podejrzane wzorce zachowań, złośliwe adresy IP, wzorce ataków itp.

Poprzez ciągłe poszukiwanie tych problemów i wskaźników możemy zapewnić odpowiednią postawę, ochronę i zerową tolerancję w całej sieci aplikacji.

**Nieznanymi lukami** mogą być:

l nieodkryte luki w oprogramowaniu, które mogą spowodować odejście składnika oprogramowania od jego pierwotnego przypisania i umożliwienie dostępu lub manipulacji.

l podatne backdoory, niektóre z założenia (na przykład funkcja zapominania hasła) i backdoory niezamierzone – wykryte przez złośliwych aktorów lub

l i wreszcie, po prostu nowe techniki i metody ataku, które hakerzy cały czas wymyślają, a o których jeszcze nie wiemy …

**Ponieważ tak naprawdę nie wiemy, jak chronić się przed tymi nieznanymi zagrożeniami – są one oczywiście dokładnie tym, czego szukają napastnicy i dlatego też exploity dnia zerowego stały się tak powszechne.**

**Numery CVE są nieznane, dopóki nie mamy podpisu**

Według Statistica tylko w 2022 r. [internauci na całym świecie odkryli ponad 25 000 nowych CVE.](https://blog.checkpoint.com/securing-the-cloud/meet-the-new-cloudguard-risk-management-in-action/#:~:text=According%20to%20a%20report%20by%20Statista%2C%20in%202022,averaging%20out%20to%2062%20critical%20vulnerabilities%20per%20day.) Warto jednak zauważyć, że rozwiązanie problemu CVE zajmuje średnio 65 dni, więc dopóki nie pojawi się stabilna łatka naprawiająca podstawowy problem, jedyną linią obrony, jaką mamy, jest WAF, a jednocześnie tenże WAF jest nie widzi nieznanych ataków, dopóki wydany nie zostanie podpis.

**Kontynuujmy tę historię, skupiając się na log4shell jako głównym przypadku użycia.**

[Log4Shell](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-44228) jest doskonałym przykładem, ponieważ spełnia wszystkie wymagania:

l Log4j to biblioteka rejestrowania Java, z której korzystają setki milionów maszyn. Trudno wyobrazić sobie dużą aplikację, która nie korzystałaby z niej w jakiś sposób lub w jakiejś formie.

l W ciągu pierwszych 72 godzin zhakowano go ponad 800 000 razy, zarówno dlatego, że był łatwy do zhakowania, jak i umożliwiał atakującym uzyskanie głębokiego dostępu.

l Oszacowanie liczby ataków od 2021 r. jest prawie niemożliwe, ale szacujemy szkody na ponad 1,5 miliarda dolarów , co powinno dać pewną wskazówkę.

l Co ciekawe, 1 na 4 aplikacje nadal ma problematyczne wersje log4j, ponieważ nie można było uzyskać dostępu do nich wszystkich, nawet jeśli próbowałeś.

**Log4shell pokazuje również, jak sytuacja może się pogorszyć, zanim się poprawi.**

l W tym przypadku luka została odkryta 24 listopada idwa dni później otrzymała numer CVE.

l Tydzień później, pierwszego grudnia, istniały już dowody wykorzystywania jej „na wolności”.

l Prawdziwy bałagan zaczął się 9 grudnia – był to pierwszy raz, kiedy Apache publicznie ujawnił tę lukę, obejmującą łatkę dla dotkniętego log4j2. Wydawało się, że organizacje mogą natychmiast rozpocząć działania łagodzące; jednakże Apache potrzebował kolejnych pięciu dni, zanim zdał sobie sprawę, że jeszcze bardziej popularny log4j1 w rzeczywistości również jest podatny na ataki, więc łatanie musiało zacząć się od nowa.

l Aby jeszcze bardziej skomplikować sprawę, 17 grudnia opublikowano dodatkowe CVE, ujawniające jeszcze więcej scenariuszy exploitów… w rzeczywistości ponad miesiąc później problemy i warianty były nadal ujawniane i dopiero po 35 dniach zespoły ds. bezpieczeństwa mogły w końcu rozpocząć łatanie aktywa krytyczne. To prosta matematyka – jeśli opanowanie tego zjawiska zajęło 35 dni, a naprawa zajmie średnio kolejne 65 dni, oznacza to, że przez sto dni jesteśmy narażeni na najgorszy rodzaj ataków!

l Osoba atakująca może z łatwością dołączyć złośliwy ciąg do prostego żądania logowania lub wyszukiwania, korzystając z luki w zabezpieczeniach protokołu Java.

Log4Shell to exploit funkcji „podstawiania komunikatów” Log4j, która umożliwia programową modyfikację dzienników zdarzeń poprzez wstawianie ciągów znaków wywołujących treść zewnętrzną. Kod obsługujący tę funkcję umożliwiał „ [wyszukiwanie](https://logging.apache.org/log4j/2.x/manual/lookups.html) ” przy użyciu adresów URL Java Naming and Directory Interface (JNDI). Zatem jedyne, co musi zrobić osoba atakująca, to wstawić tekst z osadzonymi złośliwymi adresami URL JNDI do żądań wysyłanych do oprogramowania przy użyciu Log4j na serwerze rejestrującym LDAP — adresy te powodują zdalne załadowanie i wykonanie kodu przez rejestrator. Osoby atakujące wykorzystują tę lukę do naruszania infrastruktury wirtualizacji, instalowania i uruchamiania oprogramowania ransomware, kradzieży danych uwierzytelniających systemu, przejmowania szerokiej kontroli nad zaatakowanymi sieciami i wydobywania danych.

**Od analizy statycznej po kontekstową sztuczną inteligencję**

Czy istniał sposób na zablokowanie log4Shell? Prostą odpowiedzią jest **tak** . Można to oczywiście osiągnąć poprzez wykorzystanie pełnego potencjału uczenia maszynowego i sztucznej inteligencji.

W przeszłości sztuczna inteligencja umożliwiała nam wspaniałe rzeczy – zaczęło się od możliwości optymalizacji sposobu, w jaki przetwarzamy duże ilości danych lub usprawniania procesów… a wraz z rozwojem CNAPP i konsolidacją danych możemy teraz spojrzeć na nasze dane o zagrożeniach jako całość i wyciągaj wnioski, które pomogą nam ulepszyć i skalować nasze bezpieczeństwo.

Ale co by było, gdybyśmy mogli użyć tej samej sztucznej inteligencji, aby patrzeć do wewnątrz, a nie na zewnątrz? Jeśli dogłębnie zrozumiemy normalne zachowanie aplikacji, powinniśmy być w stanie wykryć wszystko, co wykracza poza tę normę i po prostu to zablokować!

**Przyjrzyjmy się, jak dokładnie tego używa CloudGuard WAF.**

Check Point [CloudGuard WAF](https://www.checkpoint.com/cloudguard/waf/) wykorzystuje opatentowany silnik uczenia maszynowego do ciągłego analizowania żądań użytkowników w sieci i interfejsach API za pośrednictwem protokołu HTTP. Silnik sztucznej inteligencji WAF uczy się, w jaki sposób użytkownicy zwykle wchodzą w interakcję z aplikacją internetową i automatycznie wykrywa żądania wykraczające poza normalne operacje. Żądania te są dalej analizowane w celu podjęcia decyzji, czy są one złośliwe, czy nie. Ta precyzja niesie ze sobą prawie 0 fałszywych alarmów i pozwala nam blokować problemy bez polegania na podpisach i regułach.

CloudGuard WAF to nowa generacja zapór sieciowych dla aplikacji internetowych. Jest dystrybuowany jako nanoagenci w środowisku chmurowym, skutecznie przechwytujący wszystkie interakcje http i analizujący je w czasie rzeczywistym. CloudGuard WAF nie opiera się na sygnaturach ani regułach blokowania ataków. Stosuje się dwa kroki, aby zapewnić sobie ochronę zarówno przed znanymi, jak i nieznanymi zagrożeniami.

Na etapie 1 silnik wymuszający oparty na uczeniu maszynowym szuka wskaźników ataku w żądaniu HTTP. Ocena opiera się na nadzorowanym modelu, który identyfikuje wskaźniki i wiąże je z konkretnym, statystycznym prawdopodobieństwem lub „wynikiem” bycia częścią ataku.

Punktację przypisuje się każdemu wskaźnikowi z osobna oraz dla par wskaźników. Ponadto wskaźniki są powiązane z określonymi rodzinami ataków, w których zazwyczaj występują. Proces punktacji umożliwia CloudGuard podjęcie dokładnej wstępnej decyzji o prawdopodobieństwie ataku na żądanie HTTP.

**Tutaj kończy się tradycyjny WAF**

Ale w drugiej części dzieją się interesujące rzeczy:

Podejrzane żądania są analizowane w **silniku oceny kontekstowego uczenia maszynowego,** aby zyskać dodatkową pewność, że każde żądanie HTTP, które zostało wskazane jako potencjalnie złośliwe, jest rzeczywiście atakiem. Korelujemy dodatkowe parametry, takie jak: struktura aplikacji, zachowanie użytkownika/tłumu, treść użytkownika i transakcje i inne.

Dzięki temu możliwe jest niezwykle precyzyjne wykrywanie, skuteczne oznaczanie wszelkich nieprawidłowych lub szkodliwych żądań i blokowanie ich przed wejściem.

**CloudGuard WAF skutecznie zablokował WSZYSTKIE główne ataki dnia zerowego w ubiegłych latach, w tym Log4Shell, Spring4Shell i MOVEit . Klienci Check Pointa byli chronieni od pierwszego dnia, nie musieli nawet przeprowadzać aktualizacji.**