CHRIS HUGHES
NIKKI ROBINSON
PRZEDMOWA **RON GULA**

EFEKTYWNE ZARZĄDZANIE PODATNOŚCIAMI NA ZAGROŻENIA

JAK MINIMALIZOWAĆ RYZYKO W **CYFROWYM EKOSYSTEMIE**

Helion
WILEY

Tytuł oryginału: Effective Vulnerability Management:
Managing Risk in the Vulnerable Digital Ecosystem

Tłumaczenie: Małgorzata Dąbkowska-Kowalik, Witold Sikorski

ISBN: 978-83-289-2160-3

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
helion.pl/user/opinie/efzapo
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Helion S.A.
ul. Kościuszki 1c, 44-100 Gliwice
tel. 32 230 98 63
e-mail: *helion@helion.pl*
WWW: *helion.pl* (księgarnia internetowa, katalog książek)

Printed in Poland.

Spis treści

2 Zarządzanie łataniem

Każdy dobry program zarządzania łataniem składa się z automatycznych i ręcznych technik oraz procesów łatania. Z uwagi na to, że natura infrastruktury jest dynamiczna, powinien to być ciągły proces oceny, co działa, co jest łatane, a czego brakuje. Bez silnego programu zarządzania podatnościami systemy mogą stać się przestarzałe lub osiągnąć swój kres. Tak samo bez zaprojektowanego i uporządkowanego planu systemy pozostaną narażone na podatności dnia zerowego przez dni, tygodnie lub dłuższy czas, podobnie do tego, co stało się w przypadku MOVEit lub SolarWinds.

Podstawy zarządzania łataniem

Pomimo szumu w branży związanego z najnowszymi jaskrawymi podatnościami dnia zerowego złośliwi aktorzy regularnie celują w „przestarzałe podatności" — podatności z istniejącymi łatami, które są znane ze swobodnego wykorzystywania (*www.rezilion.com/blog/report-vintage-vulnerabilities-never-go-out-of-fashion*). Wynika to z faktu, że choć wiadomo, iż te podatności są wykorzystywane i istnieją do nich łaty, organizacje nadal borykają się z możliwościami naprawy, będąc w stanie naprawić średnio jedną na dziesięć nowych podatności miesięcznie. Rysunek 2.1 pokazuje warstwy działań związanych z zarządzaniem łataniem, które znajdą się w każdej infrastrukturze IT, zarówno w chmurze, jak i lokalnie.

Każda organizacja musi określić, jaki proces zarządzania łataniem będzie dla niej najlepszy, ale podstawy obejmują kompleksową inwentaryzację, okna konserwacji narzędzia lub procesy automatyzujące łatanie oraz procesy ponownego uruchamiania lub przenoszenia systemów w tryb offline w celu testowania.

Plan zarządzania łataniem musi także obejmować kwestie wysokiej dostępności, czyli nadmiarowości, aby zapewnić, że systemy są łatane i tworzone są ich kopie zapasowe w celu utrzymywania ich online na potrzeby klientów. Typowe dla umów o gwarantowanym poziomie usług (ang. *service level agreements* — SLA) jest uwzględnienie okna konserwacji, czyli ram czasu przeznaczonego na łatanie i inne działania. Obejmuje to także plan wycofania lub strategię tworzenia kopii zapasowych w sytuacji, gdy łaty przestaną działać lub spowodują zakłócenia usług.

Rysunek 2.1. Podstawy piramidy zarządzania łataniem

Ręczne zarządzanie łataniem

Ręczne łatanie to łatanie w środowisku wymagającym człowieka do pobrania i zainstalowania łaty, ponownego uruchamiania serwerów na podstawie wymagań organizacji czy też analizy łatania na potrzeby serwerów, które wykracza poza zwykłe okno konserwacji. Typowe organizacje będą stosować kombinację ręcznego i automatycznego łatania ze względu na to, że nie wszystkie czynności da się zautomatyzować, a implementacje mogą się różnić zależnie od warunków organizacyjnych i technicznych.

Wprawdzie najlepiej byłoby mieć jak najwięcej zautomatyzowanych czynności, ale zawsze będzie potrzeba ręcznego wprowadzania łat. Na przykład organizacja może mieć aplikację, która wymaga działania przez 99 procent czasu. Zespół będzie musiał zastosować łaty w środowisku deweloperskim, potem wykonać testowanie akceptacji użytkownika (ang. *user acceptance testing* — UAT), a na koniec zastosować łaty w produkcji. Choć stosowanie łat może być po części zautomatyzowane, to restartowanie serwerów i testowanie mogą być nadal zadaniami ręcznymi (na wczesnych etapach programu zarządzania łataniem) przed łataniem kolejnych serwerów produkcyjnych.

Innym częstym zastosowaniem ręcznego łatania jest testowanie jednorazowych łat, gdy ogłoszona zostanie podatność dnia zerowego. Zależnie od wagi podatności może być możliwe pobranie i automatyczne zainstalowanie łat, ale nadal konieczne będą przyspieszone testowanie i naprawa. Przy skróconym harmonogramie grupy mogą nie chcieć zmieniać istniejącej automatyzacji w celu uwzględnienia jednorazowej sytuacji lub wyjątkowych łat, które nie są zgodne ze zwykłym trybem łatania.

Ryzyko ręcznego łatania

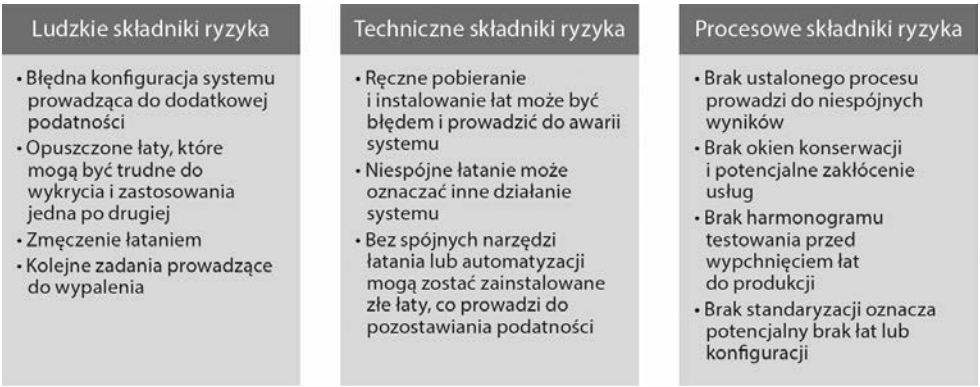
Jak wspomniano wcześniej, kombinacja łatania automatycznego i ręcznego jest bardziej rzeczywistością w organizacjach niż automatyczne wykonywanie wszystkich czynności związanych z zarządzaniem łataniem za pomocą narzędzi. Jednak każde działanie ręczne

niesie ze sobą ryzyko. Podstawowa obawa wiąże się z ludzkim błędem. Każdy administrator IT lub inżynier systemowy stara się wyważyć działania inżynierskie i zadania związane z zabezpieczeniami.

Tak jak w przypadku każdej innej roli administrator systemu, który żongluje wieloma obowiązkami, systemami, projektami i zadaniami, jest narażony na błędy tu i tam. Te mogą prowadzić do następujących skutków:

1. Brakujące łaty dla możliwych do wykorzystania podatności.
2. Źle skonfigurowane ustawienia.
3. Dodatkowe podatności narażające system.

Rysunek 2.2 uwypukla poziom ryzyka, jeśli zespół zarządzania łatami wyłącznie pobiera je i instaluje ręcznie w systemie.



Rysunek 2.2. Ryzyko związane z ręcznym łataniem

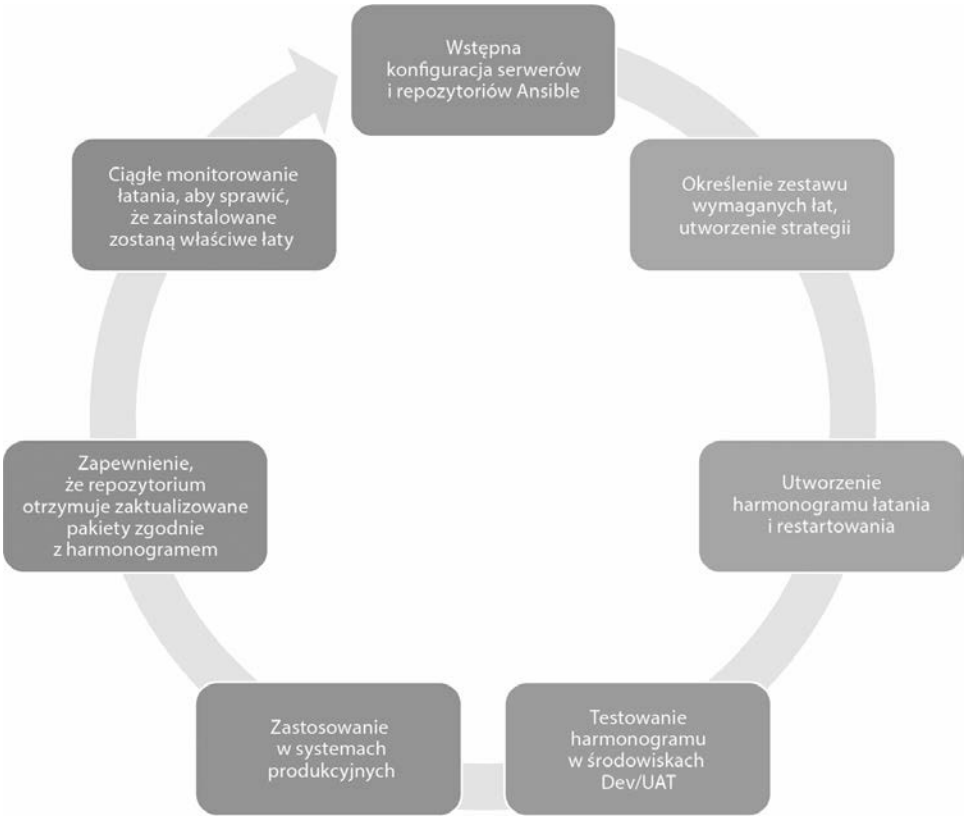
Innym potencjalnym aspektem ręcznego łatania jest wysiłek wkładany w pobieranie łat, instalowanie ich i testowanie, aby otrzymać w pełni zaktualizowany system wprowadzany do produkcji. Przy łataniu automatycznym wszelkie zaległe łatki poza harmonogramem, zbiorcze poprawki lub indywidualne łaty są zazwyczaj pobierane i instalowane zgodnie z określonym wcześniej harmonogramem. Dodatkowy czas potrzebny na ręczne instalowanie łat oznacza dodatkowe ryzyko wykorzystania podatności.

Z drugiej strony, jeśli nie ma żadnego środowiska testowego do łatania, a ręczne łaty muszą być instalowane w systemach produkcyjnych, to istnieje poważne zagrożenie dla działania. Bezpieczeństwo to równowaga między zarządzaniem ryzykiem i działaniami, zgodnie z triadą: poufność, integralność, dostępność (ang. *Confidentiality*, *Integrity*, *Availability* — CIA). Jeśli istnieje wymóg, aby systemy klienckie działały przez 99 procent czasu, to administratorzy i właściciele systemu będą poświęcać dodatkowy czas na testowanie, zanim łaty zostaną zastosowane w produkcji.

Administratorzy systemu powinni rozważyć stopniowe wdrażanie łat w całym środowisku. Ta ścieżka może złagodzić niektóre obawy związane z funkcjonalnością i operacjami. Zapewniłaby również ograniczone zakłócenia w działaniu środowiska jako całości, zmniejszając wpływ działań związanych z łataniem.

Oprzyrządowanie do ręcznego łatania

Niektóre narzędzia pozwalają administratorom wykonywać działania zawiązane z łataniem bez pełnej automatyzacji, z wykorzystaniem funkcji, które automatycznie pobierają i instalują łaty na podstawie rodzaju systemu. Jednym z przykładów może być użycie Ansible dla serwerów z systemem Linux lub Windows lub dla obu tych systemów. Administratorzy mogą wykorzystać strategie Ansible do pobrania i zainstalowania aktualizacji, pozostawiając jednak ręczną obsługę restartowania. Na rysunku 2.3 pokazano szczegóły działania Ansible w środowisku chmurowym wraz z wymaganiami dotyczącymi infrastruktury i sieci.



Rysunek 2.3. Jak działa Ansible

Dla systemu Windows można prosto utworzyć harmonogramy zadań (jeśli jesteś ze starej szkoły) lub wykorzystać do tego Ansible. Inne narzędzia, takie jak ManageEngine czy nawet SolarWinds, mogą także zarządzać łatami dla systemu Windows. Jednak SolarWinds ma swoją cenę i jeśli nie był jeszcze używany w danym środowisku do zarządzania serwerem, to należy wykorzystać możliwości nowszych serwerów Windows, w tym scentralizowane zarządzanie.

Inną płatna opcją jest Microsoft Configuration Manager, który — podobnie jak SolarWinds — stanowi świetne rozwiązanie, jeśli wykorzystywane są już jego inne funkcje. Ale korzystanie z niego wyłącznie do zarządzania łatami może być kosztownym rozwiązaniem. Każde rozwiązanie zależy od wymagań klienta, możliwości połączenia z Internetem w celu automatycznego pobierania łat oraz od tego, kto zarządza łataniem systemów.

Jeśli zespół jest doświadczony i rozumie niuanse środowiska, może nie być potrzebne kupowanie drogich rozwiązań łatania. Jeśli jednak zespół jest młody i mniej obeznany ze środowiskiem, korzystne może być rozszerzenie ręcznych działań o bardziej zautomatyzowane rozwiązania takie jak Ansible. Istnieje oczywiście kilka repozytoriów i odniesień open source, które administratorzy mogą wykorzystać do zmniejszenia wymaganej liczby zadań ręcznych. Rysunek 2.4 pokazuje przykład prostej strategii Ansible do zautomatyzowania instalacji łat.

```
---
- name: Patching Playbook
  hosts: all
  tasks:
  - name: Patching initiation
    shell: /var/tmp/update.sh
    async: 10
    poll: 0
  - name: Verifying server state once it's back
    wait_for:
     host: "{{ inventory_hostname }}"
     state: started
     delay: 60
     timeout: 600
     port: 22
    delegate_to: localhost
```

Rysunek 2.4. Przykład skryptu Ansible do zarządzania łatami

Zarządzanie automatycznym łataniem

Zautomatyzowane łatanie stanowi dużą korzyść dla każdego zarządzającego systemem. Niezależnie od tego, czy systemy wykorzystują kontenery, projekty deweloperskie, systemy operacyjne, oprogramowanie sprzętowe czy aplikacje, wszystko musi być łatane ze względu na fakt, że oprogramowanie nigdy nie jest „skończone". Większość organizacji ma harmonogram łatania oparty na wymaganiach korporacyjnych lub klienckich.

Na przykład organizacja może być zmuszona do łatania krytycznych podatności w ciągu 15 dni (około 2 tygodni), a poważnych — w ciągu 30 dni (miesiąc). Aby zastosować te ramy czasowe, w automatycznym łataniu należy dokładnie i skutecznie używać wymaganych łat. Ze względu na ogromną liczbę podatności publikowanych codziennie — zgodnie z obecnymi raportami jest to około 85 dziennie (*www.linkedin.com/posts/jgamblin_2023-ytd-cve-stats-total-number-of-cves-activity-7136354512683859968-YjfK?utm_source=share&utm_medium=member_desktop*) — administratorzy muszą redukować zadania ręczne, aby utrzymać działania związane z łataniem.

Inna organizacja może być zobowiązana do zgłaszania podatności lub stanu łatania do agencji zewnętrznej lub przygotowania ustaleń dla audytów. Przy takim scenariuszu

jeszcze cenniejsza jest jakaś automatyzacja, a także dokumentacja opisująca proces. Automatyzacja łatania dotyczy narzędzi i technik w takim samym stopniu jak uzgadniania procesów i praktyki.

Zautomatyzowane łatanie może nie tylko zmniejszyć ewentualne przeoczenia lub błędy wynikające z ręcznego planowania działań, lecz także pomóc w radzeniu sobie w tym, z czym boryka się większość organizacji, czyli brakiem talentów w dziedzinie cyberbezpieczeństwa i ograniczonymi zasobami.

Zalety łatania automatycznego w porównaniu z ręcznym

Jak widać na rysunku 2.5, istnieją znaczne korzyści wynikające ze stosowania automatycznych narzędzi i technik łatania w porównaniu z poleganiem wyłącznie na metodach ręcznych.

Rysunek 2.5. Korzyści łatania automatycznego z porównaniu z ręcznym

Zmniejszone narzuty administracyjne. Powszechnie wiadomo, że czas, jaki administratorzy, inżynierowie i deweloperzy mają na łatanie systemów, jest ograniczony. Takie zespoły zwykle żonglują różnymi zadaniami i projektami, jednocześnie pracując nad łataniem i naprawianiem podatności w ramach z góry ustalonego harmonogramu.

Szybsze i bardziej wydajne cykle łatania. Łatanie bez żadnej automatyzacji po prostu zabierze więcej czasu. Dodatkowy czas jest potrzebny na pobieranie i instalowanie łat, zwłaszcza w przypadku wielu różnych środowisk. W każdym złożonym środowisku zautomatyzowane łatanie jest koniecznością, aby nadążyć za wieloma łatami wypuszczanymi na poziomie systemów operacyjnych, aplikacji i kontenerów. W środowiskach deweloperskich będą zapewne poziomy deweloperski, UAT i produkcyjny.

Krótszy czas naprawy. Z uwagi na liczbę codziennie identyfikowanych podatności zespoły musiałyby bez automatyzacji pracować tygodniami lub miesiącami, aby naprawić podatności. Jednak przy wykorzystaniu dodatkowych narzędzi, skryptów lub strategii czas od identyfikacji podatności do jej naprawy znacząco się skraca.

Skrócenie czasu naprawy jest krytyczne, gdyż organizacje często ścigają się z napastnikami, którzy starają się wykorzystać nowe lub znane podatności szybciej, niż można je naprawić. Określenie okno *minimalizacji ataku* lub okno *wykorzystania* to rama czasowa, która może pomóc ograniczyć ryzyko naruszenia zabezpieczeń lub ujawnienia danych w wyniku incydentu naruszenia zabezpieczeń. Może to także wzmocnić argumentację organizacji, że postępuje ona z należytą starannością, łagodząc znane podatności i ryzyko.

Kombinacja łatania automatycznego i ręcznego

Jednak rzeczywiste rozwiązanie leży gdzieś pomiędzy praktykami ręcznymi a rozwiązaniami automatycznymi. Nie ma jednego rozwiązania odpowiedniego dla zespołu, zwłaszcza z uwagi na zawiłości na styku infrastruktury, działania i rozwoju. Jednak najlepszym sposobem jest zbudowanie szerokiego programu zarządzania łataniem, rozpoczynając od tworzenia macierzy ról i odpowiedzialności (ang. *Roles and Responsibilities Matrix*). To proste działanie pozwala zespołom zrozumieć, za co są odpowiedzialne, i rozpocząć tworzenie wokół tego narzędzi i praktyk. Każdy zespół może dopasować swoje działania związane z łataniem, co nie tylko zmniejsza ryzyko, lecz może także budować relacje między zespołami.

Zbudowanie kompleksowej strategii łatania uwzględniającej wszystkie zespoły pomaga im dostosować swoje cele i misję w odniesieniu do ogólnego programu zarządzania podatnościami. Każdy może być w stanie wykorzystać tę samą automatykę i wdrożyć te same dni/okresy łatania, aby ograniczyć czas przestoju i skrócić czas naprawy.

Równowaga między automatyzacją a ręcznym łataniem może być wspólnym doświadczeniem zespołów, w tym poprzez udostępnianie skryptów i strategii. Ogólnie rzecz biorąc, strategia jest tak samo ważna jak praca zespołowa. Współdzielenie technik automatyzacji z innymi zespołami, które mogą nie być tak dojrzałe w zarządzaniu łatami, pozwoli na rozwój zespołów, uzgodnienie strategii, a wreszcie zapewni lepiej zabezpieczone środowiska. Zaleca się tu stworzenie repozytorium zarządzania łataniem, w którym zespoły mogą dzielić się informacjami, zamiast pracować w silosach i potencjalnie przeszkadzać sobie w pracy.

Ryzyko przy automatycznym łataniu

Pewne ryzyko związane z automatyzacją ma więcej wspólnego z operacjami niż z zabezpieczeniami (patrz rysunek 2.6). Na przykład automatyzacja łatania i restartowania serwerów bez testowania może prowadzić do przerwania funkcjonalności lub nawet uszkodzenia w samych zestawach łat. Łaty mogą obejmować nie tylko funkcje zabezpieczeń, lecz także rozszerzenia i ewentualnie rozwiązywać inne problemy z oprogramowaniem.

Pierwszym ryzykiem jest wpływ na operacje i związane z tym przestoje dla użytkowników wynikające z łatania. Ponieważ łatanie jednej aplikacji może mieć wpływ na inną, może to spowodować, że przestanie ona działać lub nastąpi zakłócenie jej funkcjonalności. Dlatego tak ważne jest, aby zespoły testowały łaty w miarę możliwości w oddzielnym środowisku lub sporządziły plan wycofania przed rozpoczęciem łatania systemów.

Rysunek 2.6. Ryzyko przy automatycznym łataniu

Ponadto organizacja adaptuje strategie, takie jak wdrożenie Blue/Green, w których zmiany są wprowadzane do jednej wersji działającego systemu. Na przykład wersja zielona może symbolizować środowisko produkcyjne, a łaty są wprowadzane do środowiska niebieskiego, a następnie dokonuje się zmiany w konfiguracji, co powoduje przekierowanie ruchu do środowiska przejściowego, czyli niebieskiego, dzięki czemu staje się ono nowym środowiskiem produkcyjnym.

Dodatkowym ryzykiem związanym z łataniem jest fakt, że sama łata nie usuwa podatności i wymagane są zmiany konfiguracji. Przykładem może być łata Microsoftu naprawiająca podatność, wymagająca także zmiany reguł grupy Active Directory (AG) lub zmiany klucza rejestru. Ta złożoność może pozostawić otwarte podatności tylko dlatego, że instrukcje naprawy nie zawsze są jasne.

Wreszcie, jeśli zespoły nie mają doświadczenia w łataniu, mogą nie zapoznać się ze szczegółami CVE (ang. *Common Vulnerabilities and Exposures*), aby zrozumieć, czego potrzeba do pełnej naprawy. Jeśli brakuje wiedzy, jak działa łatanie, może to być początek dużych zaległości w zakresie podatności, które są niezwykle trudne do usunięcia.

Zarządzanie łataniem w środowiskach deweloperskich

Gdy organizacje przechodzą na coraz większe i zróżnicowane środowiska deweloperskie, ich łatanie staje się dla zespołów jeszcze większym priorytetem. Łatanie na poziomie deweloperskim obejmuje wszystko — poziom systemu operacyjnego (np. Red Hat Enterprise Linux, RHEL), warstwę aplikacji (np. Tomcat, Python i inne biblioteki) oraz inne narzędzia (np. Bootstrap, Java lub Jenkins).

Nowsze role takie jak DevOps oraz DevSecOps są szeroko rozpowszechnione i obejmują odpowiedzialność począwszy od tworzenia oprogramowania, przez inżynierię zabezpieczeń, po działania związane z infrastrukturą. Te nowe role wymagają dodatkowych umiejętności w każdej dyscyplinie i zapewniają harmonię i bezpieczeństwo.

Wszystko, co jest skierowane na zewnątrz, wiąże się z większym ryzykiem wykorzystania i powinno znajdować się na szczycie listy zarządzania łataniem. A przy każdej aplikacji będą występować elementy zewnętrzne wymagające dodatkowego testowania przed zainstalowaniem łat.

Łatanie oprogramowania open source

Większość deweloperów wykorzystuje w swoich aplikacjach pewien zakres oprogramowania open source (OSS). W istocie analizy pokazują, że nowoczesna baza kodów zawiera ogólnie od 60 do 80 procent kodu OSS. OSS pozwala zespołom skupić się na samych aspektach tworzenia i inżynierii, bez konieczności wymyślania na nowo każdego rodzaju funkcji. Na przykład po co tworzyć narzędzie do logowania, jeśli zespół może wykorzystać Log4j?

OSS zapewnia dużą elastyczność i zdolność do adaptacji, podczas gdy kupowanie oprogramowania lub jego tworzenie zabiera dodatkowy czas i zasoby. Przy napiętym budżecie i skondensowanym terminarzu sens ma wykorzystanie możliwie jak najwięcej bezpłatnego oprogramowania. Jednak łatanie OSS może być zdradliwe, zwłaszcza gdy wykorzystujemy narzędzia lub biblioteki, które nie są często aktualizowane.

Jak omawiamy w rozdziale 9., „Chmura, DevSecOps i bezpieczeństwo łańcucha dostaw", badania przeprowadzone przez organizacje takie jak Sypnosis wykazały, że 88 procent nowoczesnych baz kodu zawiera składniki OSS, które *nie* były rozwijane przez dwa lata, a 85 procent baz kodów zawiera OSS, które są przestarzałe od ponad czterech lat. Ten problem jeszcze bardziej komplikuje fakt, że 25 procent wszystkich projektów OSS ma *jednego* dewelopera wnoszącego swój wkład w kod, a 94 procent projektów ma dziesięciu lub mniej aktywnych deweloperów. W istocie dziewiąty roczny raport Sonatype dotyczący łańcucha dostaw oprogramowania (*https://www.sonatype.com/state-of-the-software-supply-chain/introduction*) wykazał, że tylko 11 procent całego oprogramowania open source jest aktywnie konserwowane.

Wiele projektów OSS jest szeroko wykorzystywanych, ale są konserwowane jedynie przez niewielką grupę deweloperów. Na przykład Log4j był zarządzany przez trzech deweloperów, a ponieważ produkt nie był płatny, nie mieli oni czasu i zasobów, aby zarządzać nim i go łatać. Gdy pod koniec 2022 roku pojawił się exploit Log4j, wówczas — ze względu na mały zespół i jego ograniczoną dostępność — uzyskanie łaty zajęło dużo czasu.

Choć poważne exploity takie jak Log4j nie występują przez cały czas, istnieje możliwość, że napastnik wykorzysta OSS do stworzenia złośliwego oprogramowania lub zaatakuje. Wiadomo, że te projekty nie zawsze są konserwowane, a ponieważ Log4j odniósł tak duży sukces, możliwe jest, że w innym oprogramowaniu istnieje inna okazja. Zespoły muszą być świadome możliwości wykorzystania każdego OSS, który nie jest właściwie konserwowany.

Niecałe oprogramowanie jest sobie równe

W tym podrozdziale omawiamy niektóre złożoności wewnętrznego zarządzania łataniem OSS i zmieniającymi się obowiązkami zespołów deweloperskiego i operacyjnego. Omawiamy także potrzebę jasnego podziału ról i obowiązków, takich jak odpowiedzialność za zarządzanie łatami.

Wewnętrzne zarządzanie łataniem

Pierwszą część wewnętrznego zarządzania łatami OSS w ramach zespołów deweloperskich, operacyjnych i zabezpieczeń stanowi zrozumienie, jakie aplikacje i biblioteki są aktywnie wykorzystywane. Tworzenie biblioteki aplikacji OSS pomoże zrozumieć, jakie projekty OSS są wykorzystywane i w jakim celu, kto za nie odpowiada oraz jaka wersja jest w którym systemie.

Kolejnym krokiem jest integracja łatania OSS ze zwykłym oknem zarządzania łataniem. Plany zarządzania łataniem nie dotyczą tylko oprogramowania związanego ze sprzedawcą, lecz także każdego OSS. W środowiskach systemów Linux i Windows łatwo można utworzyć scenariusz łatania za pomocą playbooka Ansible, aby pobrać i zainstalować te łaty.

Powtórzmy — niezależnie od tego, czy aplikacje związane są z dostawcą, czy są OSS, aspekty procesów i oprzyrządowania są tak samo ważne dla sukcesu programu zarządzania łataniem.

Ostatnim krokiem jest skanowanie i raportowanie podatności tych serwerów i kontenerów, aby upewnić się, czy łaty są poprawnie stosowane. Te raporty mogą zostać wykorzystane do określenia, czy dany składnik OSS nie został jakiś czas temu zaktualizowany lub czy są dostępne nowsze łaty. Zespoły zabezpieczeń mogą wtedy określić ryzyko dla wszystkich składników OSS, które przestały być konserwowane, aby ocenić alternatywy lub zdecydować o akceptacji ryzyka wynikającego z wykorzystywania tych składników.

Zarządzanie zależnościami OSS w nowoczesnych środowiskach jest bardzo złożone i żmudne, więc często określane jest jako „piekło zależności". Znakiem stosowania automatyzacji oprzyrządowania w branży jest rosnąca popularność takich narzędzi jak Dependabot (*https://github.com/dependabot*) oraz Renovate (*https://github.com/renovatebot/renovate*), które ułatwiają automatyczne aktualizacje zależności. Narzędzia te można zintegrować z istniejącymi przepływami pracy deweloperskiej i są one obsługiwane przez szeroko stosowane platformy deweloperskie takie jak GitHub.

Odpowiedzialność zespołów ds. infrastruktury i zespołów operacyjnych

W branży technologicznej od dawna toczy się debata między zespołami ds. spraw infrastruktury i zespołami operacyjnymi, kto jest za co odpowiedzialny. Zespoły ds. infrastruktury są zwykle odpowiedzialne za hosting i zarządzanie stanowiące fundament,

na którym oparte są wszystkie aplikacje, serwery i kontenery. Mogą także odpowiadać za część łatania, konfigurację zabezpieczeń, a nawet zarządzanie tożsamością i dostępem (ang. *identity and access management* — IAM). W centrach danych zespoły ds. infrastruktury zarządzają centrum danych, zaś zespoły operacyjne skupiają się na codziennej obsłudze serwerów i działaniach w górnej warstwie, w tym na rozwiązywaniu problemów z awariami serwerów, błędami aplikacji i na wszelkich problemach związanych z funkcjonalnością dla użytkownika. Zespoły te mogą być także odpowiedzialne za wysoką dostępność (ang. *high availability* — HA). Jednak ten paradygmat zmienia się wraz z nadejściem obliczeń w chmurze, infrastruktury jako kodu (IaC) oraz metodyk takich jak GitOps, które deklaratywnie zaopatrują infrastrukturę cyfrową i zarządzają nią przy użyciu praktyk wywodzących się z rozwoju oprogramowania. Coraz częściej branża przyjmuje mantrę „ty tworzysz, ty jesteś właścicielem", w ramach której wiele zespołów deweloperskich i inżynierskich jest odpowiedzialnych za systemy, które stworzyło, zaprojektowało i wdrożyło do produkcji. Istnieje też w ostatniej dekadzie nacisk na DevOps, co rozbija silosy między zespołami deweloperskimi i operacyjnymi.

Kto jest właścicielem zarządzania łataniem?

To podchwytliwe pytanie — kto jest właścicielem zarządzania poprawkami? I na jakim poziomie? Sprzęt, hosty, aplikacje, kontenery itd. — wszystkie będą potencjalnie wymagały od różnych osób przyjęcia odpowiedzialności za ich łatanie. Przykładem może być aplikacja, która korzysta z bazy danych. Zespół deweloperski może być zobowiązany do łatania i konfigurowania aplikacji, ale oddzielny administrator bazy danych (DBA) może być odpowiedzialny za bazę danych. Jeszcze inny zespół może być odpowiedzialny za łatanie na poziomie infrastruktury albo w systemie operacyjnym lub hoście, na których opierają się systemy. Przykładowo, gdy mamy łaty dla serwerów Oracle, to kto będzie je instalował?

Choć odpowiedź nie jest prosta, mogą to być administratorzy bazy danych, po prostu dlatego, że dokonują regularnej konserwacji bazy danych i powinni znać przewidywane zachowanie przed zastosowanie łat i po ich zastosowaniu. Oczekiwana wydajność systemu i jego działanie są ważne, aby ocenić zmiany, gdy się one dokonają. Jest więc możliwe, że gdy DBA zastosują poprawki, mogą współpracować z zespołem deweloperskim, aby sprawić, żeby funkcjonalność aplikacji nie uległa zmianie.

Każda sytuacja będzie inna, ale najlepszym sposobem na uruchomienie programów zarządzania łataniem jest zgranie zespołów ze strategiami łatania i koordynowanie wysiłków. Należy utworzyć macierz RACI (*Responsible*, *Accountable*, *Consulted*, *Informed*), znaną także jako **macierz odpowiedzialności**, i dokumentować role poszczególnych zespołów. Bez tego członkowie zespołów mogą wskazywać inny zespół jako odpowiedzialny za łatanie, co prowadzi do zamieszania i opóźnień w naprawie. Macierz RACI na rysunku 2.7 ilustruje przykład, jak zespoły tworzą własną macierz do zorganizowania procesu łatania.

Role	Łatanie systemu/ infrastruktury	Aplikacje łatające	Warstwa operacyjna systemu/ infrastruktury	Warstwa operacyjna aplikacji	Monitoring systemu/ infrastruktury	Warstwa aplikacji monitorujących
Operacyjne	R, A	I	R, A	I	I	nd.
Rozwój	I	R, A	I	R, A	nd.	I
Bezpieczeństwo	I	I	I	I	R, A	R, A

Działanie	Ikona
Osoba odpowiedzialna	R
Osoba nadzorująca	A
Osoba informowana	I

Rysunek 2.7. Przykład macierzy RACI do oddzielenia obowiązków infrastruktury i operacji

Podział obowiązków

Po utworzeniu macierzy RACI zespoły powinny rozważyć także klasyczną koncepcję podziału obowiązków. Zespoły IT i zabezpieczeń będą miały oddzielne obowiązki, aby zapewnić kontrolę i równowagę. Na przykład zespół ds. infrastruktury będzie obsługiwał podstawowe łatanie i bezpieczną konfigurację, zaś zespół ds. zabezpieczeń będzie uruchamiał skany podatności i oceniał, czy spełnione są wszystkie oczekiwania dotyczące bezpieczeństwa.

Zespoły IT i deweloperskie mogą współdzielić niektóre odpowiedzialności, jeśli chodzi o łatanie, ale jest też dodatkowy zbiór kontroli dokonywanych przez inny zespół w celu testowania i zapewnienia funkcjonalności w systemach. Łatanie na poziomie systemu operacyjnego może wpływać na aplikacje górnego poziomu lub bazy danych, więc ważne jest posiadanie wielopoziomowego planu zarządzania łataniem.

Rozdzielenie obowiązków daje dwie ważne korzyści przy każdej strategii zarządzania łataniem. Pierwsza jest taka, że nie tylko jedna osoba jest odpowiedzialna za każde zadanie związane z łataniem i sprawdzaniem poprawności każdego ustawienia. Dzięki temu członkowie zespołu nie są odpowiedzialni za każde zadanie i tym samym nie ryzykują zmęczeniem pracą, pominięciem łaty lub konfiguracji albo żonglowaniem zbyt wieloma zadaniami naraz.

Drugą korzyścią jest zwiększone dogłębne bezpieczeństwo przy użyciu wielu poziomów osób, które implementują i weryfikują fakt usunięcia podatności. Bez przynajmniej dwóch par oczu, aby potwierdzić, że łaty zostały wdrożone, możliwa jest porażka, której implementujący nie zauważy bez drugiego etapu weryfikacji. Można to zrobić za pomocą skanera podatności, wysyłającego raporty do zespołów operacyjnych i ds. zabezpieczeń. Inżynier ds. spraw zabezpieczeń lub analityk mogą następnie ocenić wyniki i powiadomić personel IT, jeśli jakieś łatanie się nie powiodło.

Narzędzia i raportowanie

Narzędzia są podstawą każdego programu zarządzania łataniem, niezależnie od tego, jaka technologia jest używana w danym środowisku. Raportowanie jest drugą częścią programu, która zapewnia użytkownikom dane dotyczące weryfikacji i podatności organizacji. Narzędzia i raportowanie współdziałają ze sobą w celu instalowania łat oraz zarządzania nimi i ciągłego monitorowania.

Powszechnie używane narzędzia do zarządzania łataniem takie jak Configuration Manager firmy Microsoft lub SolarWinds wymagają dodatkowej konfiguracji i mogą być kosztowne. Całe stanowiska inżynierskie są przeznaczone wyłącznie do zarządzania tymi narzędziami z uwagi na złożoność środowisk chmurowych i oprogramowania. Raportowanie można wykonywać za pomocą każdego z wymienionych wcześniej narzędzi, dostosowując je do potrzeb osób otrzymujących ten raport.

Na przykład inżynier ds. zabezpieczeń może chcieć widzieć każdą podatność, niezależnie od jej wagi lub możliwości wykorzystania. Jednak deweloper może być zainteresowany jedynie określonymi aplikacjami, a nie podatnościami po stronie infrastruktury.

Jeśli chodzi o raporty, to menedżer może być zainteresowany tylko aktywami w swoim obszarze odpowiedzialności (ang. *area of responsibility* — AOR), natomiast dyrektor może być zainteresowany tylko pięcioma najważniejszymi podatnymi systemami lub trzema najważniejszymi podatnościami najbardziej narażonymi na wykorzystanie. Każdy unikatowy zbiór danych może zostać wbudowany w narzędzie raportowania, aby role otrzymały swoje własne raporty. Takie rozwiązanie zmniejsza ilość mało znaczących danych w raportach, skracając czas potrzebny na naprawę lub poświęcony na podatności, które nie mają wysokiego priorytetu.

Łatanie przestarzałych systemów

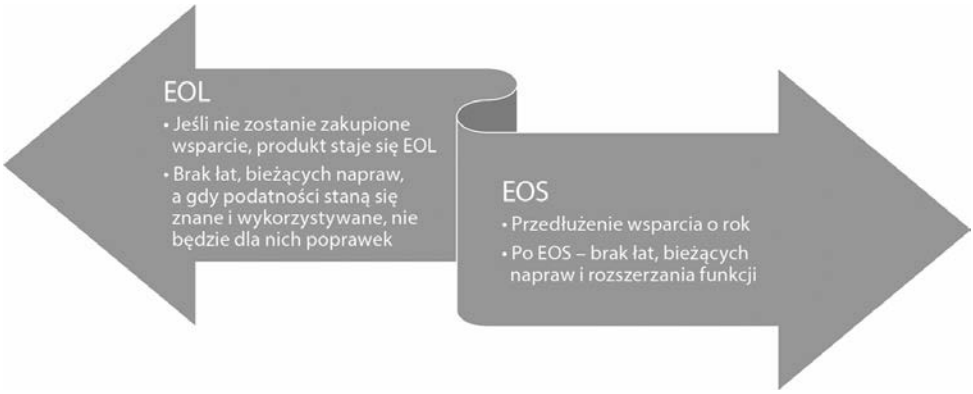
Może istnieć wiele powodów używania przez organizację oprogramowania, które nie jest już sprzedawane, czyli tzw. EOL (ang. *end-of-life*), lub przestało być obsługiwane, czyli EOS (ang. *end-of-support*). Takie oprogramowanie lub sprzęt nie są dalej łatane ani naprawiane przez producenta. Często zdarza się, że określony system operacyjny lub aplikacja są obsługiwane tylko przez określony czas, a potem są wycofywane, aby ustąpić nowym wersjom.

Jednym z przykładów jest wsparcie dla Windows Serwer 2012 do października 2023 roku. Po tej dacie klienci mogli zakupić od firmy Microsoft rozszerzone wsparcie na rok lub dwa, aby opracować plan przejścia na późniejszą wersję, np. Windows Serwer 2019 lub 2022. Ale jeśli klient nie wykupi takiego rozszerzonego wsparcia, pozostanie z potencjalnie podatnym oprogramowaniem, które nie będzie łatane.

Największym ryzykiem związanym z możliwymi do wykorzystania podatnościami są nieaktualne łaty. Jak wspomniano, większość organizacji ma ustalony harmonogram naprawy podatności w określonych ramach czasowych. Obok nieobsługiwanego oprogramowania kolejne ryzyko wiąże się z korzystaniem z oprogramowania open source, jeśli deweloperzy/konserwatorzy nie utrzymują i nie aktualizują swoich produktów i projektów. Każdy z tych scenariuszy prowadzi do nieznanych podatności i ryzyka.

Przestarzałe oprogramowanie

Organizacje muszą pozostawiać starsze oprogramowanie ze względu na wymagania klientów lub złożoność aplikacji. Możliwe jest w projekcie deweloperskim, że zespół będzie się ociągał z aktualizacją określonych bibliotek lub zależności ze względu na konieczność przeróbki kodu. Jednak jest to błędem. Im szybciej rozpocznie się planowanie aktualizacji lub migracji do nowego środowiska, tym szybciej zespoły zidentyfikują problemy funkcjonalności, które wymagają dodatkowego testowania. Na rysunku 2.8 pokazano przykłady list przestarzałego oprogramowania.



Rysunek 2.8. Przykłady list przestarzałego oprogramowania

Istnieje bardzo duże ryzyko związane z oprogramowaniem i sprzętem, które są nieobsługiwane lub wycofywane. Ponieważ te nieaktualne systemy nie są już łatane ani obsługiwane, nie ma możliwości naprawy błędów lub istniejących podatności, za to pozostają nieznane podatności. Wydłuża to listę nieznanego ryzyka, sprawiając, że organizacja jest potencjalnie podatna na popularne ataki na niełatane systemy, znane jako zdalne wykonywanie kodu (ang. *remote code execution* — RCE), nieautoryzowany dostęp lub inne podatności możliwe do wykorzystania.

Napastnik może wykorzystać RCE do wstrzyknięcia złośliwego oprogramowana lub złośliwego kodu do infrastruktury lub aplikacji sieciowych. Podatności RCE istnieją w różnych rodzajach aplikacji, systemach operacyjnych i urządzeniach, które są podatne na ataki i łatwiejsze do wykorzystania przez napastników w celu uzyskania początkowego dostępu do systemu.

Koniecznie trzeba mieć plan dla oprogramowania i sprzętu EOL i EOS, zanim oprogramowanie osiągnie kres życia. W przeciwnym razie organizacje mogą ponosić ryzyko związane z produktami EOL w środowisku podczas odchodzenia od starszego oprogramowania. Przykładem może tu być migracja aplikacji i usług na serwer RHEL.9.x przy jednoczesnym pozostawieniu starego środowiska na serwerze RHEL.7.x. Zwykle tego rodzaju projekty zajmują sporo czasu i wymagają zaangażowania kilku zespołów.

Niezałatane oprogramowanie open source

Ryzyko wynikające ze stosowania aplikacji lub pakietów OSS jest takie, że niektóre z nich nie są utrzymywane ze względu na ograniczoną liczbę deweloperów i zespołów, które zarządzają projektem open source. Każda aplikacja, biblioteka lub narzędzie będą zależne od tego, kto utrzymuje dane repozytorium. Kilka projektów open source rozpoczęło się, gdyż deweloperzy chcieli podzielić się swoją pracą ze światem i ułatwić innym wykonywanie powtarzalnych zadań.

Ale w związku z tym istnieje ryzyko — projekty te są dostępne za darmo i są utrzymywane, jeśli deweloperzy mają czas i chęć do pracy nad nimi poza swoimi normalnymi zajęciami. Jeśli OSS wygląda na nieaktualizowane od trzech miesięcy lub trzech lat, trzeba skontaktować się z właścicielami repozytorium, aby ustalić, czy jest ono aktywnie utrzymywane. Bez uzyskania odpowiedzi zespoły mogą zakładać, że wszelkie podatności znalezione w oprogramowaniu nie zostaną szybko załatane.

Biorąc pod uwagę to ryzyko, organizacja powinna mieć bibliotekę wszystkich używanych w środowisku aplikacji i komponentów OSS. Bez tego określenie ryzyka, a nawet ustalenie, gdzie te biblioteki i narzędzia są używane, może być długim procesem. Każdy zespół ds. zabezpieczeń powinien być świadomy, jakie OSS są używane i gdzie są zainstalowane oraz czy są one aktualnie wymagane. Większość organizacji boryka się z kompletną inwentaryzacją komponentów OSS używanych w ich środowiskach — czy dla wewnętrznie tworzonego oprogramowania, czy komponentów innych firm wykorzystywanych w używanym oprogramowaniu (samodzielnie hostowanym lub wykorzystywanym przez Internet jako oprogramowanie jako usługa — SaaS). Na przykład ocena incydentu Log4j przez CSRB (Cyber Safety Review Board) ujawniła, że niektóre amerykańskie agencje federalne wydają dziesiątki tysięcy skumulowanych godzin na same próby ustalenia, czy komponent Log4j znajdował się w ich systemach.

Wszystkie przestarzałe lub nieużywane OSS powinny zostać możliwie szybko zaktualizowane lub usunięte ze środowiska, co pomoże zminimalizować powierzchnię ataku aplikacji i oprogramowania do wykorzystania przez złośliwych aktorów.

Inna ważną kwestią różniącą OSS od oprogramowania autorskiego jest to, że z dostawcą mamy zwykle umowy o poziomie usług (ang. *service level agreements* — SLA) i harmonogramy związane z konserwacją oprogramowania, dostarczaniem aktualizacji i ujawnianiem podatności. W przeciwieństwie do producentów oprogramowania osoby konserwujące OSS nie są dostawcami oprogramowania, a OSS można zwykle używać bezpłatnie, co oznacza, że konsumenci nie mają pewności, czy podatności lub usterki są obsługiwane w jakichkolwiek formalnych i ścisłych terminach. Organizacje korzystające z OSS muszą być świadome tej różnicy i przygotowane na wdrożenie kompensacyjnej kontroli dla podatnego oprogramowania OSS bez dostępnych łat, a także na ewentualne przekierowanie kodu i samodzielne wdrażanie napraw. Mimo że takie są realia, większość organizacji nie rozumie dobrze tych różnic i dalej polega na osobach utrzymujących OSS, nie biorąc pod uwagę, że ostatecznie to one są odpowiedzialne za wszelkie OSS, z których korzystają.

Ryzyko szczątkowe

Ryzyko szczątkowe to wszelkie ryzyko, które pozostaje po działaniach związanych z zarządzaniem podatnościami, w tym stosowaniu łat i poprawek poza harmonogramem, które stanowią szybkie inżynierskie aktualizacje lub bezpieczne konfiguracje. Ryzyko szczątkowe może zostać przyjęte, może być ograniczane lub przekazywane innemu podmiotowi.

Ważne jest, aby pamiętać, że nie wszystkie łaty mogą zostać wdrożone we wszystkich systemach po prostu ze względu na wymagania klientów lub inne okoliczności. Ryzyko szczątkowe jest częścią każdego programu zarządzania podatnościami i musi być brane pod uwagę przy tworzeniu wyjątków, czyli akceptacji określonego ryzyka w środowisku.

Jednak najgorszą rzeczą, jaką może zrobić zespół, jest bycie przytłoczonym liczbą podatności lub pracą, jakiej będzie wymagało „nadrobienie czasu" lub naprawa wszystkich pozostałych podatności. **Paraliż decyzyjny** następuje wtedy, gdy jednostka nie jest zdolna do podjęcia na czas decyzji ze względu na ilość danych wprowadzanych jednocześnie. Oznacza to, że zespoły muszą być świadome tej kwestii i zapewnić odpowiednią przestrzeń na podejmowanie złożonych decyzji związanych z ryzykiem.

Powszechne ataki na niezałatane systemy

Według statystyk zamieszczonych w artykule agencji CISA istnieje poważna potrzeba skupienia się na zaktualizowanych praktykach zarządzania podatnościami w celu zmniejszenia ryzyka (*www.cisa.gov/news-events/news/transforming-vulnerability-management-landscape*). Pozostawienie systemów z podatnościami możliwymi do wykorzystania poszerza powierzchnię możliwego ataku. Przy większej powierzchni ataku napastnik ma wiele tras do systemu. Każda podatność, która pozostaje bez ograniczenia, jest możliwym punktem wejścia, który daje napastnikowi możliwość przeprowadzenia ataków z wykorzystaniem łańcuchów podatności. Ataki łańcuchowe na podatności są szczegółowo omawiane w rozdziale 7. „Łączenie podatności w łańcuch".

Inną częstą metodą ataku na niezałatane systemy jest celowanie w każdy nieuwierzytelniony dostęp logowania na serwery i doprowadzenie do naruszenia bezpieczeństwa, a następnie ataku eskalacji uprawnień. Te rodzaje ataków można znaleźć w różnym oprogramowaniu i nie ograniczają się one do jednego rodzaju aplikacji lub producenta.

Te dwa wektory ataku są szczególnie niszczycielskie, jeśli weźmiemy pod uwagę możliwość uzyskania zdalnego dostępu lub podwyższenia uprawnień w celu uzyskania dostępu do większej liczby systemów. Jeśli nie ma segmentacji sieci lub innych mechanizmów ograniczających, wykorzystanie tych podatności może pozwolić na dość łatwe przejście do innych systemów. Jednym z przykładów jest uzyskanie dostępu do serwera aplikacji sieciowych, a następnie przejście w bok do bazy danych lub innej usługi zawierającej informacje biznesowe lub finansowe. Nawet na tych kilku przykładach łatwo dostrzec, że niezałatane systemy stanowią masowy wektor ataku i mogą być często wykorzystywane do nieautoryzowanego dostępu do systemów.

Jednym z obszarów, który może bardzo pomóc przy niezałatanych systemach, jeśli nie można ich załatać ze względu na wymagania aplikacji lub przestarzałego systemu, są

kontrole ograniczające. Daje to dodatkową ochronę, ograniczając „promień wybuchu", czyli potencjalne szkody wyrządzane w systemach. Na przykład, jeśli jest wprowadzona segmentacja sieci w celu ograniczenia dostępu do baz danych, złośliwy aktor może włamać się na frontowy serwer sieci, ale nie będzie mógł wykorzystać go do uzyskania dostępu do bazy danych ze względu na ograniczone połączenie między nią a serwerem sieciowym.

Innym przykładem kontroli ograniczającej jest koncepcja najmniejszego uprawnienia, czyli ograniczenie liczby uprawnień, które mają użytkownicy w systemie. Ograniczenie głównego dostępu do serwerów Linuksa lub wyłączenie go wymaga, aby każdy użytkownik logował się ze znacznie mniejszymi uprawnieniami lub mniejszym dostępem niż administratorzy. Podczas gdy napastnik może nadal przeprowadzać eskalację uprawnień, będzie to dla niego znacznie trudniejsze bez wykrycia przez rejestrowanie i monitorowanie.

Ustalanie priorytetów działań związanych z łataniem

Każdy projekt IT będzie wymagał żonglowania operacjami, utrzymaniem oraz ciągłymi innowacjami i projektami inżynierskimi. Częsty przykład stanowi organizacja decydująca się na uaktualnienie wycofanej wersji RHEK, która jest ROL, do najnowszej wersji na nowych kontenerach. Inny przykład to aktualizacja serwerów Windows z 2016 do najnowszej wersji 2022. Te rodzaje migracji zwykle obejmują utrzymanie starego środowiska podczas tworzenia nowego.

Oczywiście te wszystkie sytuacje nie wiążą się ze zwiększeniem liczby pracowników potrzebnych do tworzenia nowych systemów. Ten sam zespół, który zarządza środowiskiem produkcyjnym, będzie także tworzył nowe kontenery i aplikacje. Ci administratorzy i deweloperzy będą musieli podwoić swoją pracę, aby nadążyć z łataniem w obu środowiskach.

Aby pomóc w określeniu priorytetów zarządzania łataniem, organizacja powinna w możliwie szerokim zakresie wykorzystywać automatyzację w celu ograniczenia liczby interwencji ręcznych i czasu potrzebnego administratorom i inżynierom. Jeszcze przez rozpoczęciem ustalania priorytetów zarządzania łataniem trzeba zrobić pełną inwentaryzację aktywów, aby zespoły wiedziały, które aktywa mają dla nich najwyższy priorytet.

Po pierwsze, każde łatanie systemu operacyjnego i aplikacji, które mogą zostać zautomatyzowane, powinno być przetestowane i wdrożone do produkcji, aby obsłużyć większość podatności jak najmniejszym wysiłkiem. Na tym etapie zespoły mogą także utworzyć priorytety dla systemów deweloperskiego i testowania, a następnie łatać systemy produkcyjne. Przekładem jest łatanie urządzeń w piątek rano, a potem — po tygodniu testowania — łatanie wszystkich systemów produkcyjnych w następny czwartek wieczorem.

Po drugie, organizacje powinny szukać aplikacji innych firm, które nie będą automatycznie wciągane do ich repozytoriów czy też systemu zarządzania łataniem. Administratorzy i inżynierzy powinni tworzyć alerty lub powiadomienia od producentów o tym, że wypuścili oni łaty. Te aplikacje powinny być wbudowane w proces zarządzania łataniem i zorganizowane wraz ze zwykłymi systemami łatania.

Wreszcie, proces ten powinien być stale monitorowany i udoskonalany w czasie. Zarówno zespoły operacyjne, jak i zespoły ds. zabezpieczeń powinny oceniać raporty podatności, aby łatanie następowało zgodnie z oczekiwaniami. W innym przypadku proces i oprzyrządowanie są dostosowywane.

Łatanie nie jest jednorazowym działaniem — musi być powtarzane i poprawiane. Tak długo jak zespoły będą rozumiały swoją odpowiedzialność w zakresie łatania, będzie to bardzo pomocne we włączeniu tych działań do ich codziennej rutyny.

Zarządzanie ryzykiem i łatanie

Programy zarządzania ryzykiem i łatanie nie powinny być realizowane odrębnie jako niezależne działania, czyli bez integracji między odpowiedzialnymi za nie zespołami. Niektóre większe organizacje mają oddzielne zespoły i programy do zarządzania ryzykiem. Może to wprawdzie działać w większym biznesie, ale mniejsze organizacje powinny prowadzić działania związane z zarządzaniem ryzykiem w koordynacji z ich zwykłymi regularnymi działaniami konserwacyjnymi i operacyjnymi.

Na przykład nie miałoby sensu ustalanie priorytetów w łataniu serwerów baz danych, które nie zawierają już wrażliwych lub zastrzeżonych danych, bez segmentacji sieci, gdy bazy danych o większym ryzyku są połączone z serwerami aplikacji sieciowych. Bez tej wiedzy odrębnie działające zespoły mogą usunąć priorytety innych działań związanych z łataniem, co potencjalnie może spowodować duże ryzyko dla organizacji.

Każdy zespół powinien rozważyć ryzyko związane z wszelkimi istniejącymi podatnościami i sposób, w jaki chce realizować swoje harmonogramy łatania. Jeśli zespół zdecyduje się na łatanie wszystkiego zewnętrznie i weźmie się najpierw za aktywa, może później zająć się łataniem infrastruktury i baz danych, czyli magazynów. Należy jednak pamiętać, że organizacje ochrony zdrowia nadadzą wyższy priorytet chronionej informacji o zdrowiu (ang. *protected health information* — PHI) niż serwerowi baz danych, który może być używany do testowania.

Ostatnia uwaga na temat zarządzania ryzykiem i łataniem: podobnie jak w przypadku zarządzania podatnością jest to proces ciągły i iteracyjny. Nie jest to jednorazowe ćwiczenie, ale musi być przeprowadzane poziomo i pionowo w całej organizacji.

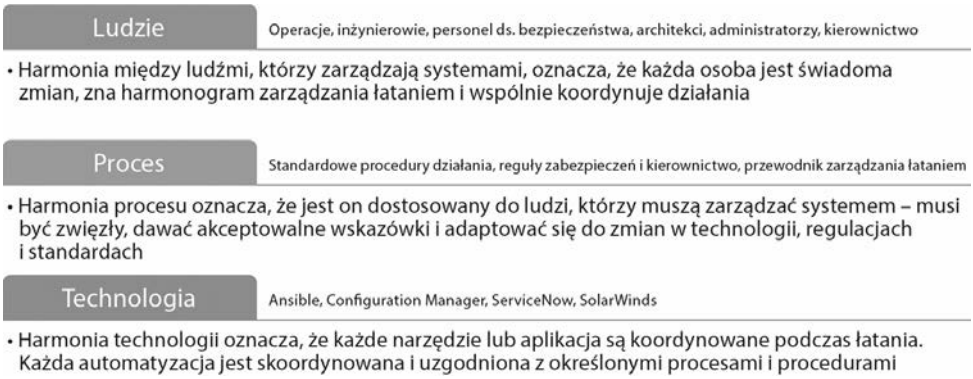
Tworzenie programu zarządzania łataniem

W tym rozdziale przedstawiono kilka elementów ryzyka, obaw i kwestii związanych z tworzeniem kompleksowego programu zarządzania łataniem. Najważniejszym tematem była przez cały czas klasyczna koncepcja ludzi, procesów i technologii. Powodem, dla którego ta koncepcja jest nadal tak powszechna, jest to, jak wyczerpująco ujmuje wszystkie aspekty organizacji. To sprawdza się przy tworzeniu programu zarządzania łataniem, gdyż klasyczne praktyki (tak jak samo ręczne łatanie) nie sprawdzają się.

Podstawą udanego programu jest harmonia między ludźmi (np. z działu deweloperskiego, operacyjnego czy zabezpieczeń), procesem (np. standardowymi procedurami działania — ang. *standard operating procedures*) lub regułami oraz technologią (np. menedżerami

konfiguracji lub skanerami podatności). Bez uzgodnienia tych trzech elementów organizacja naraża się na wszystkie wymienione wcześniej zagrożenia: exploity, podatności dnia zerowego i nieznane niewiadome.

Budowanie tej zgodności nie nastąpi z dnia na dzień, ale może być realizowane krok po kroku, aby stworzyć dojrzałość w każdej organizacji. Na rysunku 2.9 pokazano, jakie zespoły powinny być zaangażowane, jak integrować oprzyrządowanie i jak procesy mogą być spoiwem między tymi dwoma elementami podczas tworzenia lub dojrzewania programu zarządzania łatami.



Rysunek 2.9. Uzgodnienie między ludźmi, procesami i technologią

Ludzie

Aspekty związane z ludźmi, procesami i technologiami są omawiane w całej książce, ale ważne jest, aby już tutaj wiedzieć, kto jest odpowiedzialny za zarządzanie łataniem. Decyzja na ten temat jest konieczna, aby stworzyć udany system łatania w przedsiębiorstwie.

Możliwe jest, że za łatanie systemu będą odpowiedzialni właściciele systemu, administratorzy, deweloperzy, inżynierowie zabezpieczeń, a nawet osoby trzecie. Każdy zaangażowany w proces zarządzania łataniem musi rozumieć swoją obowiązki i harmonogramy stosowania łat.

Proces

Praktycy mogą mieć własny zestaw SOP, czyli procesów zdefiniowanych dla technicznej implementacji łat i związanych z nimi strategii lub skryptów. Ich ramy czasowe i harmonogramy mogą być zgodne z wymaganiami strony klientów lub innymi wytycznymi — na przykład muszą naprawiać krytyczne podatności w ciągu czterech dni.

Dla kadry zarządzającej i kierowniczej procesy mogą być bardziej zgodne z wymaganiami klientów lub nakazami federalnymi [w USA], które mają zastosowanie w ich dziedzinie biznesu. Organizacja zajmująca się ochroną zdrowia może mieć inne wymagania niż firma z branży IT. Zespoły kierownicze mogą tworzyć reguły, aby przekazywać je w dół innym osobom z kierownictwa i praktykom w celu wdrożenia metodami technicznymi.

Technologia

Technologia zarządzania łataniem zależy od decyzji osób odpowiedzialnych za regularne łatanie systemów i zarządzanie nimi. Obejmuje to wszelkie działania, od korzystania z narzędzi innych producentów, takich jak Configuration Manager lub ManageEngine, po tworzenie skryptów lub strategii za pomocą Ansible lub PowerShell.

Każdy z tych składników jest wybierany przez administratorów i — jeśli wymagają one finansowania — są zatwierdzane przez kierownictwo. W idealnym przypadku administratorzy i inżynierowie mają możliwość zarządzania wieloma systemami operacyjnymi, kontenerami lub aplikacjami za pomocą tego samego narzędzia. Jednak administratorzy i inżynierowie mogą potrzebować kreatywnych rozwiązań technicznych.

Podsumowanie

Organizacje muszą mieć kompleksową strategię zarządzania łataniem. Strategia ta obejmuje tworzenie inwentaryzacji aktywów, a także orientację w brakujących łatach i sposobie wdrażania łat w całym środowisku. Zarządzanie łataniem naprawdę wymaga harmonii między ludźmi, procesami i technologią. Organizacje powinny skupiać się na usuwaniu EOL i EOS, przygotowaniu się do regularnego łatania oraz tworzeniu planów wycofywania i wykonywania kopii zapasowych na wypadek, gdy coś pójdzie źle. Ponadto organizacje powinny mieć oczekiwane okna łatania i konserwacji.

Sprawdź, jak w erze chmury zarządzać ryzykiem informatycznym!

Musisz sobie uświadomić, że korzystanie z systemów opartych na chmurze wiąże się z cyberzagrożeniami — podobnie jak w przypadku tradycyjnej infrastruktury. Musisz się dobrze orientować w złożoności swojego systemu i w powiązaniach poszczególnych elementów. Dopiero na tej podstawie możesz planować i ustalać priorytety we własnym programie zarządzania ryzykiem, uwzględniając przy tym wiele innych zagadnień.

W tej praktycznej książce znajdziesz opis kompleksowych praktyk, dzięki którym współczesne organizacje utrzymujące złożone ekosystemy oprogramowania mogą skutecznie identyfikować podatności, zarządzać nimi i ograniczać ryzyko wystąpienia poważnych naruszeń bezpieczeństwa. Dowiesz się, dlaczego nie wystarczy po prostu „użyć łatki", aby naprawić znane luki w oprogramowaniu. Poznasz zasady profesjonalnego zarządzania podatnościami uwzględniające monitorowanie systemów i baz danych podatności. Przekonasz się, jak ważne są czynnik ludzki i identyfikacja czynników psychologicznych, które podczas interakcji użytkownika z oprogramowaniem przyczyniają się do powstawania podatności. W miarę lektury książki przyswoisz wydajne i skuteczne strategie, dzięki którym zapewnisz swojej organizacji wysoki poziom cyberbezpieczeństwa.

Najciekawsze zagadnienia:

- zarządzanie złożonymi środowiskami
- poprawki do oprogramowania i bezpieczna konfiguracja
- ocena podatności i łączenie ich w łańcuch
- czynnik ludzki w zarządzaniu podatnościami
- oprogramowanie bezpieczne już na etapie projektu
- budowa dojrzałego modelu zarządzania podatnościami

Chris Hughes jest współzałożycielem i dyrektorem generalnym firmy Aquia, specjalizującej się w usługach chmurowych i cyberbezpieczeństwie. Pracuje również w Agencji Cyberbezpieczeństwa i Infrastruktury (CISA).

Dr Nikki Robinson jest architektką do spraw bezpieczeństwa w firmie IBM i wykładowczynią na Capitol Technology University. Jest również członkinią Instytutu Technologii Infrastruktury Krytycznej (ICIT).

Helion
helion.pl
HELION S.A.
ul. Kościuszki 1c
44-100 Gliwice
tel.: 32 230 98 63
helion@helion.pl
KOD KORZYŚCI
Sięgnij po więcej!
ISBN 978-83-289-2160-3
9 788328 921603
Cena: 59,00 zł
WILEY