SKRYPT DO PRZEDMIOTU

Zabezpieczenie systemów i usług sieciowych

autor:

mgr inż. Michał Gryko

Gdańsk, 2015

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt *„Przygotowanie i realizacja kierunku inżynieria biomedyczna – studia międzywydziałowe”*
współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Spis treści

Wstęp

We współczesnym świecie coraz większa część naszego życia jest zależna od komputerów. Od zwykłej rozrywki przez nasze pieniądze, aż po zdrowie. Powierzamy technice coraz więcej obszarów naszej egzystencji. Ale czy kiedykolwiek zastanawialiście się co się stanie gdy dane które umieściliśmy w systemie informatycznym przestaną być dostępne? Kto ma do nich dostęp? Jak są zabezpieczone przed awarią sprzętu? Czy nasz dostęp do nich jest odpowiednio chroniony? Z pozoru te pytania mogą się niektórym wydawać przesadzone. Jednak nikt nie chciałby aby obca nam osoba przeglądała nasze prywatne zdjęcia lub czytała nasze e-maile, a tym bardziej miała dostęp do historii chorób i wyników badań naszej rodziny.

Aktualny trend prowadzi do gromadzenia coraz większej ilości danych na dużych grupach serwerów rozlokowanych na całym świecie. Dostęp do tych danych jest zazwyczaj zapewniany poprzez przeglądarkę internetową lub dedykowane aplikacje (np. urządzenia mobilne) komunikujące się z serwerami poprzez protokół HTTP. Mechanizmy te są już tak powszechne, że niewiele osób zastanawia się jak dokładnie działają oraz jaki wpływ na bezpieczeństwo naszych danych ma ich stosowanie.

Skrypt ten ma na celu przybliżyć czytelnikom problemy związane z zapewnieniem infrastruktury koniecznej do przechowywania i przetwarzania tych danych. Dużo uwagi poświęcono tu mechanizmom ochrony danych przed awariami sprzętu oraz zapewniania ich integralności. Wynika to z faktu, że aby móc skutecznie chronić dane przed atakami z zewnątrz musimy najpierw zacząć chronić je przed nami samymi. Zdecydowana większość przypadków utraty danych jest skutkiem awarii sprzętu lub błędu człowieka. Dobrze przemyślany system pozwala minimalizować ryzyko z tym związane.

Na koniec należy podkreślić, że zagadnienie zabezpieczania systemów i usług jest bardzo rozległe. Zebranie całej wiedzy na ten temat w jednej publikacji nie jest możliwe. Skrypt ten jest kierowany głównie do osób które nie zetknęły się wcześniej z tą tematyką. Wskazuje on jedynie podstawowe problemy związane z zabezpieczaniem danych i ma stanowić niejako wstęp do dalszych samodzielnych poszukiwań.

1. Wprowadzenie teoretyczne, czyli jak działa Internet?

Kluczowym elementem koniecznym do zapewnienia bezpieczeństwa systemów i usług jest zrozumienie podstawowego protokołu tworzącego sieć Internet, protokołu IP. W tym rozdziale dowiemy się jak wymieniane są dane w internecie. Prześledzimy także co musi się stać aby po wpisaniu w przeglądarce internetowej dowolnego adresu wyświetliła nam się strona WWW. Pozwoli to uświadomić sobie potencjalne wektory ataku których może użyć przestępca aby przejąć nasze dane. W dalszych rozdziałach czytelnik znajdzie typowe metody ochrony przed tego typu atakami.

1.1 Trochę o protokole IP

U podstaw działania sieci Internet znajduje się protokół IP (z j. ang. Internet Protocol). Jest to najbardziej rozpowszechniony protokół w sieci Internet. Pozwala na wymianę pakietów danych pomiędzy dwoma urządzeniami podłączonymi do sieci. Podstawą protokołu IP jest unikalny identyfikator (nazywany także adresem IP) każdego urządzenia. W wersji 4 protokołu (IPv4 jest obecnie najbardziej rozpowszechnioną wersją protokołu IP) identyfikator ten ma postać 32 bitowej liczby dla wygody człowieka zapisywanej jako 4 ośmiobajtowe liczby w reprezentacji dziesiętnej rozdzielone kropkami. Uważny czytelnik zapewne zauważył, że istnieje tylko 4294967296 unikalnych adresów IPv4. Dlatego też wersja 4 protokołu jest bardzo powoli wypierana przez wersję 6 (IPv6) w której adresy mają postać liczb 128 bitowych. Przykład adresu IPv4:

Postać czytelna dla człowieka: 153.19.40.40
Postać binarna: 10011001 00010011 00101000 00101000

Adresy IP zostały pogrupowane w podsieci i są przyznawane zainteresowanym podmiotom przez Internet Assigned Numbers Authority (w skrócie IANA). Granice podsieci wyznacza tzw. maska. I tak na przykład dla Politechniki Gdańskiej definicja podsieci wraz z reprezentacją bitową ma postać:

Adres sieci:	153.19.32.0	10011001 00010011 001 00000 00000000
Maska:	255.255.224.0	11111111 11111111 111 00000 00000000
Wildcard:	0.0.31.255	00000000 00000000 000 11111 11111111
Podsieć:	153.19.32.0/19	10011001 00010011 001 00000 00000000
Broadcast:	153.19.63.255	10011001 00010011 001 11111 11111111
Adres Min:	153.19.32.1	10011001 00010011 001 00000 00000001
Adres Max:	153.19.63.254	10011001 00010011 001 11111 11111110
Ilość adresów:	8190	

Maska ma postać ciągu jedynek dopełnionych od prawej zerami tak aby łączna długość wynosiła 32 bity. Wielkość maski (ilość jedynek) mówi nam o tym gdzie kończy się adres sieci, a zaczyna adres

urządzenia w tej sieci. Ostatni adres w danej sieci to tzw. adres rozgłoszeniowy (Broadcast) używany do komunikacji z wszystkimi urządzeniami w danej podsieci na raz, bez wskazywania konkretnego adresata. Taki podział pozwala na zdefiniowanie urządzeń (routerów) odpowiedzialnych za każdą podsieć. Dostawcy Internetu wymieniają się miedzy sobą informacjami o obsługiwanych podsieciach, dzięki temu nasze dane mogą wędrować miedzy komputerami podłączonymi do sieci różnych dostawców.

Zanim pakiet danych z naszego komputera trafi w miejsce docelowe może przewędrować przez wiele routerów różnych operatorów. My jako użytkownicy nie mamy najmniejszego wpływu na trasę jaką wybierze operator. Potencjalny atakujący ma w tym momencie kilka opcji:

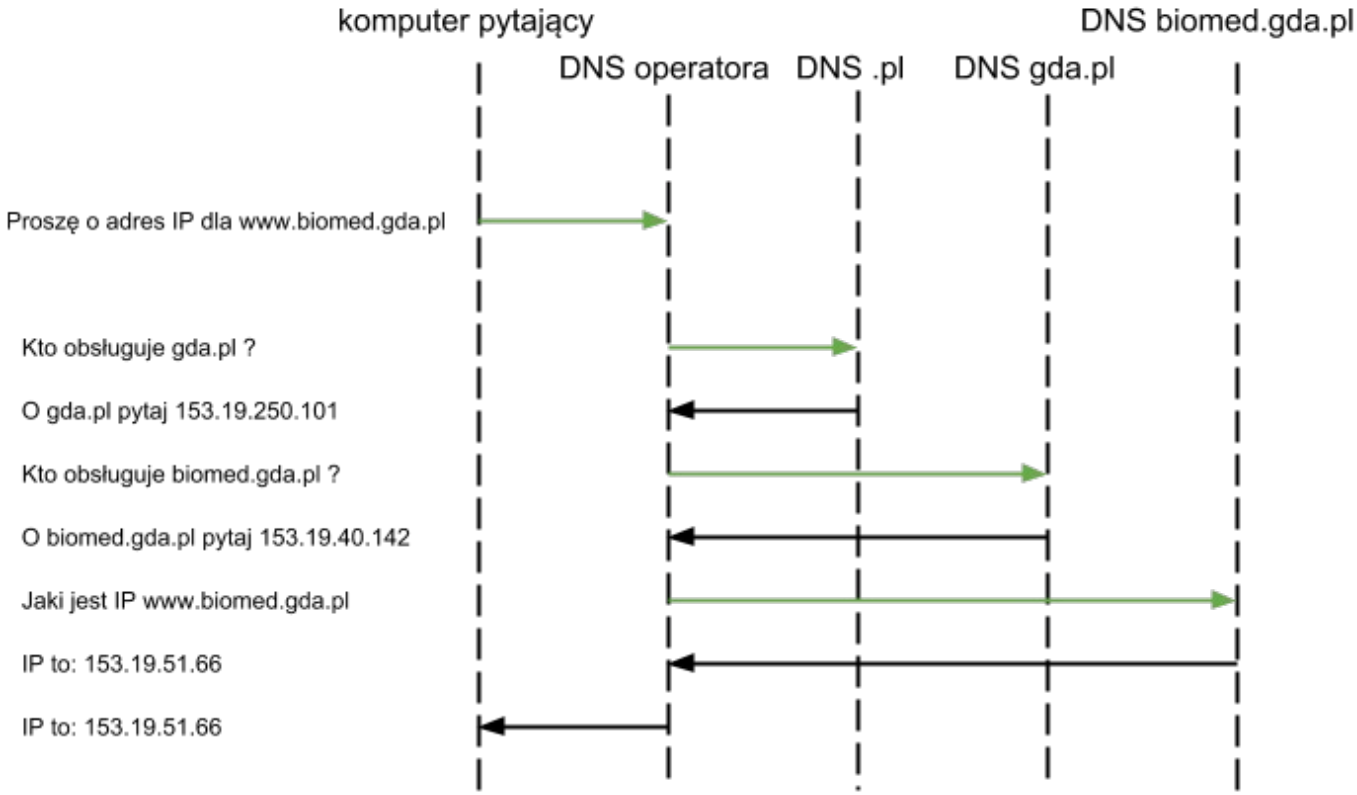
- Jeśli znajduje się w tej samej sieci lokalnej co my (np sąsiad korzystający z usług tego samego lokalnego operatora) mógłby próbować “wmówić” naszemu komputerowi, że jest routerem właściwym dla tej podsieci i cały ruch powinien być kierowany najpierw do niego. Dzięki temu byłby w stanie podsłuchiwać cały ruch do i z naszego komputera oraz dowolnie go modyfikować. W dalszej części skryptu omówimy jak się przed tym bronić oraz jak taki atak mógłby wyglądać w przypadku serwerowni w której znajdują się nasze serwery.
- Jeśli uzyska dostęp do urządzeń dowolnego z operatorów przez których sieci wędruje nasz pakiet może tak jak wcześniej rejestrować i dowolnie modyfikować cały ruch.
- Jeśli uzyska dostęp do kanału którym operatorzy wymieniają się informacjami o używanych podsieciach w zależności od różnych czynników może nawet przechwycić ruch całego kontynentu. Takie ataki pomimo, że brzmiące niewiarygodnie zdarzały się nawet w 2013 roku [1)] (wykaz linków znajduje się na końcu skryptu).

1.2 Trochę o rozwiązywaniu nazw

Protokół IP ma postać liczby, a my wpisujemy w przeglądarce adresy w postaci tzw. URL np. http://www.biomed.gda.pl. Za to skąd nasz komputer wie dokąd wysłać pakiety danych odpowiada system rozwiązywania nazw. Zazwyczaj odbywa się to z pomocą systemu DNS (z j. ang. Domain Name System). System DNS posiada strukturę hierarchiczną z podziałem na domeny globalne (np: .pl, .com, .net). Za poprawne działanie systemu odpowiada organizacja ICANN, nie zarządza ona jednak bezpośrednio procesem przyznawania domen. Poszczególne domeny globalne są przekazane podmiotom odpowiedzialnym za ich utrzymanie. Te z kolei umożliwiają zakup domen kolejnym podmiotom. Gdy stajemy się właścicielem domeny (np. kuchnia.pl) musimy podać sprzedającemu adresy IP serwerów DNS które będą obsługiwały naszą domenę. Adresy te zostaną dodane do globalnego systemu DNS i będą zwracane każdemu kto zapyta o naszą domenę. Muszą one odpowiadać pytającym adresami IP serwerów WWW odpowiedzialnych za naszą domenę. Dzięki temu możemy też utworzyć kolejne pod domeny (np. moja.kuchnia.pl lub www.kuchnia.pl co jest typową subdomeną dla większości domen widocznych w Internecie). Wszystkie zapytania o te subdomeny zostaną automatycznie przekierowane na nasze serwery DNS.

Gdy jakikolwiek komputer potrzebuje informacji o tym jaki adres IP kryje się pod daną nazwą domeny wysyła zapytanie do serwera DNS którego adres ma w swojej konfiguracji

(najczęściej jest to serwer dostawcy Internetu). Serwery dostawców nie obsługują jednak zazwyczaj żadnych domen tylko przekazują zapytanie dalej zgodnie z hierarchią domen. I tak dla przykładu zapytanie o www.biomed.gda.pl w uproszczeniu będzie miało następujący przebieg:



Jak widać jest to całkiem sporo zapytań dlatego też serwery operatorów zapisują sobie w pamięci odpowiedź i przez pewien czas odpowiadają na wszystkie zapytania o tą samą domenę z pamięci. Dzięki temu ruch do globalnych serwerów DNS jest znacznie mniejszy. Powoduje to jednak pewien problem na przykład przy zmianie IP serwera WWW. Do momentu aż wszystkie serwery DNS wszystkich operatorów nie zaktualizują adresu w swojej pamięci część użytkowników może otrzymywać stary adres IP. Taki stan może się utrzymywać nawet 72 godziny jednak w praktyce nie trwa to dłużej niż kilka godzin. Jest to uzależnione od konfiguracji naszego serwera DNS. Dodatkowo nasz komputer także zapisuje sobie w pamięci adresy IP domen o które pytał aby przyspieszyć ładowanie stron WWW oraz oszczędzić zasoby.

Potencjalny atakujący mógłby próbować wpłynąć na serwery DNS operatorów aby odpowiadały błędnymi adresami IP lub starać się podłożyć pytającemu fałszywą odpowiedź. Może także starać się wpłynąć na adresy zapamiętane w pamięci naszego komputera aby przekierować ruch z najczęściej odwiedzanych przez nas stron na swój serwer.

1.3 Podsumowanie

Gdy mamy już wszystkie potrzebne informacje aby skomunikować się z serwerem WWW możemy wreszcie wysłać żądanie otrzymania strony WWW. W odpowiedzi otrzymamy pliki które pozwolą naszej przeglądarce internetowej wyświetlić nam treść. Zagadnienie to jednak jest tak obszerne, że należało by mu poświęcić odrębną publikację. Jednak już w tym momencie wyraźnie widać, że to co wyświetliła nasza przeglądarka wcale nie musi być tym co serwer wysłał oraz, że nie mamy do końca pewności czy treść wysłał nam właściwy serwer. Widać też wyraźnie, że cała sieć Internet działa na zasadzie ogólnego zaufania. Pracownicy odpowiedzialni za utrzymanie sieci każdego z operatorów mogą uzyskać pełen wgląd w dane przesyłane ich siecią. Ma to związek z tym, że w początkach działania sieci Internet nikt nie przewidywał tak szybkiego jej rozwoju i skali jaką osiągnie. Była ona zarządzana przez grupę badaczy oraz pasjonatów i nikt nawet nie myślał o problemach związanych z bezpieczeństwem. Zaczęto zwracać na nie większą uwagę gdy poza instytucjami naukowymi i rządowymi dostęp do Internetu uzyskali także wszyscy zainteresowani na całym świecie. Pomimo upływu lat oraz wielu zmian i usprawnień walka o bezpieczeństwo naszych danych trwa po dziś dzień.

1.4 Pytania kontrolne do rozdziału pierwszego

Pyt 1. Przyporządkuj adresy IP do podsieci (wybierz właściwą odpowiedź).
Adres: 192.168.23.4 a) 10.0.0.0/24 b) 192.168.0.0/24 c) 192.168.0.0/16 d) 192.168.23.0/32
Adres: 10.2.2.2 a) 10.0.0.0/24 b) 192.168.0.0/8 c) 10.0.0.0/8 d) 10.0.0.2/16

Pyt 2. Czym jest maska sieci?
Pyt 3. Dlaczego serwery rozwiązywania nazw (DNS) są istotne ze względu na bezpieczeństwo danych?

2. (Nie)zawodność sprzętu komputerowego

Poznaliśmy już pobieżnie podstawy działania sieci Internet i niektóre zagrożenia z nich wynikające. Jednak zapewnienie bezpieczeństwa systemu komputerowego nie sprowadza się wyłącznie do kwestii związanych z transmisją danych. Nawet najlepiej zabezpieczony system zda się na nic kiedy serwery odpowiedzialne za jego działanie po prostu się zepsują. Dlatego równie ważne jest zapewnienie nieprzerwanego dostępu do usługi oraz dostępności i integralności przechowywanych danych. W tym rozdziale omówiona zostanie budowa serwera oraz rozwiązania konstrukcyjne zwiększające jego niezawodność. Na początku należy jednak wyraźnie zaznaczyć iż nie istnieją niezawodne systemy. Możemy jedynie minimalizować skutki awarii lub ukrywać ją przed użytkownikami systemu.

Typowym parametrem opisującym niezawodność elementów serwera jest MTBF (Mean Time Between Failures), czyli średni czas pomiędzy awariami. Ustala się go metodami statystycznymi na podstawie wyników badań testowej grupy elementów. Jest on wyrażany w godzinach i może mieć wartość nawet kilku milionów godzin. Nie znaczy to jednak, że element będzie pracował poprawnie przez tak długi okres czasu. Pozwala on nam jednak na wyliczenie prawdopodobieństwa wystąpienia awarii w danym okresie czasu.

Najważniejsza zasada budowy wysoko dostępnych systemów komputerowych to wyeliminowanie wszystkich tak zwanych pojedynczych punktów awarii. W przypadku sprzętu są to wszystkie elementy występujące pojedynczo, których awaria spowoduje brak dostępności danego serwera. Przykładem może być tutaj zasilacz. Dlatego w systemach serwerowych stosuje się redundancję (zwielokrotnienie) jak największej liczby elementów. Podnosi to znacząco koszt takiego systemu dlatego obecny trend to budowa aplikacji działających na wielu serwerach równocześnie. Sprawia to, że są one odporne na awarię pojedynczych serwerów. Dzięki temu możliwe jest zastosowanie mniej zaawansowanych, a tym samym tańszych serwerów. Obniża to znacząco koszt infrastruktury serwerowej przy zapewnieniu podobnej dostępności usług.

2.1 Elementy składowe typowego serwera

Lista elementów składających się na typowy serwer nie odbiega zbytnio od tej dla komputera domowego. Podobnie jak każdy inny komputer serwer posiada płytę główną, pamięć, procesor, kartę sieciową i zasilacz. Jest to absolutne minimum wymagane do uruchomienia serwera. Jednak zazwyczaj na tej liście są jeszcze: dyski twarde, kontrolery RAID oraz obudowy.

Tym co odróżnia elementy serwerowe od konsumenckich są zazwyczaj parametry pracy oraz jakość wykonania. Dyski serwerowe cechuje wyższa trwałość oraz prędkość zapisu i odczytu. Procesory posiadają dodatkowe jednostki odpowiedzialne za wspomaganie szyfrowania lub obsługę znacznych ilości pamięci operacyjnej. Specjalizowane kontrolery pozwalają na obsługę kilkunastu dysków twardych równocześnie. Poniżej przedstawiona zostanie charakterystyka poszczególnych elementów serwerowych.

2.2 Procesory do zastosowań serwerowych

W komputerach domowych mamy do czynienia głównie z jedną architekturą procesorów to jest x86/amd64 (x86_64). W przypadku systemów serwerowych wybór jest zdecydowanie większy. Poza amd64 (x86 jest już praktycznie niespotykane) mamy jeszcze do wyboru takie architektury jak PowerPC, Sparc czy ARM.

Każda z tych architektur ma swoje cech szczególne dające jej przewagę w niektórych obszarach zastosowań. Procesory amd64 są uznawane za najbardziej uniwersalne, są także najbardziej rozpowszechnione. Jednak wybór architektury procesora nie ma wpływu na podstawy bezpieczeństwa systemu dlatego też dokładny ich opis nie jest częścią tego podręcznika.

Procesory te mają też kilka innych cech odróżniających je od tych spotykanych w komputerach domowych i biurowych. Po pierwsze jakość wykonania, do procesorów serwerowych używane są najlepsze partie krzemu oraz przetestowane i dopracowane procesy technologiczne. Sprawia to, że układy te są dużo mniej awaryjne i przystosowane do nieprzerwanej wieloletniej pracy.

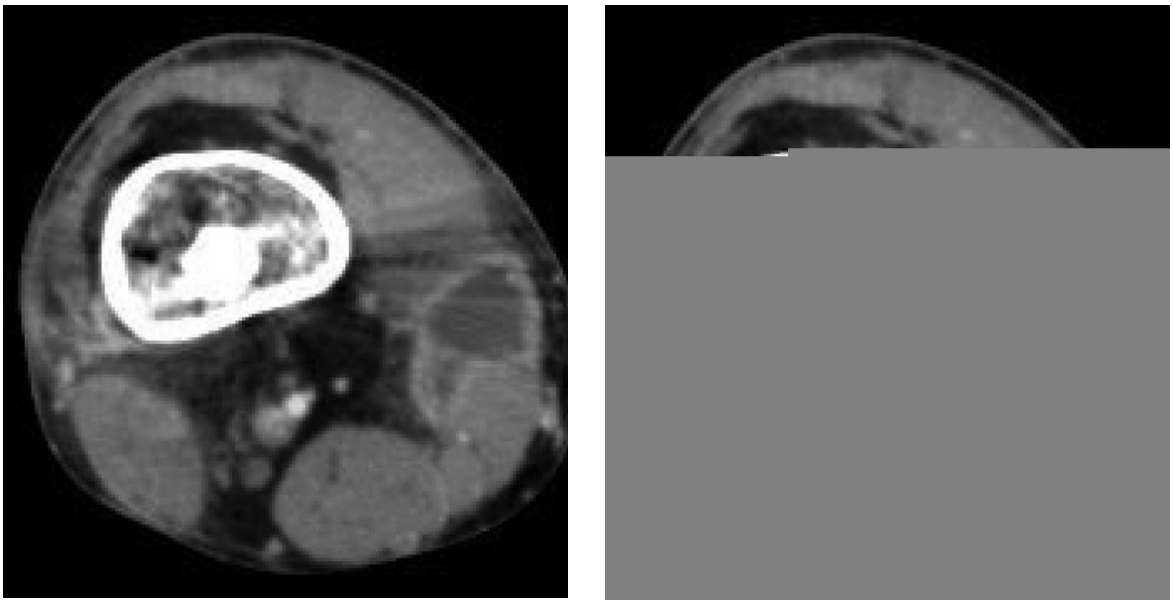
Inną cechą procesorów serwerowych jest zdolność do pracy w środowisku wieloprocesorowym. Zdecydowana większość domowych płyt głównych pozwala na zamontowanie tylko jednego procesora (może to być jednostka wielordzeniowa) dzieje się tak dlatego, że zwykłe procesory zazwyczaj nawet nie potrafią działać w konfiguracji wieloprocesorowej. W przypadku procesorów serwerowych często spotykane są rozwiązania dwu lub cztero procesorowe. W przypadku użycia procesorów 10 rdzeniowych daje to nawet 40 rdzeni w jednym serwerze.

Kolejne cechy to możliwość obsługi znacznej ilości pamięci. Serwery wyposażone w 265GB lub 512GB pamięci RAM nie są niczym niezwykłym. Dodatkowe podukłady wspomagające szyfrowanie, czy też zwiększona przepustowość operacji wejścia/wyjścia (I/O). Wszystko to sprawia, że układy te charakteryzują się niezwykłą wydajnością i stabilnością pracy. Dzięki temu w porównaniu do układów domowo-biurowych dużo lepiej niż nadają się do pracy w środowisku serwerowym.

2.3 Pamięci do zastosowań serwerowych

Ten element serwera na pierwszy rzut oka nie różni się tak bardzo od swojego konsumenckiego odpowiednika. Główną różnicą są wbudowane mechanizmy wykrywania i korekcji błędów (tzw. ECC) oraz dodatkowe rejestry pomiędzy układami pamięci, a kontrolerem (tzw. Registered memory). Dzięki dodatkowemu rejestrowi możliwe jest podłączenie większej ilości pamięci do jednego kontrolera ponieważ generują one mniejsze obciążenie elektryczne. Korekcja błędów natomiast zwiększa stabilność systemu oraz zapewnia integralność przetwarzanych danych.

Aby uzmysłowić sobie jak ważna jest korekcja błędów pamięci w przypadku danych medycznych poniżej przedstawiono ten sam plik w 2 wersjach, poprawnej oraz z celowym uszkodzeniem 1 bitu które nie zostało skorygowane.



Źródło: April C. Pettit , A. Alex Jahangir, and Patty W. Wright Author affiliations: Vanderbilt University School of Medicine, Nashville, Tennessee, USA (http://commons.wikimedia.org/wiki/File:Mycobacterium_doricum_Osteomyelitis_and_Soft_Tissue_Infection.jpg)

Pokazuje to wyraźnie jak istotne jest zapewnienie integralności przetwarzanych danych. W normalnym działaniu taki błąd występuje co prawda bardzo rzadko jednak przekłamanie wyniku badania (niekoniecznie zdjęcia) może nieść z sobą poważne konsekwencje.

2.4 Dyski twarde do zastosowań serwerowych

Dysk twardy jest jednym z najbardziej krytycznych elementów nie tylko serwerów ale także każdego komputera. Awaria procesora czy kości pamięci będzie skutkowała jedynie brakiem dostępności serwera, po wymianie uszkodzonego elementu na nowy maszyna może niezwłocznie wznowić pracę. W przypadku dysku twardego najcenniejsze są jednak dane które się na nim znajdują. Sama wymiana dysku na nowy nie wystarczy, konieczne jest również przywrócenie wszystkich danych które znajdowały się na uszkodzonym dysku. Bez tej operacji pomimo wymiany elementu na sprawny, serwer w dalszym ciągu nie będzie w stanie wznowić pracy. Często także ze względów wydajnościowych kopie zapasowe danych są tworzone raz dziennie, więc awaria dysku w ciągu dnia może nas bezpowrotnie pozbawić danych zgromadzonych od momentu ostatniego składowania kopii zapasowej.

Z tego powodu stosuje się tak zwane macierze dyskowe dzięki którym poprzez redundancję dysków możliwe jest zniwelowanie wpływu awarii pojedynczego dysku na pracę całego systemu.

Realizowane jest to poprzez użycie kontrolerów RAID (z j. ang. redundant array of independent disks). Zapewniają one możliwość podłączenia dysków w kilku konfiguracjach:
RAID 0 - jest to tak zwany stripping, pozwala na połączenie pojemności kilku dysków w jeden duży zasób widziany przez system operacyjny. Nie zapewnia on jednak żadnej redundancji, awaria jednego dysku skutkuje utratą wszystkich danych. Pozwala on dodatkowo zwiększyć prędkość odczytu oraz zapisu danych.
RAID 1 - tzw. mirror, jest to tryb w którym dane są zapisywane równocześnie na dwóch lub więcej dyskach. Wynikowy zasób posiada wielkość najmniejszego dysku i pozwala przetrwać awarię dysku bez większego wpływu na pracę systemu. Po wymianie uszkodzonego dysku na nowy kontroler wykona w tle operację kopiowania danych ze sprawnego dysku. Poza zwiększeniem niezawodności uzyskujemy także zwiększenie prędkości odczytu danych, gdyż możemy je odczytywać z wszystkich dysków jednocześnie.
RAID 10 - połączenie RAID 0 oraz RAID 1, poprzez połączenie 2 zasobów RAID 1 w jeden zasób RAID 0 uzyskujemy zwiększoną pojemność oraz niezawodność systemu.
RAID 5 - wymaga minimum 3 dysków, zasada działania jest podobna do RAID 0 z tą różnicą, że jeden dysk jest przeznaczany na zapis sum kontrolnych dla danych z pozostałych dysków. W przypadku awarii jednego dysku możliwe jest jego odtworzenie na podstawie informacji zgromadzonych na pozostałych dyskach. To podejście pozwala na zaoszczędzenie przestrzeni dyskowej ponieważ w przypadku 3 dysków tylko 33% łącznej pojemności jest tracone na zapewnienie redundancji. Podejście to ma też jednak kilka wad, po pierwsze nawet drobna zmiana danych wymaga rekalkulacji sumy kontrolnej. Odbudowywanie macierzy po awarii jest czasochłonne ponieważ wymaga odczytu danych z wszystkich dysków i dokonania stosownych obliczeń, ma to wyraźny wpływ na wydajność całego systemu. Pomimo wad jest to jednak rozwiązanie najczęściej spotykane ponieważ pozwala na zaoszczędzenie cennej przestrzeni dyskowej.

We współczesnych serwerowniach dyski nie są jednak umieszczone bezpośrednio w serwerach. Do składowania danych używa się dedykowanych macierzy dyskowych umożliwiających podłączenie kilkudziesięciu dysków twardych jednocześnie. Macierze te można następnie łączyć co pozwala na zbudowanie systemów wyposażonych w kilka tysięcy dysków twardych pracujących razem i widocznych z poziomu serwera jako jeden spójny zasób.

Kolejnym po niezawodności parametrem odróżniającym dyski serwerowe od konsumenckich jest prędkość ich działania. Czas dostępu do danych oraz maksymalna ilość operacji wejścia/wyjścia na sekundę (IOPS) to dwa bardzo istotne parametry wydajnościowe. W przypadku domowego komputera zazwyczaj nie wykonujemy zbyt wielu zadań równocześnie. Serwery często muszą natomiast przetwarzać żądania od kilkuset lub nawet kilku tysięcy klientów równocześnie.

Aby odczytać dane z tradycyjnego dysku konieczne jest ustawienie głowicy we właściwym miejscu oraz poczekanie aż dysk obróci się do miejsca w którym znajdują się dane. Głowica może być tylko w jednym miejscu na raz więc odczyt jest sekwencyjny i nie jest możliwe odczytanie danych z kilku miejsc jednocześnie. Widać wyraźnie, że prędkość obrotowa dysku ma duży wpływ na czas dostępu do danych, a tym samym na IOPS. Wpływ prędkości obrotowej na czas dostępu do danych przedstawia się następująco:

Prędkość obrotowa (rpm)	Średnie opóźnienie (ms)
15 000	2
10 000	3
7 200	4,16
5 400	5,55
4 800	6,25

Tabela 1: Związek prędkości obrotowej dysku ze średnim czasem dostępu do danych.

Typowe dyski konsumenckie mają prędkość 7200 obrotów na minutę, natomiast serwerowe 15000 obrotów na minutę. Pozwala to na dwukrotne zwiększenie wydajności jednak dyski takie są także znacznie droższe. Przedstawione w tabeli opóźnienie jest wartością średnią, ponieważ jak powszechnie wiadomo prędkość w ruchu po okręgu zależy od odległości od jego środka:

$$V=\omega \cdot r=2\pi f \cdot r$$

Widać wyraźnie, że fizyczne rozmieszczenie danych na dysku także ma znaczenie. W przypadku dysków 7200 obrotów dane znajdujące się w brzegowych sektorach mogą mieć czas dostępu zbliżony do 2ms, a więc podobny do średniego czasu dla dysków serwerowych.

Obecnie w systemach wymagających dużej wydajności dysków twardych (np. bazy danych) coraz bardziej popularne stają się dyski SSD (Solid State Disk). W dyskach tych wirujący talerz magnetyczny zastąpiono pamięcią półprzewodnikową. Pozwala ona na dostęp do dowolnego obszaru dysku w jednakowym czasie. Dzięki temu dostęp do losowych bloków jest równie szybki, a nawet szybszy niż dostęp sekwencyjny w dyskach tradycyjnych. Czas dostępu do danych dla dysków SSD to około 70µs, a więc prawie 30 krotnie szybciej niż w przypadku najszybszych dysków talerzowych. Możliwe jest, także czytanie wielu bloków równocześnie, co dodatkowo zwiększa wydajność. Wadą dysków SSD jest jednak ich pojemność, która nie przekracza zazwyczaj kilkuset gigabajtów. Czasem do wad zaliczana jest także ich ograniczona trwałość. Skutki tego zjawiska można jednak minimalizować poprzez stosowanie macierzy RAID, a wzrost wydajności jest tak znaczący, że problem ten można zazwyczaj uznać za pomijalny i zaakceptować ryzyko z nim związane.

2.5 Serwerowe płyty główne

Poza wymienionymi już cechami (wiele procesorów, duża ilość kości pamięci) serwerowe płyty główne posiadają także kilka innych istotnych cech. Współcześnie praktycznie każda taka płyta wyposażone jest w interfejs zdalnego zarządzania pozwalający na zdalne monitorowanie parametrów pracy maszyny. Umożliwia on także zazwyczaj włączenie i wyłączenie serwera oraz

zdalną instalację systemu operacyjnego. Płyta taka jest również wyposażona w wiele dodatkowych czujników temperatury oraz napięcia co pozwala na wykrycie wielu problemów sprzętowych zanim doprowadzą do całkowitej awarii systemu. Sam układ elementów na płycie sprzyja wydajnemu chłodzeniu podzespołów co jest konieczne w przypadku gęsto upakowanych centrów danych.

Płyta główna jest także jedynym elementem serwera którego nie jesteśmy w stanie zwielokrotnić. Dlatego też musi być wykonana z najwyższą starannością oraz przejść rygorystyczne testy co znacząco podnosi koszty produkcji.

2.6 Podsumowanie

W tym krótkim rozdziale poznaliśmy wpływ doboru elementów na stabilność i wydajność serwerów. Poznaliśmy także sposoby na zapewnienie większej integralności i bezpieczeństwa danych. Wszystko to jednak były rozwiązania sprzętowe wymagające znacznych nakładów finansowych. Należy tu także ponownie wspomnieć o stosowanych coraz szerzej rozwiązaniach czysto programowych. Ich idea polega na wytwarzaniu oprogramowania działającego na wielu serwerach jednocześnie w taki sposób, aby awaria pojedynczego serwera nie miała wpływu na pracę całej aplikacji. Podejście to wymaga jednak podporządkowania się pewnym ograniczeniom już na początku procesu projektowania aplikacji.

2.7 Pytania kontrolne do rozdziału drugiego

Pyt 1. Jaka jest główna metoda zapewnienia ciągłości działania serwera?
Pyt 2. Co odróżnia komponenty serwerowe od konsumenckich?
Pyt 3. Opisz krótko jedną wybraną konfigurację RAID. Podaj jaki ma ona wpływ na prędkość odczytu i zapisu danych w porównaniu z pojedynczym dyskiem.

3. Środowisko pracy oraz fizyczne bezpieczeństwo serwerów

Wiemy już jak chronić nasze dane przed awarią sprzętu. Jednak awaria nie jest jedyną przyczyną fizycznej utraty danych. Innymi równie częstymi przyczynami są pożary, powodzie, trzęsienia ziemi, czy też zwyczajna kradzież. Wiele średnich firm nie przykłada do tego problemu zbyt dużej wagi. Często serwery kluczowe dla poprawnego działania firmy są umieszczane w przysłowiowym schowku na szczotki gdzieś w piwnicy budynku. Dopiero przerwa w dostępnie do nich spowodowana na przykład pęknięciem rury uzmysławia jak bardzo istotne jest zapewnienie fizycznego bezpieczeństwa serwerów. Sporej części zagrożeń nie możemy całkowicie wyeliminować, możemy jednak znacznie zminimalizować ryzyko z nimi związane.

Na trwałość serwerów ma także wpływ środowisko ich pracy. W pomieszczeniu w którym są zlokalizowane musi panować odpowiednia temperatura oraz wilgotność. Serwery wydzielają znaczne ilości ciepła konieczne jest, więc zapewnienie wydajnej klimatyzacji. Utrzymywanie zbyt wysokiej temperatury może też prowadzić do zwiększenia częstotliwości występowania awarii sprzętu. Należy przy tym pamiętać, że nie ma konieczności chłodzenia całego pomieszczenia. Wystarczy dostarczyć zimne powietrze do miejsca z którego serwery pobierają je do chłodzenia. Takie podejście pozwala znacznie obniżyć koszty odprowadzania ciepła z serwerowni.

3.1 Środowisko pracy

Jak wspomniano we wstępie do tego rozdziału jednym z problemów przed którymi stają osoby odpowiedzialne za utrzymanie serwerowni jest wydajne odprowadzanie ciepła. Koszty chłodzenia stanowią zazwyczaj znaczy procent kosztów utrzymania serwerowni, dlatego każda optymalizacja tego procesu daje wymierne korzyści finansowe. Zazwyczaj zaleca się utrzymywanie temperatury w zakresie od 20 do 28 stopni Celsjusza. Przy czym należy pamiętać, że nie musi to być temperatura całego pomieszczenia a jedynie powietrza wlatującego do serwerów. Niektóre firmy utrzymują na wlocie powietrza do serwerów trochę wyższe temperatury (28°C - 32°C), daje to wymierne oszczędności energii użytej do chłodzenia. Jednak może prowadzić do częstszych awarii sprzętu. Nie zawsze jednak jest to istotny problem. Jeśli zwiększona temperatura skraca np. żywotność sprzętu do 3 lat to może to nie mieć negatywnego wpływu na koszt utrzymania serwerowni. Postęp w informatyce jest tak szybki, że wiele firm i tak wymienia sporą część swojego sprzętu na nowy co kilka lat. Pieniądze zaoszczędzone na koszcie chłodzenia pozwalają na częstszą wymianę sprzętu na nowy, a więc zazwyczaj bardziej wydajny i energooszczędny.

Kolejną bardzo istotną kwestią jest zapewnienie odpowiednio wydajnego i niezawodnego zasilania. Poza dostępem do kilku linii miejskich prowadzących do niezależnych i odległych od siebie punktów poboru konieczne jest także stosowanie zasilaczy awaryjnych (UPS). Zasilacze te mają za zadanie ciągłe dostarczanie zasilania do maszyn, w przypadku awarii linii miejskich energia jest pobierana z akumulatorów. Daje to czas na przełączenie zasilania na inną linię lub uruchomienie agregatów prądotwórczych bez zakłóceń w dostawie energii do serwerów.

Dystrybucja energii wewnątrz dużej serwerowni również nie jest sprawą prostą. Serwery cechują się zazwyczaj znacznym zużyciem mocy, a ich gęste upakowanie sprawia często konieczność dostarczenia kilku kilowatów mocy do każdej szafy serwerowej. Należy zawczasu przemyśleć procedury uruchamiania i zatrzymywania serwerów, ponieważ nagłe skoki poboru mocy mogą prowadzić do przepięć i awarii w systemie zasilania.

Niemniej ważną kwestią jest także dostęp do sieci Internet. Bez niego nawet najbardziej zaawansowana serwerownia jest zazwyczaj nieprzydatna. Konieczne jest doprowadzenie łączy światłowodowych z kilku punktów wymiany ruchu, tak aby awaria któregokolwiek z nich nie miała znaczącego wpływu na pracę naszej serwerowni. Zazwyczaj każdy punkt łączy się dwoma kablami światłowodowymi biegnącymi różnymi geograficznie trasami. Daje to pewność, że przy awarii jednego z nich spowodowanej np. pracami budowlanymi na jego trasie, łączność z punktem wymiany ruchu nie zostanie przerwana.

3.2 Lokalizacja

Niezależnie od położenia każda z lokalizacji ma swoje wady i zalety. Przed wybraniem lokalizacji powinniśmy, więc wykonać krótką analizę zagrożeń. Pozwoli nam ona na przygotowanie się do potencjalnych problemów które mogą nas spotkać. Nie ma tu znaczenia czy planujemy dopiero budowę lub adaptację budynku, czy też mamy już działającą instalację.

Serwerownia jest pomieszczeniem specyficznym, wymagającym sporej otwartej przestrzeni oraz dużej nośności stropu. Dlatego też im wyższe piętro tym mocniejsza musi być konstrukcja całego budynku. Lokalizacja na wyższych piętrach chroni przed zalaniem sprzętu na wypadek powodzi, ale znacznie zwiększa koszty budowy. W przypadku mniejszych instytucji potrzebujących zaledwie kilku serwerów nie ma to aż tak dużego znaczenia ponieważ spokojnie zmieszczą się w jednej szafie serwerowej. Jednak w przypadku większych instalacji musimy o tym pamiętać.

Odpowiednia lokalizacja może także obniżyć koszty chłodzenia. Jeśli temperatura na zewnątrz budynku jest odpowiednio niska możemy wspomagać nim pracę urządzeń chłodniczych. Dlatego zlokalizowanie dużej serwerowni w zimniejszej strefie klimatycznej przyniesie nam wymierne oszczędności. Nie możemy jednak zapominać o konieczności zapewnienia zasilania oraz dostępu do internetu. Wyklucza to zazwyczaj lokalizacje w których niskie temperatury utrzymują się przez cały rok.

Innym niemniej ważnym czynnikiem jest odległość pomiędzy serwerownią a naszymi użytkownikami. Każdy kolejny kilometr zwiększa opóźnienia transmisji ponieważ sygnał ma do pokonania dłuższą drogę. Wartości te nie wydają się duże (ok 10ms na każde 1000km) jednak porównując to do prędkości pracy współczesnego procesora (0.3ns na cykl) transmisja z Nowego Jorku do Londynu trwa wieczność.

3.3 Ochrona i kontrola dostępu

Gdy zdecydowaliśmy się już na lokalizację zapewniającą nam optymalne środowisko oraz niskie ryzyko katastrof naturalnych, przyszedł czas na odpowiednie zabezpieczenie sprzętu przed innymi ludźmi. Wykradzenia danych przedsiębiorstwa może przynieść wymierne korzyści. W przypadku gdy mamy do czynienia z firmą opracowującą nowe technologie, musimy pamiętać, że dane zawarte w ich systemie informatycznym mają realną wartość. Jeżeli opracowanie nowej technologii wymagało 2 lat pracy dziesięciu dobrze opłacanych inżynierów oraz dodatkowych nakładów na przygotowanie prototypów, to wykradzenie tych danych może oszczędzić konkurencji wiele milionów dolarów oraz dwa lata pracy.

Przy braku odpowiednich zabezpieczeń prostszym od ataku zdalnego mogło by się okazać wejście do siedziby firmy i wyniesienie nośników z danymi. Dlatego też teren serwerowi powinien być ogrodzony i odpowiednio chroniony. Jeżeli nie jest to odrębny budynek to bezwzględnie powinno być to wydzielone pomieszczenie, przeznaczone tylko i wyłącznie do tego celu. Każde wejście do niego powinno być rejestrowane, a dostęp osób postronnych ograniczony do niezbędnego minimum.

Sprzęt wewnątrz samej serwerowni zazwyczaj umieszczany jest w szafach teletechnicznych. Poza ułatwieniem montażu i utrzymania serwerów oferują one możliwość zamknięcia szafy bez ograniczania przepływu powietrza. Utrudnia to dodatkowo dostęp do samych maszyn osobom znajdującym się w pomieszczeniu (np. personelowi sprzątającemu).

3.4 Potencjalne wektory ataku na warstwę sprzętową

Poznaliśmy już problemy z którymi borykają się osoby odpowiedzialne za utrzymanie infrastruktury serwerowej. Teraz skupimy się na metodach których mógłby użyć potencjalny włamywacz w celu wykradzenia danych. Ograniczymy się tu wyłącznie do typowych ataków na warstwę sprzętową. Ataki na oprogramowanie zostaną omówione w kolejnych rozdziałach.

3.4.1 Podłączenie dodatkowego sprzętu

Jedną z najczęściej wykorzystywanych metod jest podłożenie dodatkowego sprzętu do siedziby atakowanego. Metoda ta może mieć wiele różnych wariantów i dotyczyć różnych części naszej infrastruktury.

Jeśli celem ataku jest konkretna osoba (np. prezes lub główny inżynier) atakujący może podszywając się pod jednego z dostawców przysłać mu prezent w postaci akcesorium do komputera. Może być to na przykład nowa bardzo wygodna klawiatura wyposażona dodatkowo w moduł radiowy wysyłający informacje o wciśniętych klawiszach do atakującego. Przy obecnej miniaturyzacji modemów z powodzeniem można do tego celu użyć nawet sieci komórkowej. Dzięki temu możliwe będzie poznanie loginów i haseł dostępowych. Metoda jest całkowicie

niewykrywalna z poziomu programów antywirusowych ponieważ nadajnik nie jest widoczny z poziomu systemu operacyjnego.

Jeśli celem ataku jest infrastruktura informatyczna firmy w ogólności atakujący może próbować podłączyć do sieci firmy swój komputer. Należy przy tym pamiętać, że zazwyczaj sieć wewnętrzna nie jest chroniona i kontrolowana tak dokładnie jak jej styk z siecią Internet. Samo pojęcie komputera też odbiega w tym przypadku od tego z czym spotykamy się na co dzień. Może być on zamaskowany jako drobny przełącznik sieciowy (switch) lub inne nie wzbudzające podejrzeń urządzenie. W celu umieszczenia urządzenia w firmie atakujący może zatrudnić się jako pracownik ochrony, personel sprzątający lub podać się za przedstawiciela jednego z dostawców który ma usunąć usterkę.

Ryzyko takiego ataku jest tym większe im bardziej wartościowe dane można w jego skutku uzyskać. Dobrze umieszczone i zamaskowane urządzenie może pozostawać niewykryte przez wiele miesięcy. Dlatego bardzo istotne jest szkolenie pracowników w tym zakresie oraz posiadanie mechanizmów automatycznego wykrywania i inwentaryzacji sprzętu podłączonego do naszej sieci. Pozwoli to na szybkie wykrycie i usunięcie nieautoryzowanego sprzętu. Jeszcze lepszym rozwiązaniem jest uwierzytelnianie urządzeń na poziomie przełączników sieciowych. Dzięki temu atakujący nie uzyska dostępu do naszej sieci wewnętrznej nawet na krótką chwilę. W przypadku stacji roboczych możliwa jest blokada portów USB oraz zezwolenie na podłączenie tylko wybranych i sprawdzonych urządzeń. Funkcje takie oferuje większość pakietów antywirusowych w wersji dla firm i instytucji.

3.4.2 Kradzież sprzętu

Jest to najmniej subtelna i zarazem najbardziej widoczna metoda. Wyniesienie dysków twardych z komputerów zostanie bardzo szybko wykryte, jednakże dane zostaną skradzione. Wartość samego sprzętu jest najczęściej wielokrotnie niższa niż wartość danych które się na nim znajdują.

Odpowiednio solidne zamki, kontrola dostępu oraz ochrona powinny skutecznie zapobiec tego typu atakom. Gdyby jednak ryzyko związane z utratą danych było zbyt wysokie można zastosować pełne szyfrowanie danych znajdujących się na dyskach twardych i innych nośnikach. Dzięki temu nawet w przypadku kradzieży nośnika dane które się na nim znajdują pozostaną bezpieczne.

3.4.3 Kradzież nośników z kopią zapasową danych

Metoda ta jest bardzo zbliżona do poprzedniej. Różnica polega na tym, że zniknięcie nośnika z kopią zapasową może pozostać nie zauważone przez dłuższy czas. Często także w firmach istnieją procedury wymagające przechowywania nośników z kopią zapasową poza terenem firmy tak aby zminimalizować ryzyko utraty danych na skutek lokalnego kataklizmu. Atakujący ma więc więcej potencjalnych punków ataku.

Obrona wygląda dokładnie tak samo jak w poprzednim przypadku. Odpowiednia ochrona oraz szyfrowanie danych znajdujących się na nośnikach powinno znacznie zminimalizować ryzyko wykradzenia informacji.

3.5 Podsumowanie

W rozdziale tym przedstawiono problemy stojące przed osobami odpowiedzialnymi za projektowanie i utrzymanie infrastruktury informatycznej. Począwszy od doboru odpowiedniej lokalizacji, poprzez zapewnienie właściwego środowiska pracy, na zabezpieczeniu przed atakami na warstwę sprzętową kończąc. Jest to jedynie pobieżne przedstawienie tego zagadnienia, gdyż dokładniejsze rozwinięcie tego tematu wykracza znacznie poza zakres przedmiotu. Zainteresowanych odsyłam do literatury dodatkowej której wykaz znajduje się na końcu skryptu.

3.6 Pytania kontrolne do rozdziału trzeciego.

Pyt 1. Jaki wpływ na koszt utrzymania serwerowni może mieć jej lokalizacja?
Pyt 2. Jakie są podstawowe metody obrony przed atakami na warstwę sprzętową?

4. Kontrola parametrów pracy serwerów

Pomimo pełnej redundancji elementów serwera oraz odpowiedniego przygotowania aplikacji na awarię, konieczne jest również monitorowanie jego parametrów. Redundancja ma sens jedynie wtedy gdy natychmiast wymienimy uszkodzony element na nowy. Jednak bez odpowiedniego systemu monitoringu i powiadamiania możemy przegapić awarię np. dysku twardego, gdyż cały serwer zdaje się pracować poprawnie.

Monitorowanie parametrów pracy odgrywa także istotną rolę w wykrywaniu ataków na naszą infrastrukturę. Nagły wzrost ruchu lub ilości błędnych żądań http może być pierwszą oznaką ataku. Dzięki wczesnemu wykryciu problemu możemy znacznie zminimalizować ewentualne szkody lub całkowicie im zapobiec.

4.1 Jakie parametry możemy monitorować

Współczesny sprzęt daje nam możliwość monitorowania praktycznie każdego parametru naszego serwera. Funkcjonalność ta została rozbudowana do tego stopnia, że aktualnie samo zbieranie danych o parametrach pracy serwerów wymaga kilku lub nawet kilkunastu szybkich serwerów. Poza parametrami udostępnianymi przez sprzęt sporo danych możemy także uzyskać z systemu operacyjnego oraz naszej aplikacji. Do podstawowych parametrów które należy monitorować zaliczamy:

- temperatura na wlocie i wylocie serwerów
- temperatura procesora, pamięci, dysku twardego
- awarie podzespołów
- stan zasilaczy UPS, parametry napięcia
- aktualny pobór prądu
- dostępność maszyn i usług
- obciążenie procesora
- zapełnienie przestrzeni dyskowej
- ilość operacji wejścia/wyjścia dla dysków twardych
- użycie pamięci ram
- ilość użytkowników obsługiwanych przez dany serwer
- ilość danych przesyłanych do i z serwera
- liczniki i alarmy zaimplementowane w naszych aplikacjach

Jest to jedynie podstawa lista parametrów, gdyż aktualnie możliwe jest sczytywanie wartości z setek różnych liczników dla każdego serwera. Sam procesor oferuje szereg liczników dotyczących ilości instrukcji na cykl czy skuteczności pamięci cache. O tym czy i jak często chcemy zbierać te dane decydujemy my. Zebranie zbyt wielkiej ilości danych nie przyniesie nam niczego dobrego gdyż będzie wymagało znacznej mocy obliczeniowej oraz bardzo skomplikowanego

sytemu automatycznej analizy danych. Przydaje się jedynie w przypadku wystąpienia problemów z wydajnością aplikacji lub chęcią jej optymalizacji, więc warto zostawić sobie możliwość dynamicznego dodawania liczników do sytemu monitoringu. Nie ma jednak nic gorszego niż częste fałszywe alarmy. Prowadzą one do zmęczenia administratorów, a w następstwie do pojawienia się nawyku ignorowania alarmów z systemu. Dlatego do systemu powiadamiania starajmy się dodać jedynie te parametry które bezpośrednio zagrażają bezpieczeństwu i stabilności naszej aplikacji.

4.2 Źródła danych

Wiemy już jakie parametry chcielibyśmy monitorować. Warto zastanowić się teraz skąd uzyskamy potrzebne nam informacje oraz w jaki sposób je sczytamy. Informacje o stanie sprzętu oraz środowisku pracy otrzymamy zazwyczaj poprzez kartę zdalnego zarządzania (OBM - out-of-band management). Karty takie są standardowo montowane w większości sprzedawanych obecnie serwerów. Informacje są zazwyczaj udostępniane poprzez protokół SNMP lub IPMI. Protokoły te są ogólnie przyjętym i implementowanym standardem. Poza udostępnianiem informacji, karty takie pozwalają również na zdalne sterowanie pracą sprzętu. Możemy na przykład włączyć lub wyłączyć serwer, zainicjować instalację systemu operacyjnego, podłączyć się zdalnie do portu szeregowego lub przesłać obraz z wyjścia wideo.

Karty zdalnego zarządzania informują nas o stanie sprzętu jednak nie przesyłają zazwyczaj informacji o jego wykorzystaniu. Wiele komponentów serwera udostępnia możliwość odczytu parametrów ich pracy. Istotne dla bezpieczeństwa danych są informacje o stanie dysku twardego (system S.M.A.R.T.), liczniki kart sieciowych oraz wskazania czujników temperatury. Parametry te odczytujemy z poziomu systemu operacyjnego, zazwyczaj poprzez dedykowane oprogramowanie.

Innym źródłem informacji są te udostępniane przez sprzęt sieciowy oraz systemy chłodzenia i zasilania. Dzięki nim możemy łatwo uzyskać globalny obraz naszej serwerowni. Pozwala to lepiej planować jej utrzymanie i rozbudowę.

Ostatnim lecz niemniej ważnym źródłem są mechanizmy wbudowane w system operacyjny oraz naszą aplikację. Są to źródła czysto programowe i mogą być dowolnie modyfikowane oraz rozbudowywane. Jednak zbyt duże ich skomplikowanie może prowadzić do spadku wydajności całego systemu. Pozwalają nam monitorować wydajność naszej aplikacji oraz zużycie zasobów serwera. Nie wymagają one żadnego dodatkowego wsparcia w sprzęcie. Należy pamiętać, że w przypadku oprogramowania każda dodatkowa instrukcja do wykonania, a taką jest na przykład inkrementacja licznika pomiarowego prowadzi do wydłużenia czasu wykonywania programu. Ma więc to bezpośredni wpływ na wartości pomiarowe.

4.3 Narzędzia monitorujące

Na rynku istnieje bardzo wiele specjalistycznych narzędzi pozwalających na zbieranie i analizę informacji o naszej infrastrukturze. Wiele z tych narzędzi jest dostępnych za darmo wraz z pełnym kodem źródłowym. Dzięki temu możemy w pełni dostosować narzędzie do wymagań naszej infrastruktury. Poza zbieraniem danych oferują one możliwość powiadamiania administratorów w przypadku gdy któryś z parametrów przekroczy zadaną wartość. Pozwala to na szybką reakcję i rozwiązanie problemu zanim zauważą go nasi użytkownicy.

Oprogramowanie takie składa się zazwyczaj z dwóch elementów. Agenta monitorującego instalowanego na serwerach oraz centralnej aplikacji przetwarzającej dane nadsyłane przez agentów. Zadaniem agenta jest zebranie wszystkich parametrów i przesłanie ich dalej, nie przeprowadza on analizy zebranych danych. Aplikacja centralna (serwer monitoringu) odpowiada za przetwarzanie i składowanie danych. Powinna ona analizować dane pod kątem przekraczania zadanych wartości progowych oraz wysyłać komunikaty do administratorów. Dodatkowo powinna umożliwiać podgląd danych historycznych oraz automatyczną archiwizację danych. Najistotniejsze kwestie przy wyborze oprogramowania monitorującego to:

- dostępność agentów monitorujących na różne platformy
- możliwość samodzielnego definiowania monitorowanych parametrów
- łatwość skalowania centralnej aplikacji wraz ze wzrostem ilości monitorowanych parametrów
- możliwość definiowania wielu warunków detekcji problemu (alarm zależy od wartości kilku parametrów)

4.4 Wyznaczanie wartości progowych

Dobranie odpowiednich wartości progowych jest bardzo trudne i wymaga dobrej znajomości naszej infrastruktury oraz przede wszystkim aplikacji. Zbyt niskie wartości skutkują fałszywymi alarmami, zbyt wysokie brakiem alarmu. Należy przy tym pamiętać, że środowisko w którym działa aplikacja ulega ciągłym zmianom. Zmienia się aktualna liczba użytkowników, wersje oprogramowania czy ilość wolnego miejsca na dyskach twardych. Wszystko to sprawia, że stosowanie sztywnych wartości progowych nie jest najlepszą metodą. Często prowadzi do wielu fałszywych alarmów lub niewykrycia problemu.

Jedną z metod dającą pozytywne rezultaty jest wykrywanie anomalii w wartościach parametrów. Jeżeli w historii danego elementu mamy informację o tym iż zazwyczaj we wtorki o godzinie 15 mieliśmy 1000 użytkowników to aktualna wartość 10000 powinna wzbudzić nasze zainteresowanie.

Podejście to wymaga dynamicznego sterowania wartościami progowymi na podstawie danych historycznych dzięki metodom statystycznym. Po ustaleniu szerokości okna czasowego w którym chcemy operować (np. ten sam dzień w zeszłym tygodniu lub ostatnia godzina) możliwe jest wyliczenie histogramu. Pozwala nam on poznać rozkład zebranych próbek. Znajomość rozkładu

jest konieczna ponieważ metody skuteczne przy rozkładzie normalnym (Gaussa) zazwyczaj nie działają poprawnie w innych przypadkach. Znając rozkład próbek możemy łatwo znaleźć wszystkie dla których szansa ich wystąpienia jest tak niska, iż należy je uznać za anomalie. W przypadku rozkładu normalnego będą to wszystkie próbki na brzegach “dzwonu”. Dzięki temu podejściu to czy system zgłosi problem zależy od tego jak dany parametr zachowywał się w przeszłości.

Inna i zarazem prostsza metoda to wyliczenie pierwszej pochodnej dla ostatnich n wartości pomiarowych. Im większa pochodna tym bardziej nagły był skok wartości mierzonej. Tutaj też mamy co prawda wartości progowe jednak metoda nie jest czuła na powolną zmianę wartości mierzonej. Dzieje się tak ponieważ wartości progowe dotyczą tempa zmian, a nie wartości parametru mierzonego. Typowe systemy charakteryzują się stabilnym ale powolnym wzrostem wartości (np. liczby użytkowników) dodatkowo wzrost wartości jest czymś porządanym i normalnym. Czymś normalnym są również wahania dobowe. W zależności od tego co oferuje nasza aplikacja możemy mieć znacząco różną liczbę użytkowników o różnych porach dnia. Pierwsza pochodna pozwala nam na wychwycenie anomalii niezależnie od wartości mierzonej, gdyż istotne jest tutaj tempo zmian.

4.5 Pytania kontrolne do rozdziału czwartego

Pyt 1. Podaj nazwę protokołu przyjętego jako ogólny standard udostępniania informacji o parametrach pracy sprzętu.
Pyt 2. Wymień 3 źródła informacji o parametrach pracy serwerów.

5. Podstawy administracji systemem

W tym rozdziale przedstawiona zostanie ogólna filozofia systemów z rodziny Unix oraz podstawowe narzędzia systemu Linux. Poznamy także typowe zadania z jakimi zmagają się administratorzy systemów i sieci komputerowych. Zapoznanie się z tymi zagadnieniami pozwoli lepiej uzmysłowić sobie jak ważną rolę w procesie zapewnienia integralności i bezpieczeństwa danych odgrywa człowiek. Wiedza tu zawarta pozwoli na rozpoczęcie przygody z systemami z rodziny Unix. Nie będziemy tu jednak przedstawiać wszystkich serwerowych systemów operacyjnych.

5.1 Filozofia systemów z rodziny UNIX

System UNIX i jego pochodne są najczęściej spotykanymi systemami operacyjnymi na sprzęcie serwerowym. Wszystkie te systemy pomimo często całkiem odrębnego rodowodu łączy wspólna filozofia działania. Każdy z nich jest połączeniem wielu małych programów stworzonych w jednym konkretnym celu. Każdy z tych programów spełnia najczęściej tylko jedną funkcję na przykład pobranie pliku z konkretnego adresu URL lub wyświetlenie zawartości pliku tekstowego (bez możliwości jego edytowania). Dzięki łączeniu wielu programów ze sobą możliwe jest wykonywanie bardziej skomplikowanych czynności. Podejście to sprawia, że programy są szybkie, optymalizowane do konkretnych zadań i zawierają zazwyczaj mniej błędów (ponieważ są prostsze i mniejsze). Do łączenia wyjść jednych programów z wejściami drugich używane są tak zwane potoki, oznaczane w linii poleceń znakiem ‘|’.

Dodatkowo każdy zasób w systemie jest reprezentowany jako plik. Nieważne czy jest to karta dźwiękowa, czy też dysk twardy. Dlatego możliwe jest np. odtworzenie zwartości dysku twardego na naszej karcie dźwiękowej za pomocą komendy do wyświetlania zawartości pliku. Jest to przykład dość abstrakcyjny którego wynikiem będzie jedynie dziwny szum ale dobrze obrazuje swobodę jaką zapewniają te systemy użytkownikowi. Wyobraźmy sobie teraz, że nasz szef kazał nam sprawdzić jak dużo maili przyszło do naszej firmy dnia 25 maja 2014. Nie musimy w tym celu sprawdzać skrzynki każdego pracownika czy szukać informacji ręcznie w logu systemu poczty. Wystarczy nam do tego jedna linijka:

```
>#  grep 25.05.2014 /var/log/mail* | grep -e “to=<.*@naszafirma.pl” | wc -l
```

Wyszuka ona najpierw wszystkie linie zawierające tekst “25.05.2014” w logach systemu poczty. Następnie w znalezionych liniach wyszuka te które mówią o liście kierowanym do naszej firmy. Na koniec podliczy wszystkie linie spełniające oba te kryteria i wypisze wynik na ekran. Aby wykonać to zadanie użyliśmy tylko 2 instrukcji, gdybyśmy chcieli wysłać wynik szefowi wystarczy dodać:

```
 | mail -s ‘Ilość listów dnia 25.05.2014’ szef@naszafirma.pl
```

Dodanie kolejnego programu spowoduje, że wynik nie zostanie wypisany na ekran tylko przekazany kolejnemu programowi (w tym przypadku wysłany poprzez email). Moglibyśmy równie dobrze

zapisać go do pliku lub wydrukować zmieniając jedynie komendę użytą na końcu. Wszystko to sprawia, że z pomocą drobnych narzędzi możemy szybko rozwiązywać nawet z pozoru duże i skomplikowane problemy.

5.2 Podstawowe polecenia systemu Linux

Jako, że ta publikacja skupia się głównie na zagadnieniach szeroko pojętego bezpieczeństwa danych, ograniczymy się do komend najbardziej popularnego serwerowego systemu operacyjnego jakim jest system Linux. Niemniej większość z przedstawionych komend jest dostępna także na innych systemach z rodziny Unix lub ma tam swój odpowiednik. Dla poprawy czytelności polecenia zostały zebrane w tabelę i pogrupowane pod względem funkcji jakie spełniają w systemie. Każde z poleceń posiada szereg flag i parametrów modyfikujących jego zachowanie. Można je poznać z pomocą polecenia **man** wyświetlającego instrukcję do wybranego programu (np. **man grep**).

Polecenie	Opis
Operacje na zawartości pliku	
cat	wypisanie zawartości pliku, domyślnie na ekran
grep	wyszukanie wystąpień wzorca w strumieniu wejściowym
sed	edycja strumienia, na przykład zamiana jednego słowa na inne w całym pliku
Vim, nano, emacs	popularne edytory plików
Operacje na plikach i katalogach	
ls	wypisanie zawartości katalogu
cd	zmiana bieżącego katalogu
cp	skopiowanie pliku
mv	przeniesienie pliku lub zmiana jego nazwy
rm	skasowanie pliku
touch	utworzenie pustego pliku
mkdir	utworzenie katalogu
du	podliczenie rozmiaru plików
find	wyszukiwanie plików i katalogów
chmod	zmiana uprawnień do pliku

Polecenie	Opis
chown	zmiana właściciela pliku
Informacje o systemie operacyjnym	
ps	lista uruchomionych procesów
top	interaktywny program informujący o obciążeniu systemu
ip	wyświetlanie i edycja ustawień sieciowych
df	informacja o zajętości dysków twardych
env	wypisuje zmienne środowiskowe
lspci	wypisuje urządzenia na szynie PCI

Tabela 2: Zbiór najpopularniejszych poleceń systemowych.

Tabela ta jest tylko zbiorem podstawowych poleceń których znajomość jest absolutnie niezbędna. Kompletny system zawiera tysiące poleceń, a poznanie ich wymaga czasu i praktyki.

5.3 Skrypty powłoki

Na początku rozdziału poznaliśmy potoki dzięki którym możemy łączyć wyjścia i wejścia różnych programów. Istnieje jednak jeszcze jeden sposób na wykorzystanie potencjału systemu. Są to skrypty powłoki. Powłoka jest to program odpowiadający za wykonywanie w systemie zadań zleconych przez użytkownika. Tak więc każde wpisywane przez użytkownika polecenie jest interpretowane i wykonywane przez powłokę. Skrypt powłoki jest plikiem tekstowym zawierającym polecenia które zostaną wykonane po jego uruchomieniu. Poza zwykłym wykonaniem poleceń skrypty powłoki mogą zawierać dodatkowo znane z większości języków programowania instrukcje warunkowe (if/else, switch) oraz pętle (for, while). Posiadają także możliwość definiowania zmiennych. Wszystkie te elementy sprawiają, że możliwe jest łatwe tworzenie programów dopasowanych do naszych konkretnych wymagań. Gotowy skrypt powłoki staje się kolejnym programem dostępnym w systemie i tak jak pozostałe może być łączony z innymi poprzez potoki.

Skrypty pozwalają na łatwą automatyzację codziennych zadań i tym samym zmniejszenie nakładu pracy wymaganej do utrzymania systemu. Dodatkowo eliminują błędy pojawiające się przy często powtarzających się czynnościach. Nasz mózg ma tendencję do usypiania naszej uwagi w takich przypadkach, co może prowadzić do pomyłek. Raz napisany skrypt zachowuje się tak samo niezależnie od liczby uruchomień. Ma to pośredni wpływ na bezpieczeństwo danych przetwarzanych w naszym systemie. Ponieważ mniejsza ilość błędów to bardziej bezpieczny system. Napisanie dobrego skryptu wymaga jednak dokładnej analizy problemu który ma on rozwiązać. Konieczne jest sprawdzenie wszystkich możliwych przypadków użycia i formatów danych

wejściowych. Bez tego kroku skrypt może być niepełny i w pewnych przypadkach może sam stać się źródłem błędów w systemie, a nawet prowadzić do utraty danych (na przykład błąd w skrypcie wykonującym kopie zapasowe).

Aby plik tekstowy został rozpoznany jako skrypt musi być oznaczony jako wykonywalny oraz musi rozpoczynać się specjalną linią:

```
#!<ścieżka do programu powłoki>
```

Najczęściej linia ta będzie miała postać:

```
#!/bin/bash
```

Powłoka bash jest obecnie najczęściej spotykana i wiele innych stara się zapewnić jak największą z nią zgodność. Dlatego też jest to bez wątpienia najlepszy wybór dla osób chcących rozpocząć naukę administracji systemami. Poniżej możemy zobaczyć przykładowy skrypt wykonujący kopię zapasową wybranego folderu oraz przechowujący wykonaną kopię przez 7 dni.

```
#!/bin/bash

numer_dnia_tygodnia=`date '+%u'`
katalog_docelowy='/backup/'
sciezka_do_programu_pakujacego='/usr/bin/7z'

katalogi_do_kopiowania=(\
"</sciezka/pierwsza>" \
"</sciezka/druga>"\
)

baza_nazwy="kopia_z_dnia_numer_${numer_dnia_tygodnia}.7z"

for dir in ${!katalogi_do_kopiowania[*]}
    do
        echo "Dodaję: ${katalogi_do_kopiowania[$dir]}"
        $sciezka_do_programu_pakujacego a "${katalog_docelowy}${baza_nazwy}" "${katalogi_do_kopiowania[$dir]}"
    done
```

Skrypt taki raz na zawsze rozwiązuje problem wykonywania cyklicznej kopii zapasowej wybranych folderów. Dzięki użyciu zmiennych do konfiguracji może być szybko przenoszony pomiędzy różnymi serwerami. Widzimy tutaj także przykład użycia pętli for w skrypcie. Ten prosty skrypt pokazuje jak łatwe jest łączenie kilku programów realizujących pojedyncze funkcje w większą całość. Skrypt taki można łatwo dostosować do naszych potrzeb lub rozbudować o kolejne funkcjonalności, jak na przykład kopiowanie wykonanego archiwum na inny serwer.

5.4 Typowe zadania administratora

Obecnie szacuje się, że każdego dnia na świecie przesyła się 2,64 eksabajta (1 eksabajt = 10^{18} bajtów) danych. W przeliczeniu daje to około 2 gigabajty (1 gigabajt = 10^{9} bajtów) danych na każdego mieszkańca Ziemi dziennie. Taka ilość informacji wymaga ogromnych mocy obliczeniowych. Ilość urządzeń koniecznych do obsłużenia tego nieprzerwanego strumienia danych sprawia, że niemożliwym jest ręczne kontrolowanie każdego z tych urządzeń. Dlatego też praca administratorów polega głównie na możliwie jak największej automatyzacji każdej czynności związanej z utrzymaniem infrastruktury. Poza tym również na doskonaleniu systemów monitorujących pracę urządzeń. Tak aby na podstawie nowych danych pomiarowych móc automatyzować kolejne czynności. Przyjęło się nawet ogólne powiedzenie, że dobry administrator jest człowiekiem któremu płaci się za to, że z pozoru nic nie robi. Ponieważ wszystkie czynności zostały zautomatyzowane, a gdy coś robi oznacza to, że mamy awarię.

Jednak aby rozpocząć wdrażanie automatyzacji konieczne jest poznanie typowych zadań. Zaliczają się do nich:

- wykrywanie i usuwanie awarii sprzętowych
- instalacja i wstępna konfiguracja nowego sprzętu
- kontrola poprawności wykonania kopii zapasowych
- reagowanie na problemy z wydajnością przetwarzania danych
- reagowanie na problemy zgłaszane przez użytkowników
- aktualizacje oprogramowania
- analiza logów aplikacji w celu wykrycia problemów, anomalii lub prób nieuprawnionego dostępu

Nie są to wszystkie zadania związane z tą funkcją jednak już teraz wyraźnie widać, że bez automatyzacji tych zadań liczba osób które trzeba by było zatrudnić do obsługi infrastruktury jest znacząca.

Gdybyśmy chcieli modelować większość z tych zadań z pomocą grafu decyzyjnego okazałoby się, że sprowadzają się one do wykonania operacji na plikach tekstowych w zależności od zawartości innych plików. Wynika to bezpośrednio z filozofii Unixa którą poznaliśmy na początku rozdziału. Wszystko jest plikiem dlatego zmiana ustawień programu sprowadza się do edycji jego pliku konfiguracyjnego. Podobnie jest ze zbieraniem informacji o systemie. Jądro systemu udostępnia je w postaci plików tekstowych umieszczonych w specjalnym katalogu (w systemach Linux są to katalogi */proc* oraz */sys*). Pozwala to na szybkie poznanie aktualnego stanu systemu przy użyciu narzędzi do operacji na plikach. Znacząco ułatwia to monitorowanie systemu i automatyzację zadań.

Warto już na samym początku naszej przygody z administracją systemami zwracać uwagę na automatyzację zadań. Im wcześniej wyrobimy w sobie taki nawyk tym nasza praca będzie łatwiejsza. Każda czynność którą wykonujemy powinna rodzić pewne pytania: Czy i jak mogę ją zautomatyzować? Jak mogę to wykonać na kilkuset serwerach na raz? Czy mogę monitorować poprawność wykonania? Nawet jeśli nie potrzebujemy aktualnie tak dużej skali operacji warto

zastanowić się co będzie trzeba zrobić aby utrzymać naszą infrastrukturę gdy urośnie. Każda działalność w internecie ma potencjalnie globalny zasięg. Nigdy nie wiadomo czy i kiedy nasz produkt zyska popularność. Ta krótka chwila zastanowienia może ujawnić nam wady naszego podejścia do danego problemu i pozwolić na rozwiązanie problemów zanim zaczną być przeszkodą w rozwoju.

5.5 Podsumowanie

Systemy z rodziny Unix pomimo swej prostej konstrukcji są bardzo potężnym narzędziem, które może być łatwo dostosowane do każdego zadania. Przedstawione w tym rozdziale podstawowe polecenia oraz zasady pracy z systemem pozwolą na rozpoczęcie nauki obsługi systemu. Mnogość wersji i dystrybucji systemów sprawia, że niemożliwe jest napisanie uniwersalnego podręcznika. Jednakże większość dystrybucji posiada rozbudowaną dokumentację która wprowadzi nas w elementy specyficzne dla danej wersji.

Należy również pamiętać, że w przypadku systemów operacyjnych najlepszą metodą nauki jest ciągła praktyka i samodzielne rozwiązywanie napotkanych problemów. Nawet gdy korzystamy z podpowiedzi zamieszczonych na różnych forach dyskusyjnych musimy zastanowić się dlaczego ta podpowiedź zadziałała lub też nie w naszym konkretnym przypadku.

5.6 Pytania kontrolne do rozdziału piątego

Pyt 1. Czym są potoki w systemach z rodziny Unix?
Pyt 2. Podaj do czego służą polecenia: cat, ls, cd i rm.
Pyt 3. Czym są skrypty powłoki?

6. Wspólne sekrety i tajne klucze, czyli wstęp do kryptografii

Rozdział ten poświęcony będzie najbardziej chyba istotnemu aspektowi bezpieczeństwa danych. Zagadnienia z dziedziny kryptografii są kluczem do zrozumienia tego w jaki sposób zapewniamy poufność oraz integralność transmisji. Dlatego też zrozumienie zasad ich działania pozwoli uzmysłowić sobie potencjalne zagrożenia z nimi związane. Nie będziemy tu jednak przeprowadzać dokładnej analizy działania metod kryptograficznych. Wykraczało by to znacznie poza zakres tego przedmiotu. Skupimy się jedynie na krótkiej ich charakterystyce i typowych zastosowaniach.

Na początek warto jednak wspomnieć czym jest kryptografia. Jest to dziedzina nauki zajmująca się metodami bezpiecznego przekazywania komunikatów (danych). Zajmuje się zarówno kwestiami poufności jak i integralności komunikatu. Łączy ona zagadnienia z dziedziny matematyki, informatyki oraz teorii informacji. Początkowo stosowana głównie w wojskowości oraz dyplomacji, obecnie jednak bardzo szeroko wykorzystywana.

6.1 Podział metod kryptograficznych

Obecnie dostępnych jest wiele funkcji kryptograficznych, każda z nich charakteryzuję się inną złożonością obliczeniową oraz możliwymi zastosowaniami. Poniżej przedstawione zostały trzy jej rodzaje najszerzej stosowane w informatyce.

6.1.1 Kryptografia symetryczna

Termin “kryptografia symetryczna” odnosi się do wszystkich metod w których do zaszyfrowania i odszyfrowania komunikatu używa się tego samego klucza. Wyróżniamy tutaj dwa podstawowe typy szyfrów: blokowy oraz strumieniowy.

W przypadku szyfrów blokowych dane dzielimy na bloki o określonej długości które następnie przekształcamy w szyfrogram. Szyfry strumieniowe operują na strumieniu danych. W ich przypadku kolejne elementy szyfrogramu często zależą także od poprzednich wartości komunikatu, a nie wyłącznie od klucza szyfrującego. Typowymi przedstawicielami tych szyfrów w informatyce są:

- AES oraz wycofywany już DES (szyfry blokowe)
- RC4 (szyfr strumieniowy)

6.1.2 Kryptografia asymetryczna

Głównym problemem kryptografii symetrycznej jest konieczność bezpiecznego przekazania klucza drugiej stronie. Dlatego też konieczne było opracowanie metod wolnych od tego problemu. W kryptografii asymetrycznej stosowane są pary kluczy, klucz publiczny oraz klucz prywatny. Są one

konstruowane w taki sposób aby obliczenie klucza prywatnego na podstawie publicznego było praktycznie niewykonalne i wymagało znacznej mocy obliczeniowej. Klucz publiczny pozwala na zaszyfrowanie komunikatu, który można odczytać jedynie przy użyciu klucza prywatnego. Pozwala to na przesłanie klucza publicznego otwartym kanałem komunikacji bez obaw o poufność transmisji. Gdy wszystkie strony transmisji wymienią się nawzajem kluczami publicznymi, możliwe jest nawiązanie bezpiecznego kanału komunikacji. Wiadomość będzie zaszyfrowana oddzielnie dla każdego odbiorcy przy użyciu jego klucza publicznego.

Poza szyfrowaniem metody te mogą zostać użyte także do potwierdzenia integralności oraz nadawcy komunikatu. Nadawca może przy pomocy klucza prywatnego dołączyć do jawnej wiadomości podpis elektroniczny, wygenerowany na podstawie wiadomości oraz klucza prywatnego. Podpis ten można potwierdzić przy pomocy klucza publicznego, co daje pewność integralności oraz źródła komunikatu. Każda zmiana przypadkowa (błąd transmisji) lub celowa (zmiana treści) spowoduje, że podpis przestanie się weryfikować.

Metody tego typu są złożone obliczeniowo dlatego zazwyczaj są używane jedynie do ustalenia bezpiecznego kanału komunikacji, którym następnie przesyłany jest klucz do szyfrowania symetrycznego. Typowymi przedstawicielami są metody: Diffiego-Hellmana i RSA.

6.1.3 Jednokierunkowe funkcje skrótu

Funkcje te nie są funkcjami szyfrującymi ponieważ komunikat jest tracony bezpowrotnie (stąd jednokierunkowość). Wynikowy ciąg znaków jest zazwyczaj stałej długości, niezależnie od długości danych wejściowych. Nie ma tutaj kluczy szyfrujących, a główną zaletą jest to, że dla tych samych danych wejściowy otrzymamy zawsze ten sam skrót.

Pewną podgrupą funkcji jednokierunkowych są kody uwierzytelniające wiadomości (MAC - message authentication code). Główna różnica to konieczność posiadania tajnego klucza w celu sprawdzenia integralności komunikatu. W praktyce implementowane jest to zazwyczaj poprzez doklejenie tajnego klucza na końcu jawnego komunikatu, a następnie wyliczenie skrótu z tak przygotowanej wiadomości. Tajny klucz jest następnie usuwany przed wysłaniem komunikatu. Dzięki temu potwierdzenie integralności może być dokonane jedynie przez osobę znającą tajny klucz.

Funkcje jednokierunkowe są bardzo proste obliczeniowo dlatego mogą być powszechnie stosowane w wielu aplikacjach bez większych obaw o wydajność sytemu. Typowi przedstawiciele to: MD5, SHA i jego warianty.

6.2 Przykłady zastosowań kryptografii w informatyce

Każda z wymienionych wyżej metod może mieć wiele różnych zastosowań. Zazwyczaj w zależności od potrzeb do jednego przypadku możliwe jest użycie kilku metod. Poniżej przedstawiono najbardziej typowe zastosowania kryptografii w informatyce:

6.2.1 Poświadczanie tożsamości serwera

Jest to chyba najczęstsze zastosowanie kryptografii asymetrycznej. Opiera się na certyfikatach klucza publicznego, ich dokładniejszy opisz znajduje się w podrozdziale 6.3.

Za każdym razem gdy łączymy się ze stroną WWW za pomocą bezpiecznego protokołu HTTPS, serwer wysyła nam podpisany przez zaufanego wystawcę certyfikat poświadczający jego tożsamość. Następnie nasz komputer sprawdza jego poprawność przy użyciu klucza publicznego zaufanego dostawcy certyfikatów. Cały proces wymaga zaufania trzeciej stronie (wystawcy certyfikatu) w celu poprawnej weryfikacji serwera.

6.2.2 Zapewnienie poufności transmisji

W codziennym użyciu zapewnienie poufności transmisji sprowadza się do zastosowania wspomnianego wcześniej protokołu HTTPS. Poprawna weryfikacja serwera nie jest warunkiem wymaganym dla zapewnienia poufności transmisji, jednak w tej fazie ustanawiany jest bezpieczny kanał komunikacji. Jest on następnie użyty do przesłania klucza symetrycznego. Dalsza transmisja jest już szyfrowana z użyciem metod symetrycznych, ponieważ jak pamiętamy, metody symetryczne wymagają mniejszej mocy obliczeniowej. Dzięki temu serwery są w stanie obsłużyć większą liczbę użytkowników.

Protokołami używanymi do przez HTTPS do zapewnienia bezpiecznej komunikacji są TLS (Transport Layer Security) oraz jego poprzednik SSL (Secure Sockets Layer).

6.2.3 Szyfrowanie poczty email

Szyfrowanie poczty email to tak naprawdę podgrupa problemów dotyczących szyfrowania plików w ogólności. Dodatkowe szyfrowanie poczty, pomimo zapewnienia poufności transmisji, jest konieczne ponieważ wiadomości są składowane na serwerach poczty. Bez dodatkowego szyfrowania mogłyby zostać odczytane przez zarządzających serwerem poczty lub potencjalnych atakujących którzy uzyskają do niego dostęp. Umożliwia ono dodatkowo podpisywanie wiadomości, dzięki czemu odbiorca może potwierdzić tożsamość nadawcy oraz integralność wiadomości.

Najczęściej stosowanym do szyfrowania poczty oprogramowaniem jest OpenPGP. Jego działanie opiera się zarówno na kryptografii asymetrycznej jak i symetrycznej. Do poprawnego działania wymagana jest wcześniejsza wzajemna wymiana kluczy publicznych przez wszystkie strony komunikacji. Wtyczkę do obsługi OpenPGP można znaleźć w większości popularnych programów pocztowych.

6.2.4 Poświadczanie tożsamości użytkownika

Poza poświadczeniem tożsamości serwera podobny mechanizm może zostać wykorzystany przez serwer do uwierzytelnienia użytkownika. Mechanizm ten opiera się na kryptografii asymetrycznej oraz może przyjąć dwa warianty.

Pierwszym jest sytuacja w której serwer wystawia klientowi (użytkownikowi) certyfikat podpisany swoim kluczem prywatnym. Następnie użytkownik w celu potwierdzenia tożsamości wysyła serwerowi plik certyfikatu. Jest to przypadek porównywalny do przesyłania zwykłego hasła. Dzieje się tak ponieważ przechwycenie transmitowanego certyfikatu przez atakującego jest równoznaczne z uzyskaniem przez niego dostępu do serwera na prawach tego użytkownika.

Wolnym od tego problemu wariantem jest sytuacja w której użytkownik generuje parę kluczy (prywatny oraz publiczny), po czym wysyła na serwer swój klucz publiczny. Następnie w celu potwierdzenia tożsamości serwer wysyła komunikat zaszyfrowany kluczem publicznym użytkownika. Jedynie osoba posiadająca klucz prywatny jest w stanie poprawnie odczytać komunikat, a następnie użyć go do uwierzytelnienia. W tym przypadku nawet przechwycenie komunikatu przez atakującego nie jest równoznaczne z uzyskaniem dostępu, ponieważ nie posiada on klucza prywatnego.

Najczęstszym zastosowaniem dla poświadczenia tożsamości użytkownika jest protokół SSH, służący do połączeń terminalowych ze zdalnymi serwerami. Umożliwia on uwierzytelnianie użytkowników za pomocą kluczy asymetrycznych.

6.2.5 Elektroniczny podpis dokumentu

Podpis elektroniczny jest kolejnym możliwym zastosowaniem kluczy asymetrycznych. Jest on bardzo zbliżony do procesu poświadczania tożsamości serwera za pomocą certyfikatu. Podobnie jak poprzednio jego bezpieczeństwo opiera się na zaufaniu do strony trzeciej. Osoba chcąca się posługiwać podpisem elektronicznym występuje do certyfikowanej instytucji z wnioskiem o wydanie certyfikatu. Certyfikat jest zazwyczaj wydawany w formie karty procesorowej zawierającej klucz prywatny oraz odpowiadający mu certyfikat podpisany przez tę instytucję. Użytkownik może następnie użyć karty do podpisania dowolnego dokumentu (pliku). Odbiorca dokumentu może zweryfikować poprawność podpisu przy użyciu klucza publicznego należącego do instytucji wydającej certyfikat.

Zazwyczaj karty procesorowe używane do podpisu elektronicznego nie dają możliwości wyeksportowania klucza prywatnego. Dlatego teoretycznie tylko i wyłącznie posiadacz karty jest w stanie złożyć poprawny podpis. Podpis ten jest generowany bezpośrednio przez mikroprocesor znajdujący się na karcie. Proces ten ma zapobiegać wykradaniu klucza prywatnego w celu składania nieautoryzowanego podpisu.

6.2.6 Sprawdzanie integralności danych

Opisane w rozdziale 2 problemy z przekłamaniem danych nie dotyczą wyłącznie pamięci RAM. Zmiana możliwa jest także podczas transmisji lub przechowywania danych na dysku twardym (tzw. “bit rot”). Dlatego istotne jest zapewnienie możliwości sprawdzenia, czy dany plik nie zawiera przekłamań. Typowo realizowane jest to poprzez jednokierunkowe funkcje skrótu. W momencie zapisu każdego pliku wystarczy wyliczyć jego funkcję skrótu. Następnie w każdej chwili możemy powtórzyć ten proces aby sprawdzić poprawność pliku. Możliwe jest także dokonanie identycznego sprawdzenia po przesłaniu pliku przez sieć lub skopiowaniu na inny nośnik.

Niektóre systemy plików (np. ZFS) oferują taką kontrolę na poziomie bloków danych. Odbywa się ona w sposób ciągły przy każdym dostępie do pliku. Systemy takie zapewniają praktycznie zerowe prawdopodobieństwo niewykrytego przekłamania danych, a w przypadku pracy w konfiguracji RAID również automatyczną naprawę uszkodzonych danych.

6.2.7 Przechowywanie haseł użytkowników

Zdecydowana większość systemów informatycznych wymaga uwierzytelnienia w celu uzyskania dostępu do pewnych funkcji systemu. Sprawdzenie tożsamości zazwyczaj opiera się na zapytaniu o login i hasło użytkownika, a następnie porównaniu go z zapisanym w systemie. W przypadku wycieku danych na skutek błędu w aplikacji lub włamania, wszystkie zapisane hasła zostałyby ujawnione. Potencjalny atakujący mógłby użyć tych danych do zalogowania się na dowolne konto w systemie.

Rozwiązaniem tego problemu jest użycie funkcji jednokierunkowych do przechowywania haseł. Przed zapisaniem w systemie hasło jest przynajmniej jednokrotnie podawane na wejście funkcji jednokierunkowej. Następnie podczas logowania proces ten jest powtarzany i jego wynik porównywany z zapisanym w systemie. Dzięki temu procesowi nawet w przypadku wycieku danych, atakujący nie jest w stanie użyć ich w celu uzyskania dostępu do systemu. Należy tu jednak zaznaczyć, że obecnie dostępna moc obliczeniowa pozwala na szybkie wyliczenie funkcji skrótu (jednokierunkowej) dla wszystkich możliwych ciągów ośmioznakowych. Dlatego też zalecana długość hasła to minimum 12 znaków. Innym sposobem jest dodawanie pewnego ciągu znaków do hasła użytkownika co dodatkowo je wydłuża.

Przez znalezione błędy w implementacji nie zaleca się stosowania funkcji MD5 do przechowywania haseł. Nie zmniejsza to jednak jej użyteczności przy sprawdzaniu integralności danych.

6.2.8 Podpisywanie aplikacji

Podpisywanie kodu jest kolejnym zastosowanie kryptografii asymetrycznej. Główny cel to wyeliminowanie możliwości instalacji oprogramowania lub sterowników sprzętu pochodzących z niezaufanych źródeł. Podszywanie się pod znane programy to popularna technika wśród

atakujących. Prowadzi zazwyczaj do wycieku danych lub całkowitego przejęcia kontroli nad komputerem.

Zasada działania jest identyczna jak w przypadku podpisu cyfrowego. Podpisywanym dokumentem jest tutaj plik wykonywalny lub instalator programu. Podpis ten może być weryfikowany na etapie instalacji lub przy każdorazowym uruchomieniu programu.

6.2.9 Szyfrowanie plików oraz nośników danych

W celu ochrony danych przetrzymywanych na nośnikach wymiennych oraz dyskach twardych należy stosować szyfrowanie plików lub całych nośników danych. Jest to kolejne typowe zastosowanie metod symetrycznych. Dane na nośnikach w rzeczywistości podzielone są na bloki o stałym rozmiarze. Z tego powodu do ich zabezpieczania stosowane są zazwyczaj szyfry blokowe.

Klucz szyfrujący musi być cały czas w pamięci systemu, więc każde jego uruchomienie wymaga podania klucza. Przez to systemy stosujące pełne szyfrowanie danych narażone są na wyciek klucza szyfrującego (oraz danych) jedynie w przypadku uzyskania przez atakującego pełnego dostępu do już uruchomionego systemu. Znacznie ogranicza to ryzyko wycieku danych związane z kradzieżą nośników z kopią zapasową lub nawet całych serwerów z siedziby firmy.

6.3 Certyfikaty klucza publicznego

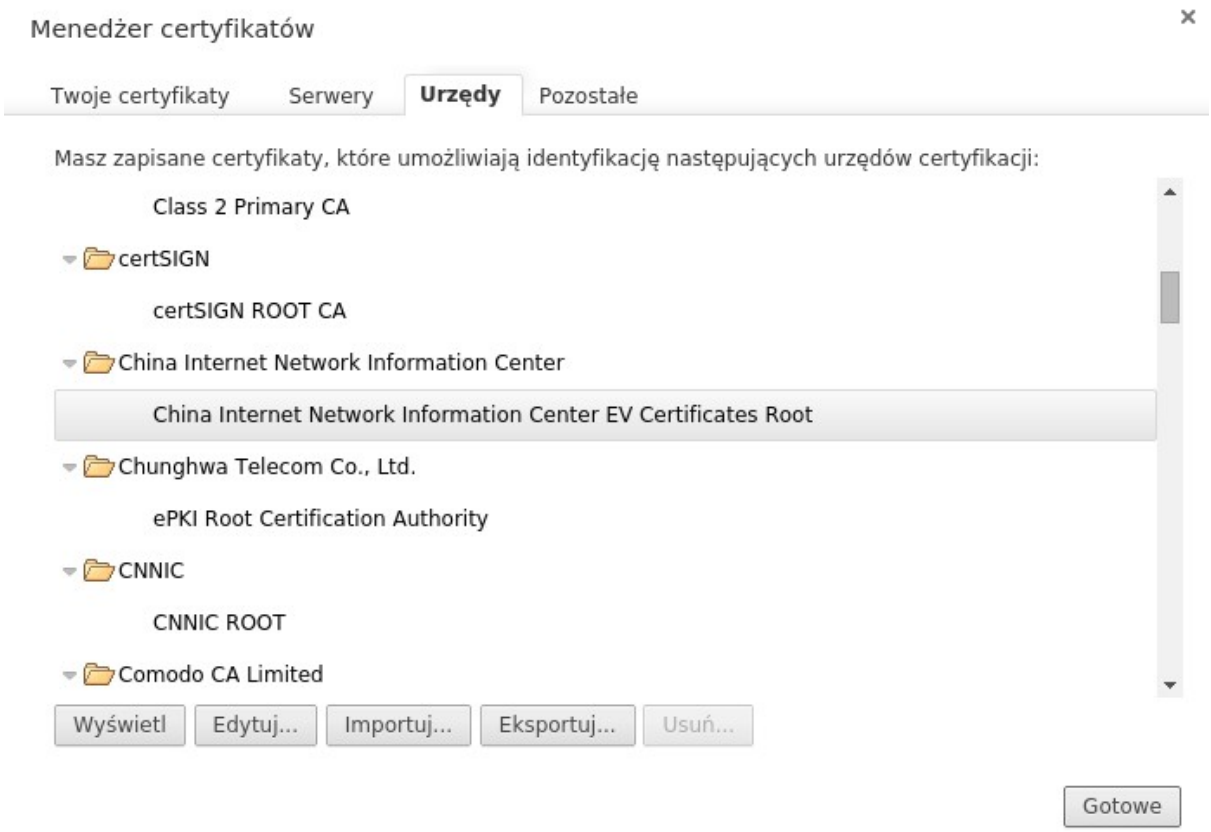
Wiele poznanych w poprzednim podrozdziale zastosowań kryptografii wykorzystuje mechanizm certyfikatów. Typowy certyfikat ma postać podpisanego cyfrowo pliku. Plik ten poza informacjami technicznymi zawiera, także informację o tym kto wydał dany certyfikat, od i do kiedy jest on ważny, możliwe przypadki użycia (poświadczenie tożsamości, podpisywanie dokumentów) oraz przede wszystkim czyją tożsamość potwierdza. Zawiera również odcisk klucza publicznego powiązanego z kluczem prywatnym.

Organ wydający certyfikat jest w tym przypadku stroną trzecią gwarantującą tożsamość podmiotu posługującego się certyfikatem. Organy zajmujące się wydawaniem certyfikatów są nazywane często centrami certyfikacji (z j ang. CA - certificate authority). Aby móc wydawać certyfikaty poprawnie rozpoznawane przez większość systemów, centrum certyfikacji musi umieścić swój klucz publiczny w jak największej liczbie systemów. Dlatego też wszystkie przeglądarki internetowe oraz systemy operacyjne dostarczane są razem z bazą kluczy publicznych wielu centrów certyfikacji.

Problem zaufania stronie trzeciej jest głównym problemem mechanizmu certyfikatów. Baza zawiera również klucze wielu zagranicznych agencji rządowych, które są w stanie wygenerować dowolny certyfikat poprawnie rozpoznawany przez większość urządzeń. Istnieje także problem bezpieczeństwa systemów w samym centrum certyfikacji. Atakujący który uzyskałby do niego dostęp byłby w stanie wygenerować poprawny certyfikat poświadczający tożsamość dowolnego serwera. Takie sytuacje zdarzały się już w przeszłości [2)] i nie ma gwarancji, że nie zdarzą się ponownie.

Wszystko powyższe powinno uświadomić nam, że symbol kłódki widoczny w naszej przeglądarce internetowej nie oznacza wcale iż połączenie zostało nawiązane z właściwym serwerem. Jest to jedynie informacja o tym, że posiadamy w naszej lokalnej bazie kluczy, taki klucz który potwierdza przedstawiony przez serwer certyfikat. To czy nasza lokalna baza kluczy nie została zmieniona, czy programista nie umieścił w niej dodatkowych kluczy lub też czy centrum certyfikacji jest nadal bezpieczne, pozostaje niewiadomą.

Zachęcam czytelnika do własnoręcznego sprawdzenia bazy kluczy używanej przez jego przeglądarkę internetową. Panel taki jest zazwyczaj dostępny w ustawieniach zaawansowanych, w sekcji bezpieczeństwa jako “Zarządzanie certyfikatami”. Przykładowe okno dla jednej z popularnych przeglądarek przedstawione zostało na poniższym zrzucie ekranu.



To czy certyfikat wysyłany przez dany serwer jest poprawny możemy w łatwy sposób sprawdzić z pomocą komendy **openssl** dostępnej w większości dystrybucji systemu Linux:

```
>$ openssl s_client -connect moja.pg.gda.pl:443
```

Komenda ta wyświetli nam pełną informację o certyfikacie serwera https://moja.pg.gda.pl oraz centrum certyfikacji, użytego do jego wygenerowania.

Komenda **openssl** może być również stosowana do generowania nowych lub podpisywania kolejnych certyfikatów. Certyfikaty podpisane własnoręcznie (tzw. self signed) będą weryfikowane poprawnie jedynie na tych urządzeniach na których ręcznie dodamy nasz klucz publiczny do

lokalnej bazy kluczy. Osoby zainteresowane dokładniejszym poznaniem mechanizmu certyfikatów zachęcam do zapoznania się z dokumentacją **openssl** [3)].

6.4 Losowość w informatyce

Problemem silnie związanym z kryptografią jest problem generowania liczb losowych. Aby wygenerować bezpieczny i tajny klucz prywatny konieczne jest posiadanie dobrego generatora liczb losowych. Powinien on generować losowy ciąg elementów, który posiada jednorodny rozkład prawdopodobieństwa oraz którego parametry statystyczne (wartość średnia oraz wariancja) są stałe w czasie. W innym przypadku teoretycznie możliwe byłoby poznanie kolejnych wartości generatora losowego, co mogło by prowadzić do złamania algorytmu szyfrującego. Mogło by się również okazać, że wartości generatora oscylują wokół pewnego punktu, co znacznie zwiększa prawdopodobieństwo poznania kolejnej wartości generatora. Problem ten nie dotyczy jedynie kryptografii ale także generowania unikalnych identyfikatorów sesji oraz wszystkich algorytmów wymagających wartości losowych.

Systemy informatyczne nie są w stanie programowo wygenerować prawdziwie losowych danych. Generatory losowości wbudowane w praktycznie każdy system operacyjny generują ciąg będący wynikiem skomplikowanych operacji matematycznych. Bazą tych operacji są różne zdarzenia systemowe takie jak: ruchy myszy, aktualna ilość pakietów wysyłanych i odbieranych przez kartę sieciową i tym podobne. Dzięki temu uzyskujemy ciąg bitów spełniający podstawowe założenia generatora losowego. Jednak generator losowy jak każdy inny program komputerowy może zawierać błędy które osłabią jego skuteczność.

Inną grupą generatorów są generatory sprzętowe. Do wytworzenia losowego ciągu znaków używają one danych pochodzących z obserwacji dowolnego stochastycznego procesu fizycznego. Może to być na przykład szum elektryczny lub termiczny. Generatory te są stosowane w systemach wymagających wydajnego źródła entropii (losowości) o wysokiej jakości.

6.5 Podsumowanie

Rozdział ten przybliżył czytelnikom podstawowe zagadnienia z dziedziny kryptografii oraz typowe ich zastosowania. Jak widać stworzenie bezpiecznego systemu nie jest możliwe bez zrozumienia podstawowych metod kryptograficznych oraz zagrożeń związanych z ich stosowaniem. Nie poruszano tu kwestii związanych z bezpiecznym przechowywaniem kluczy szyfrujących. Ich utrata mogła by prowadzić do utraty danych, gdyż sam użytkownik nie byłby w stanie ich odczytać.

Tematowi kryptografii w informatyce poświęcono już wiele publikacji specjalistycznych, mam jednak nadzieję, że rozdział ten przybliżył najistotniejsze kwestie praktyczne oraz będzie stanowił dobrą bazę do poszerzania wiedzy w tym temacie.

6.6 Pytanie kontrolne do rozdziału szóstego

Pyt 1. Podaj 3 przykładowe zastosowania metod kryptograficznych w informatyce.
Pyt 2. Podaj różnicę pomiędzy kryptografią symetryczną oraz asymetryczną.
Pyt 3. Jakie zagrożenia niesie z sobą stosowanie mechanizmu certyfikatów?
Pyt 4. Co to jest i do czego może być stosowana funkcja jednokierunkowa?

7. Fikcja literacka, czyli bezpieczne systemy operacyjne

Złożoność współczesnych systemów operacyjnych sprawia, że praktycznie niemożliwe jest wytworzenie w pełni bezpiecznego systemu. Ilość użytych bibliotek dodatkowych oraz sterowników dostarczonych przez innych programistów dodatkowo utrudnia to zadanie. Nie możemy, więc nigdy jednoznacznie stwierdzić, że system jest bezpieczny. Możemy jednak, poprzez stosowanie odpowiednich procedur i dobrych praktyk, znacząco zminimalizować ryzyko włamania oraz utraty danych.

7.1 Fizyczne bezpieczeństwo danych - zarządzanie ryzykiem

Poziom fizycznego bezpieczeństwa danych może być mierzony jako pewien współczynnik prawdopodobieństwa. Jest on możliwy do wyliczenia na podstawie średniego czasu pomiędzy awariami (MTBF). Określenie tego jak krytyczne są to dane oraz tym samym jakie ryzyko ich utraty jest akceptowalne, należy do kierownictwa firmy oraz jej administratorów.

Niemniej istotną kwestią jest, także dostępność danych. Nawet najlepiej zabezpieczone przed awarią dane zdadzą się na nic kiedy nie będziemy mieli do nich dostępu. Dostępność danych oraz usług jest przedstawiana zazwyczaj jako wartość procentowa gwarantowanej dostępności w danym okresie do czasu trwania tego okresu. Na przykład gwarantowana dostępność na poziomie 99,9% rocznie oznacza, że dane mogą być niedostępne przez maksymalnie 8 godzin i 45 minut w roku. Nie oznacza to jednak, że na skutek awarii dane nie mogą być niedostępne dłużej. Mówi to nam jedynie o tym, że zastosowane rozwiązania sprawiają iż prawdopodobieństwo wystąpienia takiej sytuacji jest znikome.

Należy przy tym pamiętać, że każde zmniejszenie ryzyka utraty danych oraz zagwarantowanie wyższej dostępności niesie z sobą zwiększenie kosztów infrastruktury. Pomocne może być tu również określenie teoretycznego kosztu odtworzenia tych danych oraz stwierdzenie czy jest to w ogóle możliwe. Pozwoli to na lepsze oszacowanie akceptowalnego ryzyka ich utraty.

7.2 Aktualizacje oprogramowania

Jedną z podstawowych czynności pozwalających na zmniejszenie ryzyka wycieku lub utraty danych na skutek włamania jest ciągła aktualizacja zainstalowanego oprogramowania. Współczesne systemy operacyjne udostępniają rozbudowane narzędzia które czynią ten proces praktycznie bezobsługowym. Proces ten jest dokładnie opisany w dokumentacji każdego systemu operacyjnego.

Warto jednak opracować i wdrożyć w firmie procedurę instalowania aktualizacji. Pozwoli to na uniknięcie problemów związanych ze zmianą zasad działania bibliotek lub narzędzi wykorzystywanych w naszych aplikacjach. Procedura taka powinna zawierać informację o tym kogo należy powiadomić przed planowaną zmianą, kiedy aktualizacja może zostać przeprowadzona, jakie kroki należy wykonać przed każdą aktualizacją (np. kopie zapasowe danych) oraz jak wycofać

zmianę. Pozwoli to na usystematyzowanie procesu oraz zmniejszenie ilości problemów związanych z aktualizacjami oprogramowania. Dodatkowym punktem procedury powinny być kroki postępowania w przypadku krytycznych poprawek związanych z bezpieczeństwem. W takim przypadku aktualizacja powinna zostać przeprowadzona jak najszybciej jednak z zachowaniem kroków wymaganych przed normalną aktualizacją (np. kopie zapasowe danych).

7.3 Podstawowe zabezpieczenie systemu operacyjnego

Ten podrozdział został poświęcony ogólnie przyjętym metodom podstawowego zabezpieczenia systemu operacyjnego. Jest to swego rodzaju zbiór dobrych praktyk. Skupimy się tutaj głównie na dystrybucjach systemu Linux jednak część metod można z powodzeniem zastosować w innych systemach.

7.3.1 Ograniczenie dostępu zdalnym użytkownikom

Jedną z czynności którą najlepiej wykonać przed podłączeniem serwera do sieci Internet jest ograniczenie dostępu zdalnego. Sprowadza się to do utworzenia nowej grupy w systemie i zmianie konfiguracji demona SSH (program używany do realizacji połączeń zdalnych). Zmiana polega na dodaniu do pliku konfiguracyjnego (zazwyczaj */etc/ssh/sshd_config*) linii:

```
AllowGroups <nazwa_grupy>
```

Dzięki temu możliwość połączenia zdalnego będą mieli tylko użytkownicy należący do tej grupy. Nie musimy więc analizować wszystkich kont w systemie i decydować o ich blokowaniu lub usunięciu (choć operacja ta jest również zalecana).

Jeśli dany serwer ma być zarządzany tylko z siedziby firmy to warto również ograniczyć dostęp do demona SSH z innych niż firmowe adresów IP.

7.3.2 Wyłączanie zbędnych usług

Kolejnym krokiem jest wyłączenie wszystkich zbędnych usług sieciowych. Część systemów operacyjnych posiada wiele preinstalowanych serwerów usług. Część z tych aplikacji może być domyślnie uruchomiona. Stwarza to potencjalne zagrożenie bezpieczeństwa ponieważ aplikacje te nie są zazwyczaj skonfigurowane i często używają domyślnych haseł lub udostępniają wrażliwe informacje. Aby się przekonać jakie serwery usług sieciowych działają w naszym systemie najlepiej jest wyświetlić listę wszystkich portów TCP oraz UDP otwartych w trybie nasłuchiwania. W tym celu wydajemy komendę:

```
># netstat -lpnut
```

Spowoduje to wypisanie wszystkich otwartych portów TCP oraz UDP wraz z nazwami aplikacji. Teraz wystarczy jedynie wyłączyć wszystkie zbędne aplikacje, informację o tym jak tego dokonać znajdziemy w dokumentacji naszego systemu operacyjnego.

7.3.3 Uwierzytelnianie w serwerze SSH z użyciem klucza publicznego

W przypadku gdy nie możemy ograniczyć dostępu do demona SSH tylko dla wybranych adresów IP, warto zablokować możliwość uwierzytelniania z użyciem hasła. Dzięki temu zablokujemy możliwość odgadnięcia hasła metodą prób i błędów (tzw. brute force).

Demon SSH oferuje możliwość uwierzytelniania za pomocą pary kluczy kryptograficznych. Na komputerze używanym do zarządzania serwerami generujemy parę kluczy, prywatny oraz powiązany z nim publiczny. Najprościej wykonać to za pomocą komendy:

```
>? ssh-keygen
```

Następnie klucz publiczny kopiujemy na serwer do pliku wskazanego w konfiguracji demona SSH (domyślnie jest to plik *.ssh/authorized_keys* w katalogu domowym użytkownika). Obecnie większość systemów oferuje narzędzie **ssh-copy-id** wspomagające ten proces. Po zweryfikowaniu, że logowanie na podstawie klucza działa poprawnie możemy zablokować dostęp z użyciem hasła.

Od tego momentu musimy chronić nasz klucz prywatny, ponieważ pozwala on na poprawne logowanie do serwera każdemu kto go posiada. Dodatkowo jego utrata spowoduje brak możliwości zalogowania się na serwer. Warto zawczasu opracować procedurę postępowania w takiej sytuacji.

7.3.4 Sprawdzenie praw dostępu do plików oraz katalogów

Kolejnym wartym sprawdzenia elementem są uprawnienia do plików oraz katalogów. Programiści, często z sobie tylko znanych powodów, ustawiają bardzo liberalne prawa dostępu do plików i katalogów swojej aplikacji. Może to prowadzić do wycieku wrażliwych informacji lub zmiany części kodu aplikacji przez atakującego. Dokładne sprawdzenie i ewentualna korekta uprawnień pozwoli na uniknięcie wielu problemów w przypadku ataku na system poprzez błąd w aplikacji.

7.4 Dzienniki zdarzeń

Podstawowym elementem pozwalającym na wykrycie i analizę problemu są dzienniki zdarzeń. Mają one zazwyczaj postać plików tekstowych, które domyślnie znajdują się w katalogu */var/log*. Większość aplikacji zapisuje w nich użyteczne dla administratorów informacje o wykonanej pracy oraz napotkanych problemach. Sprawdzenie dzienników zdarzeń (zwanych potocznie „logami”) powinno być pierwszą czynnością w przypadku napotkania problemów z systemem lub aplikacjami. Za zapisywanie tych informacji odpowiada zazwyczaj demon **syslogd**. Oferuje on możliwość centralnej kontroli nad zapisem zdarzeń generowanych przez działające w systemie aplikacje oraz sam system operacyjny.

Dzienniki te są również istotne ze względu na bezpieczeństwo systemu. Zawierają informacje takie jak próby nieudanego logowania, historię poprawnych logowań do systemu, czy też historię dostępu do usług sieciowych. Dodatkowo możliwe jest zapisanie każdej akcji wykonanej

w systemie. Do tego celu stosuje się program **auditd**. Po dostosowaniu jego konfiguracji do naszych potrzeb, zapisze on każdą interesującą nas operację wykonaną w systemie operacyjnym.

Po dostosowaniu systemu logowania do własnych potrzeb warto wdrożyć centralne składowanie logów. Sprowadza się to do przeznaczenia dodatkowego serwera odpowiedzialnego za zbieranie, przetwarzanie i archiwizację dzienników zdarzeń. Dzięki takiemu podejściu odbieramy potencjalnemu atakującemu możliwość zatarcia śladów swojej działalności. Możemy także zaimplementować centralny system analizy zdarzeń systemowych, uprości to znacząco ich automatyczne przetwarzanie oraz pozwoli na łatwe odnajdywanie korelacji pomiędzy zdarzeniami na różnych serwerach. Zwiększa to również stabilność oraz wydajność całego systemu. Ciągle przyrastające pliki dzienników mogłyby zapełnić cały dysk twardy serwera, prowadząc do problemów z stabilnością systemu operacyjnego. Natomiast ciągły jednostajny zapis na dyskach twardych zużywa również zasoby systemowe, co wpływa na ogólną wydajność systemu.

7.5 Kopie zapasowe

Nawet najlepsze zabezpieczenia można złamać i nawet najbardziej niezawodny system może się zepsuć. W takich przypadkach konieczne może się okazać odtworzenie danych z kopii zapasowej. Jednak najpierw trzeba ją posiadać. Warto poświęcić czas na dokładne opracowanie procedur wykonywania i składowania kopii zapasowych. Dodatkowo należy też opracować i co ważniejsze przetestować instrukcję odtwarzania danych na wypadek awarii. Regularne testy odtwarzania danych pozwalają sprawdzić czy procedury składowania są skuteczne i czy cały proces działa poprawnie. Musimy pamiętać, że przed wystąpieniem awarii możemy spokojnie i bez nerwów się do niej przygotować. Gdy jednak nastąpi moment awarii musimy być pewni, że jesteśmy w stanie szybko i sprawnie przywrócić system do normalnego stanu. Dlatego też poza danymi roboczymi warto składować również pliki konfiguracyjne, tak aby jak najszybciej przywrócić system po awarii.

Pamiętajmy o uwzględnieniu trwałości nośnika z kopią zapasową oraz maksymalnej ilości cykli kasowania i zapisu zalecanej przez producenta. Ważnym czynnikiem wpływającym na bezpieczeństwo danych jest także miejsce przechowywania kopii zapasowych. Kopie zapasowe przechowywane na tym samym serwerze na którym znajdują się oryginalne dane, lub nawet na tym samym dysku, nie spełnią swego podstawowego zadania w czasie awarii.

Częstotliwość wykonywania kopii zapasowych oraz czas ich przechowywania należy dobrać do rodzaju składowanych danych. W niektórych przypadkach wystarczy dostęp do kopii z ostatniego tygodnia, w innych natomiast może być konieczne odwołanie się do kopii sprzed kilku lat. Wpływ na czas przechowywania mogą mieć także uwarunkowania prawne. Obecnie dla dokumentacji medycznej okres taki wynosi 20 lat od końca roku kalendarzowego w którym dokonano ostatniego wpisu. W szczególnych przypadkach, gdy zgon pacjenta nastąpił wskutek uszkodzenia ciała lub zatrucia, okres ten wydłuża się do 30 lat. Dodatkowo dokumentację pacjenta należy traktować jako całość, a nie jako zbiór odrębnych dokumentów, dlatego poszczególne dokumenty (np. wyniki badań) nie mają odrębnej daty wygaśnięcia obowiązku ich

przechowywania. Powoduje to iż każdy nowy dokument dołączony do akt pacjenta wydłuża okres składowania całości.

7.6 Postępowanie po wykryciu włamania

Czasami nawet pomimo dołożenia wszelkich starań przy zabezpieczaniu systemu włamywaczom udaje się uzyskać do niego dostęp. Ten podrozdział przedstawi w skrócie kroki jakie należy wykonać po wykryciu takiego incydentu.

Pierwszym krokiem powinno być odcięcie serwera od sieci Internet. Może to być zrealizowane poprzez fizyczne wyjęcie przewodu sieciowego lub przeniesienie interfejsu sieciowego do wydzielonego segmentu sieci. Przeniesienie takie jest zazwyczaj realizowane poprzez rekonfigurację przełączników sieciowych. Krok ten zapobiega dalszemu wyciekowi danych oraz potencjalnemu atakowi na pozostałe serwery przy wykorzystaniu zainfekowanej maszyny.

Następnie wykonujemy procedurę zmiany hasła i ponownej generacji kluczy dla wszystkich użytkowników którzy posiadali dostęp do serwera. Krok ten ma na celu powstrzymanie eskalacji problemu w przypadku gdyby okazało się, że przyczyną włamania był wyciek danych dostępowych.

Kolejnym krokiem powinno być dokładne sprawdzenie zainfekowanego serwera oraz wszystkich dzienników zdarzeń mogących zawierać informacje na temat włamania. Konieczne jest bowiem znalezienie luki w zabezpieczeniach której użył atakujący oraz określenie od jakiego czasu system jest zainfekowany. Dokładne poznanie przyczyn włamania jest niezbędne do poprawnego usunięcia luki w zabezpieczeniach oraz sprawdzenie pozostałych serwerów pod kątem tej samej podatności.

Kiedy luka została już znaleziona i usunięta dokładnie kasujemy całą zawartość dysków zainfekowanego serwera i odtwarzamy jego stan z kopii zapasowej wykonanej przed włamaniem. Zmiany w systemie operacyjnym dokonane przez atakującego mogą być tak znaczące, że ilość pracy potrzebna do ich odwrócenia znacznie przewyższy czas potrzebny na odtworzenie z kopii zapasowej. Istnieje także szansa przegapienia zmiany i w konsekwencji pozostawienia złośliwego oprogramowania na serwerze.

7.7 Podsumowanie

W rozdziale tym poznaliśmy podstawowe zasady zabezpieczania systemów operacyjnych na wypadek włamania lub awarii. Poznaliśmy także podstawowe procedury związane z zabezpieczaniem danych. Ich opracowanie pozwoli określić wagę przetwarzanych danych oraz dostosować metody ochrony do postawionych wymagań. Niektóre przykłady takie jak zmiany konfiguracyjne nie wymagają dodatkowych nakładów finansowych. Inne natomiast jak wieloletnie składowanie danych mogą nieść z sobą konieczność poniesienia znacznych wydatków na infrastrukturę odpowiedzialną za składowanie kopii zapasowych.

7.8 Pytania kontrolne do rozdziału siódmego

Pyt 1. Co oznacza dostępność danych na poziomie 99.9% rocznie?
Pyt 2. Dlaczego należy testować procedury odtwarzania danych z kopii zapasowej?
Pyt 3. Co należy zrobić po wykryciu udanego ataku na jeden z naszych serwerów?

8. Hakerzy, wirusy i inne niebezpieczeństwa

Poznaliśmy już podstawowe problemy związane z zabezpieczaniem danych oraz zalecenia dotyczące konfiguracji systemów operacyjnych. Nadal nie wiemy jednak kto nam zagraża. W tym rozdziale omówiona zostanie klasyfikacja oraz motywy działania potencjalnych atakujących. Dodatkowo zawiera on także opis mechanizmów używanych przez nich do przeprowadzenia ataku.

8.1 Podział atakujących oraz motywy ich działań

Osoby lub grupy atakujące nasze systemy informatyczne dzielą się na kilka podstawowych typów. Przedstawiona uogólniona ich klasyfikacja wywodzi się z powszechnie przyjętej nomenklatury używanej w tekstach branżowych. W wielu tekstach czytelnik może spotkać się z bardziej szczegółowym podziałem, nie jest on jednak istotny z punktu widzenia tej publikacji.

8.1.1 Hakerzy

Hakerami nazywa się osoby poszukujące luk w oprogramowaniu. Luki te mogą zostać wykorzystane do przeprowadzenia ataku na system lub do poprawienia samego oprogramowania. Często osoby takie zajmują się przeprowadzaniem audytów informatycznych zlecanych przez przedsiębiorstwa w celu poprawy własnego bezpieczeństwa. Osoby takie posiadają zazwyczaj dużą wiedzę z zakresu wytwarzania oprogramowania, protokołów wymiany danych oraz budowy sprzętu komputerowego.

W zależności od konkretnego celu motywem ich działania może być chęć uzyskania rozgłosu, poprawa zabezpieczeń systemu, osiągnięcie zysków finansowych lub wsparcie akcji politycznych. Ich działania mogą prowadzić do uszkodzenia lub utraty danych i w wielu krajach mogą zostać sklasyfikowane jako przestępstwo.

8.1.2 Crackerzy

Crackerami nazywane są osoby zajmujące się łamaniem zabezpieczeń wbudowanych w oprogramowanie. Zabezpieczenia te są implementowane przez twórców oprogramowania w celu ochrony przed kopiowaniem. Dlatego też crackerzy działają zazwyczaj nielegalnie. Podobnie jak w przypadku hakerów posiadają oni zazwyczaj głęboką wiedzę z zakresu wytwarzania oprogramowania oraz zasad działania systemów informatycznych.

Zazwyczaj głównym motywem ich działania jest chęć osiągnięcia zysku lub zdobycia sławy. Ataków z ich strony powinna spodziewać się każda firma wytwarzająca zamknięte oprogramowanie lub związana z ochroną praw autorskich.

8.1.3 Script kiddie

Osoby należące do tej grupy najczęściej nie posiadają głębokiej wiedzy z zakresu wytwarzania oprogramowania lub protokołów sieciowych. Całą swoją działalność opierają na wykorzystywaniu informacji oraz oprogramowania udostępnianego przez hakerów oraz crackerów. Wyszukują i przejmują podatne komputery dzięki oprogramowaniu takiemu jak darmowy i łatwo dostępny Metasploit. Najczęściej nie znają oni szczegółów podatności które wykorzystują w swoich atakach.

Motywem ich działania może być chęć podbudowania własnej wartości lub zdobycia rozgłosu, czasem także chęć osiągnięcia korzyści finansowych.

8.1.4 Konkurencja

Konkurencyjne firmy nie pasują do typowej definicji hakera lub crackera, jednak z pewnością mogą być zainteresowane atakiem na nasze systemy informatyczne. Zazwyczaj firmy takie nie posiadają zasobów potrzebnych do przeprowadzenia skutecznego ataku. Dlatego najczęściej zlecają one atak grupom hakerów lub crackerów stanowiąc tym samym jedno ze źródeł ich dochodu.

Motywy działania są tu zazwyczaj czysto finansowe. Zdobycie planów projektowych nowego produktu przed jego premierą może prowadzić do zmniejszenia przewagi konkurencji oraz oszczędzić zasoby potrzebne do samodzielnego wytworzenia podobnego produktu. Szpiegostwo przemysłowe było znane na długo przed upowszechnieniem się komputerów. Jednak powszechna informatyzacja procesów projektowych dodatkowo uprościła ten proceder, ponieważ kopiowanie danych stało się dużo prostsze. Innym motywem może być chęć osłabienia wizerunku konkurencyjnej firmy, co w konsekwencji może prowadzić do osłabienia jej pozycji rynkowej.

8.2 Klasyfikacja złośliwego oprogramowania

Poznaliśmy już podstawowy podział potencjalnych atakujących, ten podrozdział przybliży nam narzędzia używane do ataków. Wszystkie opisane poniżej rodzaje programów, poza Rootkitami, są zazwyczaj używane do infekowania komputerów użytkowników końcowych. Taka kolejność ataku jest typowa z kilku powodów. Po pierwsze atak na komputery użytkowników jest zazwyczaj prostszy niż włamanie do serwera. Po wtóre przeprowadzenie skutecznego ataku na infrastrukturę serwerową jest zazwyczaj dużo prostsze kiedy działamy od wewnątrz. Po trzecie przejęcie kontroli nad dużą liczbą komputerów podłączonych do sieci jest konieczne do przeprowadzenia skutecznych ataków odmowy dostępu (DDOS – distributed denial of service). Ataki takie nie wykorzystują luk w oprogramowaniu, polegają one na nawiązaniu znacznej liczby połączeń sieciowych z atakowanym serwerem. Prowadzi to do wyczerpania jego zasobów oraz tym samym do przerwy w dostępie do usług. Dodatkowo przejęte komputery mogą zostać

wykorzystane do ukrycia prawdziwej tożsamości atakującego podczas przeprowadzenia ataku na inne systemy. Podstawowa klasyfikacja złośliwego oprogramowania przedstawia się następująco:

8.2.1 Wirus

Złośliwy kod lub program dołączający swój kod do innych programów w celu rozpowszechnienia się. Wirusy służą zazwyczaj jako narzędzie do instalacji bardziej zaawansowanych programów jakimi są trojany.

8.2.2 Trojan

Program pozwalający na przejęcie kontroli nad komputerem bez wiedzy właściciela. Ukrywa on swoją obecność w systemie oraz blokuje działanie programów antywirusowych. Zainfekowane komputery są łączone w sieci zwane botnetami. Sieci te są następnie wykorzystywane w atakach typu DDOS, łamaniu haseł, wysyłaniu niechcianych listów e-mail (SPAM) oraz wielu innych celach wybranych przez właściciela botnetu.

8.2.3 Spyware

Mianem tym określa się oprogramowanie zbierające informacje o użytkowniku bez jego wiedzy. Może również dokonywać zmian w ustawieniach innych programów (np. zmiana strony startowej w przeglądarce). Rozpowszechniane jest często jako dodatek do innych programów instalowanych przez użytkownika. Zazwyczaj nie jest ono także wykrywane przez programy antywirusowe.

8.2.4 Exploit

Program pozwalający na wykonanie kodu dostarczonego przez atakującego dzięki wykorzystaniu luki w oprogramowaniu lub systemie operacyjnym. Inaczej mówiąc, jest to program pozwalający na wykorzystanie błędów w oprogramowaniu zainstalowanym na komputerze użytkownika lub serwerze. Exploit sam w sobie nie infekuje zaatakowanego systemu, pozwala jednak na instalację innych programów takich jak trojany lub rootkity.

8.2.5 Rootkit

Programy te są funkcjonalnie zbliżone do trojanów. W literaturze przyjęło się jednak używanie tej nazwy do określania programów pozwalających atakującemu na przejęcie kontroli nad serwerami działającymi pod kontrolą systemów z rodziny Unix. Podobnie jak trojany pozwalają one

na przejęcie kontroli nad serwerem przez atakującego oraz ukrycie swojej obecności przed administratorami systemu.

8.3 Typowe podatności występujące w aplikacjach

Podatnościami w aplikacji nazywamy wszystkie błędy, zarówno projektowe jak i w implementacji, pozwalające atakującemu na wykonanie własnego kodu lub zmianę zasad działania programu. Wykorzystywane są do instalacji złośliwego oprogramowania, zmiany treści serwisów internetowych lub wykradania danych. Poziom skomplikowania współczesnego oprogramowania sprawia, że bardzo trudno jest uniknąć błędów powodujących pojawienie się podatności. Dodatkowo, jeżeli używamy zewnętrznych bibliotek programistycznych to nigdy nie możemy być pewni czy nie wprowadzają one dodatkowych podatności do naszej aplikacji.

To jak wiele podatności będzie zawierała nasze oprogramowanie zależy od doświadczenia zespołu, wybranych narzędzi programistycznych oraz sposobu zarządzania projektem. Każdy z tych czynników może przyczynić się do powstania podatności. Wśród najczęściej spotykanych podatności możemy wymienić:

Buffer overflows Polega na zapisaniu do zmiennej w programie wartości znacznie przekraczającej jej zdefiniowaną przez programistę długość. Powoduje to nadpisanie pamięci aplikacji przez wartości podane przez atakującego. Może to prowadzić do przejęcia kontroli nad wykonywaniem programu, a w konsekwencji uruchomienia dowolnego kodu przesłanego przez atakującego.

Dangling pointers Jest to przypadek w którym program używa wskaźnika do nieistniejącego już obiektu. Istnieje duże prawdopodobieństwo, że wskaźnik taki prowadzi do pamięci zajmowanej przez inny obiekt. Może to skutkować wyciekiem informacji zawartych w pamięci lub nadpisaniem jej zawartości.

SQL injection W przypadku użycia w naszej aplikacji bazy danych konieczne jest dokładne kontrolowanie poleceń wysyłanych do bazy danych. Jeżeli użytkownik ma możliwość kontrolowania fragmentu zapytania do bazy (np. poprzez pole w formularzu przesyłanym przez aplikację), to jest on w stanie umieścić w nim dodatkowe polecenia dla bazy danych. Może to prowadzić do wycieku informacji, zniszczenia danych lub nadania dodatkowych uprawnień nieupoważnionym użytkownikom. Dlatego też bardzo istotne jest dokładne sprawdzenie wszystkich danych przesyłanych przez użytkownika.

Directory traversal Podatność polegająca na braku sprawdzenia ścieżki do pliku podanej przez użytkownika. Umożliwia to odczyt oraz zmianę plików w obrębie całego systemu. Dostęp do plików ograniczony jest uprawnieniami użytkownika który uruchomił aplikację. Z tego powodu tak ważne jest nadawanie wyłącznie niezbędnego minimum uprawnień każdej uruchomionej aplikacji oraz dokładna kontrola danych przesyłanych przez użytkownika.

Cross-site scripting	Problemem w tym przypadku jest ponownie brak sprawdzenia danych wysłanych przez użytkownika. W przypadku gdy dane takie są następnie wyświetlane innym użytkownikom (np. treść postu na forum internetowym) możliwe jest umieszczenie w treści postu kodu wykonywanego po stronie użytkownika (np. JavaScript). Kod taki może mieć za zadanie przechwycenie danych innych użytkowników lub zaatakowanie ich lokalnych zasobów (np. domowego routera).
HTTP header injection	Jest to podatność dotycząca serwerów http. Niektóre serwery przetwarzają wszystkie nagłówki wysyłane przez aplikację, czasem nawet umieszczają ich zawartość w zmiennych środowiskowych. W zależności od konkretnej implementacji serwera, atakujący mógłby użyć tego mechanizmu do zmiany sposobu działania serwera (np. wymuszenie zmiany wersji protokołu komunikacyjnego) lub wstrzyknięcia dodatkowych parametrów do zmiennych środowiskowych.
Time-of-check-to-time-of-use	Jest to przykładowy błąd związany z wyścigami krytycznymi. Problem w tego typu podatnościach polega na odstępie czasowym pomiędzy sprawdzeniem warunku (np. praw dostępu do pliku), a faktycznym użyciem wyników tego sprawdzenia. W czasie pomiędzy tymi operacjami atakujący mógł zmienić stan systemu powodując tym samym błędne działanie programu. W zależności od konkretnego przypadku atak taki może prowadzić nawet do uzyskania uprawnień administratora na danym serwerze.
Cross-site request forgery	Atak na podatność tego typu polega na skłonieniu użytkownika, najczęściej poprzez sfabrykowany link, do wykonania akcji na dowolnej stronie na której dany użytkownik jest uwierzytelniony. Przykładem może być zmiana adresu email używanego do przypominania hasła, a w konsekwencji przejęcie kontroli nad danym kontem przez atakującego.

Lista ta stanowi jedynie wykaz najbardziej typowych problemów występujących w aplikacjach. W żadnym wypadku nie należy jej traktować jako kompletnej listy podatności przed którymi musimy się zabezpieczyć. Najważniejsze to zachować zdrowy rozsądek oraz dokładnie zaplanować logikę działania naszej aplikacji. Warto używać także sprawdzonych i przetestowanych bibliotek oraz narzędzi, minimalizuje to ryzyko poważnych luk bezpieczeństwa. Często jednak do przeprowadzenia udanego ataku wystarczy połączenie kilku potencjalnie niegroźnych podatności naszej aplikacji prowadzące do znacznie poważniejszej luki w zabezpieczeniach.

8.4 Człowiek - najsłabsze ogniwo

Do tej pory poruszaliśmy jedynie kwestie bezpieczeństwa związane ze sprzętem lub oprogramowaniem. Jednak równie często problemem są sami użytkownicy. Brak odpowiedniego przeszkolenia sprawa, że często nie są oni świadomi potencjalnych zagrożeń wynikających z ich działań. Dobór słabych haseł lub zapisywanie ich w widocznych miejscach „dla wygody” sprawia, że nawet najlepsze zabezpieczenia nie są wiele warte.

Istnieje cała gałąź ataków polegających na nakłanianiu użytkowników do wyjawienia danych dostępowych lub wykonania operacji podanej przez atakującego. Ataki te noszą wspólną nazwę ataków socjotechnicznych. Najczęściej spotykaną formą są wszelkiego rodzaju fałszywe e-maile informujące o wygaśnięciu naszego konta lub zmianach na firmowych serwerach. Listy takie zawierają zazwyczaj prośbę o zalogowanie się na podanej stronie w celu potwierdzenia konta na nowym systemie. Pomimo często niegramatycznej składni listu i wyraźnie podejrzanej domeny na której znajduje się strona do zbierania haseł, wiele osób daje się oszukać. Przejęte w ten sposób konta są używane zazwyczaj do rozsyłania jeszcze większej ilości niechcianych listów e-mail.

Powyższy przykład pomimo, że trudno w niego uwierzyć jest dość skuteczny. Można wiec sobie wyobrazić jak skuteczny jest precyzyjnie przygotowany atak socjotechniczny wycelowany w konkretną osobę lub grupę osób. Dlatego bardzo istotną kwestią jest ciągłe szkolenie użytkowników z zakresu bezpieczeństwa informatycznego. Szkolenia te pozwalają uzmysłowić im jak ważną rolę w zapewnieniu bezpieczeństwa danych odgrywają oni sami.

8.5 Podsumowanie

Ten krótki rozdział przybliżył nam charakterystykę osób stojących za atakami na infrastrukturę informatyczną, a także sposoby ich działania. Jest to absolutne minimum wiedzy na ten temat którą należy przyswoić. Przez lata powstało wiele serwisów i publikacji poświęconych tej tematyce, bardziej dociekliwych czytelników zachęcam do przejrzenia wykazu literatury dodatkowej na końcu tej publikacji. Zawiera on kilka ciekawych pozycji poświęconych tej tematyce.

8.6 Pytania kontrolne do rozdziału ósmego

Pyt 1. Podaj kluczowe różnice pomiędzy „trojanem”, a „exploitem”.
Pyt 2. Wymień 5 typowych podatności występujących w aplikacja i krótko scharakteryzuj jedną z nich.
Pyt 3. Dlaczego szkolenie wszystkich pracowników z zakresu bezpieczeństwa informatycznego jest istotne?

9. Analiza ruchu sieciowego i obrona przed zagrożeniami z internetu

Obecnie większość ataków na systemy informatyczne jest przeprowadzana z sieci Internet. Minimalizuje to koszt ataku oraz pozwala na dość łatwe ukrycie jego prawdziwego źródła. Zdecydowana większość atakujących nie jest również nastawiona na konkretny cel. Skanują oni sieć Internet w poszukiwaniu podatnych na atak systemów. Odbywa się to w sposób podobny do tego jak działają roboty wyszukiwarek internetowych. Rozdział ten ma na celu przybliżenie sposobów zbierania i analizy danych o ruchu sieciowym związanym z naszą infrastrukturą.

9.1 Jakie dane możemy pozyskać analizując ruch sieciowy

Każda infrastruktura cechuje się inną strukturą sieci, istnieje jednak szereg elementów wspólnych dla większości z nich. Do podstawowych parametrów opisujących ruch należy zaliczyć ilość danych wpływających i wypływających z sieci, liczbę pakietów na sekundę oraz częstotliwość nawiązywania nowych połączeń. Wszystkie z tych parametrów możemy mierzyć na wielu poziomach. Po pierwsze, najbardziej ogólnie czyli dla całej sieci. Następnie, dla poszczególnych segmentów sieci. I ostatecznie dla pojedynczych serwerów. Dodatkowo możemy także mierzyć te parametry z podziałem na zewnętrzne adresy IP łączące się z naszą siecią. Taki podział pozwoli na wykrywanie skanowania portów lub nietypowych sekwencji pakietów. Jak widać już te trzy proste parametry dają wiele wariantów pomiaru.

Zebranie tych danych pozwala nam na uzyskanie spójnego obrazu naszej infrastruktury. Stosując opisane wcześniej metody analizy danych jesteśmy w stanie automatycznie wykryć nagłe skoki w liczbie nowo nawiązywanych połączeń lub ilości przesyłanych danych. Zjawiska takie często są pierwszą oznaką problemu i zazwyczaj wymagają sprawdzenia. Mogą one być oznaką wykradania firmowych danych lub używania naszych systemów w celu atakowania kolejnych.

Przedstawione wyżej parametry pozwalają opisać wolumen ruchu przepływający przez nasze systemy. Nie mówią nam one jednak nic na temat zawartości. Analiza każdego pakietu pod kątem zawartości jest już zadaniem dużo bardziej wymagającym ze względu na ilość danych oraz ich fragmentacje będącą wynikiem podziału na pakiety. Przy odpowiednich nakładach jest to jednak możliwe. Na rynku dostępnych jest wiele rozwiązań pozwalających na dokładną analizę pakietów. Systemy takie oferują możliwość wykrywania oraz blokowania znanych exploitów, wirusów oraz innego złośliwego oprogramowania. Mogą również analizować dane wychodzące w poszukiwaniu śladów przesyłania firmowych dokumentów. Są to jednak rozwiązania dość drogie, a więc dostępne jedynie dla większych firm oraz instytucji.

9.2 Metody akwizycji danych o ruchu sieciowym

Wiemy już jakie dane moglibyśmy zbierać, pozostaje pytanie jak to robić? Jeśli chodzi o dane ilościowe to doskonałym miejscem jest nasza infrastruktura sieciowa, a więc wszystkie przełączniki sieciowe oraz routery. Zazwyczaj oferują one dostęp do takich informacji za pośrednictwem protokołu sFlow lub netFlow, a także bezpośrednio poprzez protokół SNMP. Dzięki temu w łatwy sposób możemy monitorować ruch na każdym poziomie sieci, od poszczególnych serwerów, poprzez gałęzie sieci do całej infrastruktury.

W przypadku gdy wynajmujemy serwery w innym centrum danych i nie mamy możliwości zbierania informacji z przełączników sieciowych, musimy zdać się na liczniki wbudowane w używany przez nas system operacyjny. Jest to metoda równie dokładna co zbieranie danych na urządzeniach sieciowych. Jednak zużywa ona część zasobów serwera oraz jest dużo bardziej podatna na manipulację w przypadku przejęcia serwera przez atakującego.

Kontrola zawartości pakietów wymaga zdecydowanie większych zasobów. Dlatego jest zazwyczaj realizowana przez dedykowany sprzęt. Możemy tu wyróżnić dwa podejścia. Pierwszym jest użycie nowoczesnej zapory sieciowej umieszczonej bezpośrednio na łączu sieciowym. Przez urządzenie przepływa wtedy każdy pakiet wchodzący i wychodzący z sieci. Może ono sprawdzić go poszukując zdefiniowanych przez użytkownika cech lub sygnatur złośliwego oprogramowania. Taka kontrola musi zostać przeprowadzona w możliwie jak najkrótszym czasie aby nie wprowadzać dodatkowych opóźnień transmisji danych. Drugie podejście to kopiowanie całego ruchu na inną maszynę oraz umieszczenie na łączu jedynie prostej zapory sieciowej pozwalającej na połączenia tylko do wybranych serwerów oraz portów sieciowych. Prostsza zapora wymaga mniejszych zasobów i wprowadza mniejsze opóźnienie. Wydajność dodatkowych maszyn do analizy ruchu nie wpływa wtedy na opóźnienia, użycie zbyt słabego sprzętu spowoduje jedynie problemy z analizą danych. Może to prowadzić do zmniejszenia bezpieczeństwa naszej sieci ale nie zagrozi jej wydajności. Inną wadą jest wolniejsze reagowanie na zagrożenia. Po wykryciu ataku, serwery analizujące dane muszą przesłać informację do zapory sieciowej w celu zablokowania ruchu. Opóźnienie z tym związane dodatkowo zwiększa ryzyko udanego ataku. Zaletą jest zwykle niższy całkowity koszt systemu oraz możliwość stosowania podczas analizy bardziej skomplikowanych reguł.

9.3 Podstawowe metody obrony przed atakami z zewnątrz

Istnieje kilka prostych metod pozwalających na znaczne zwiększenie bezpieczeństwa naszej sieci. Ich implementacja nie jest związana z koniecznością znacznych inwestycji w sprzęt czy oprogramowanie. Z tego powodu warto wprowadzić je w każdej infrastrukturze.

Do najbardziej podstawowych metod należy zaliczyć konfigurację zapory sieciowej wbudowanej w system operacyjny. Powinna być ustawiona na blokowanie całego ruchu i zezwalanie jedynie na minimum połączeń potrzebnych do poprawnego działania. Dla przykładu, dostęp do serwera z bazą danych powinien być ograniczony wyłącznie do serwerów aplikacji. Nie

ma najmniejszej potrzeby aby taki serwer był dostępny dla wszystkich. Dzięki temu wyeliminujemy wiele potencjalnych wektorów ataku na naszą infrastrukturę. Konfiguracja zapory jest operacją prostą i niewymagającą dużych zasobów dlatego też nie należy pomijać tego kroku.

Kolejnym prostym krokiem jest zainstalowanie programu **fail2ban**. Pozwala on na automatyczne blokowanie dostępu do serwera po kilku nieudanych próbach zalogowania. Program analizuje logi zapisywane przez aplikacje w poszukiwaniu śladów błędnych logowań lub innego niewłaściwego zachowania. Po wykryciu problemu blokuje dostęp do serwera dla adresu IP z którego pochodziły błędne zapytania. Przy odpowiedniej konfiguracji może on być używany jako uproszczony system analizy ruchu. Należy jednak pamiętać, że analizuje on logi aplikacji a nie sam ruch sieciowy. Do poprawnego wykrycia problemu konieczne jest, więc zalogowanie błędu przez aplikację.

Przedstawione wyżej metody pozwalają na znaczne zwiększenie bezpieczeństwa i powinny być stosowane w każdej sieci, bez względu na jej rozmiar i przeznaczenie.

9.4 Zaawansowane metody obrony przed atakami z zewnątrz

Istnieje również szereg bardziej zaawansowanych metod obrony wymagających dokładniejszej konfiguracji lub inwestycji w dodatkowy sprzęt. Do najbardziej rozpowszechnionych można zaliczyć web application firewall (w skrócie WAF).

WAF może być dostępne w wielu postaciach. Od darmowych aplikacji instalowanych na serwerze do dedykowanych urządzeń skonstruowanych wyłączenie do tego celu. Niezależnie od wybranego rozwiązania ich działanie polega wyszukiwaniu znanych ataków na aplikacje webowe w przepływającym przez nie ruchu sieciowym. Pozwala to na ochronę przed atakami typu SQL injection oraz innymi typowymi atakami na aplikacje webowe. WAF jest umieszczany bezpośrednio przed serwerami WWW i może być połączony z funkcją rozkładania obciążenia (load balance).

Ogół rozwiązań służących analizie ruchu oraz dynamicznemu blokowaniu ataków nosi nazwę systemu IPS (Intrusion Prevention System). System taki umożliwia automatyczną analizę oraz blokowanie ruchu. Zazwyczaj potrafi on wyszukiwać sygnatury znanych ataków, analizować stan konkretnych połączeń oraz przeprowadzać analizę statystyczną w poszukiwaniu anomalii. System taki może być zbudowany z wielu uzupełniających się komponentów umieszczonych w różnych częściach infrastruktury. Poprawna implementacja systemu IPS wymaga dokładnego zaplanowania i sporego doświadczenia. Błędnie lub niedokładnie skonfigurowany system może nie tylko nie spełniać swojego zadania ale również blokować poprawny ruch kierowany do naszej aplikacji.

9.5 Podsumowanie

Dzięki temu rozdziałowi wiemy już jakie informacje możemy uzyskać analizując ruch sieciowy, jak to zrobić, a co ważniejsze jak tą wiedzę wykorzystać. Cały temat jest bardzo rozległy dlatego zawarto tu jedynie zbiór podstawowych zasad i pojęć dzięki któremu czytelnik będzie w stanie podjąć dalszą samodzielną naukę. Dużą rolę odgrywa tu także doświadczenie, dzięki niemu jesteśmy w stanie przewidywać problemy na długo przed ich wystąpieniem i co ważniejsze odpowiednio się do nich przygotowywać. Z tego powodu przy braku odpowiednich zasobów ludzkich, na początku warto wprowadzić jedynie podstawowe zabezpieczenia opisane w tym rozdziale, a następnie sukcesywnie rozbudowywać je o dodatkowe elementy wraz ze wzrostem wiedzy o naszej infrastrukturze. Pozwoli to na uniknięcie wielu problemów oraz lepsze zrozumienie tego jak dane zabezpieczenie działa i kiedy jest ono najbardziej skuteczne.

9.6 Pytania kontrolne do rozdziału dziewiątego

Pyt 1. Jakie informacje możemy uzyskać analizując ruch sieciowy?
Pyt 2. Podaj najbardziej podstawowe sposoby zabezpieczenia infrastruktury przed atakami z zewnątrz.
Pyt 3. Co to jest WAF i do czego służy?
Pyt 4. Co to jest system IDS i do czego służy?

10. Wpływ automatyzacji na bezpieczeństwo

W przypadku gdy utrzymujemy więcej niż kilka serwerów istotną rolę zaczyna odgrywać automatyzacja procesów związanych z ich utrzymaniem. Sprzęt ulega awariom, oprogramowanie wymaga aktualizacji, nasza aplikacja staje się coraz bardziej popularna i wymaga większej infrastruktury. Każdy z tych przypadków wymaga od nas poświęcenia czasu. Dzięki automatyzacji jesteśmy w stanie sporą część zadań przekazać oprogramowaniu które zajmie się za nas utrzymaniem infrastruktury. Oczywiście konieczne jest wyznaczenie ram działania oraz odpowiednie monitorowanie automatycznych procedur.

Automatyzacja ma również istotny wpływ na bezpieczeństwo. Przykładowo, jeśli odpowiada ona za instalację oprogramowania to możemy być pewni, że poprawki bezpieczeństwa zostaną zainstalowane szybko oraz obejmą wszystkie serwery. W przypadku ręcznej instalacji łatwo pominąć jeden serwer, dodatkowo sam proces będzie długi i monotonny. Automatyzacja dba zatem o spójność naszej infrastruktury. Pozwala także na uniknięcie błędów, jako że każdy system jest identyczny możliwe jest przetestowanie zmian na jednym systemie i automatyczne wprowadzenie ich na wszystkich pozostałych.

10.1 Stopień zaawansowania monitoringu

Automatyzacja utrzymania infrastruktury powinna zawsze iść w parze z jej monitorowaniem. Bez tego moglibyśmy przegapić informację o błędnym działaniu automatu. Jednak monitoring może być mniej lub bardziej zaawansowany. W zależności od stopnia zaawansowania może on oferować różne funkcje oraz integrować się bezpośrednio z systemami automatyzacji. Zazwyczaj możemy wyróżnić następujące funkcjonalności monitoringu:

10.1.1 Zbieranie aktualnych informacji o infrastrukturze

Jest to najbardziej podstawowa funkcjonalność. Jedynie zbieramy dane. Nie ma tu ich przetwarzania, długotrwałego składowania i archiwizacji. Każde dostępne na rynku oprogramowanie do monitoringu posiada tą funkcjonalność. Systemy wyposażone jedynie w tą funkcjonalność wymagają aby operator nieustannie obserwował parametry i wyszukiwał te nie mieszczące się w założonym przedziale. Jest to podejście pracochłonne i niepraktyczne dlatego większość systemów oferuje więcej możliwości.

10.1.2 Składowanie zebranych informacji

Kolejną funkcjonalnością jest możliwość składowania danych w celu ich późniejszej analizy. Systemy takie pozwalają na określenie maksymalnego okresu składowania danych dotyczących każdego z monitorowanych parametrów. Zazwyczaj możliwe jest też zdefiniowanie po jakim czasie

dane ulegną uśrednieniu co pozwala na zaoszczędzenie przestrzeni dyskowej przy zachowaniu informacji o trendzie.

Składowanie dużej liczby parametrów niesie z sobą konieczność zapewnienia odpowiednio wydajnych dysków pozwalających na ciągły zapis napływających danych. Istotna jest również pojemność dysków ponieważ dane mogą szybko przyrastać. Z tego powodu ważne jest odpowiednie dobranie częstotliwości odczytu parametrów oraz czasu ich składowania.

10.1.3 Wykrywanie i składowanie zdarzeń oraz generowanie alarmów

Kolejną często spotykaną funkcjonalnością jest możliwość analizy danych i sprawdzenia czy wartości mieszczą się w zadanych przez operatora przedziałach. Przedziały te mogą być sztywne lub też być wynikiem obliczeń. W drugim przypadku mogą zależeć od wartości parametru w przeszłości. Każde odstępstwo od normy generuje zdarzenie które jest zapisane w systemie monitoringu i może posłużyć do generowania alarmów.

Alarm jest zdefiniowanym przez operatora warunkiem którego spełnienie powoduje wyświetlenie komunikatu, wysłanie wiadomości lub wykonanie akcji. Nie każde zdarzenie musi powodować wywołanie alarmu. Dla przykładu możliwe jest ustawienie w którym tylko zdarzenie utrzymujące się przez dłużej niż 5 minut spowoduje wywołanie alarmu. Odpowiednie zdefiniowanie alarmów pozwala na zwolnienie operatora z konieczności ciągłego śledzenia monitorowanych parametrów. Możliwe jest także zautomatyzowanie obsługi niektórych alarmów dodatkowo minimalizujące czas rozwiązania problemu. Przykładem może być automatyczne uruchomienie kolejnego serwera w przypadku wykrycia zwiększonego ruchu.

W bardziej rozbudowanej wersji zdarzenia i alarmy mogą dotyczyć wielu parametrów jednocześnie. Przykładowo odnotujemy zdarzenie związane z dużym obciążeniem procesora tylko i wyłącznie wtedy kiedy ilość użytkowników naszej aplikacji jest niewielka. W pozostałych przypadkach duże użycie procesora może być oznaką normalnej pracy serwera.

10.1.4 Przewidywanie zdarzeń

Ostatnią i zarazem najbardziej skomplikowaną funkcjonalnością jest przewidywanie zdarzeń oraz reagowanie zanim staną się one problemem. Użycie składowanych danych do wytworzenia systemów eksperckich lub wytrenowania sieci neuronowych może pozwolić na wykrywanie wczesnych oznak problemów zanim spowodują one wykrycie zdarzenia. Dotyczy to zwłaszcza zdarzeń zależnych od wielu parametrów równocześnie.

Dodatkowo systemy takie często znajdują zależności pomiędzy parametrami które zostały przeoczone przez operatora podczas konfiguracji zdarzeń i alarmów. Prowadzi to do jeszcze skuteczniejszego wykrywania problemów, a co za tym idzie szybszego ich rozwiązywania.

10.2 Stopień zaawansowania automatyzacji

Automatyzacja zarządzania może dotyczyć wielu obszarów infrastruktury. Od instalacji systemu, poprzez jego konfigurację do automatycznego przydzielania zadań poszczególnym serwerom. Poziom komplikacji systemu zarządzania rośnie wraz z ilością oferowanych przez niego funkcji. Poniżej przedstawiono typowe zadania realizowane przez takie systemy.

10.2.1 Przygotowanie obrazów systemu

W przypadku gdy posiadamy własne fizyczne serwery bardzo przydatna jest możliwość automatycznego tworzenia obrazów instalacyjnych systemu operacyjnego. Obraz taki, w porównaniu do dostarczanego przez dostawcę systemu, pozbawiony jest zbędnych aplikacji i usług co czyni go mniejszym. Dodatkowo posiada pre instalowane oprogramowanie używane w naszej infrastrukturze. Obrazy takie znacznie przyspieszają instalację nowych serwerów. Automatyzacja ich tworzenia pozwala szybko przygotować nową wersję obrazu zawierającą aktualne poprawki bezpieczeństwa i nowe wersje używanego oprogramowania.

Również większość dostawców serwerów wirtualnych oferuje możliwość uruchamiania nowych instancji z przygotowanego przez użytkownika obrazu systemu. Pozwala to dodatkowo skrócić czas od uruchomienia nowej instancji do rozpoczęcia przez nią obsługi ruchu naszej aplikacji.

10.2.2 Instalacja systemu

Kolejnym obszarem automatyzacji jest instalacja systemu operacyjnego. Dzięki interfejsom zdalnego zarządzania wbudowanym w serwery oraz protokołom takim jak BOOTP (Bootstrap Protocol) oraz specyfikacji PXE (Preboot eXecution Environment) możliwe jest wywołanie zdalnej i automatycznej instalacji systemu operacyjnego. Dotyczy to zarówno nowych serwerów jak i będących w użytkowaniu. Dzięki temu jesteśmy w stanie szybko i bez wysiłku przywrócić każdy serwer do znanego stanu bazowego.

Problem ten nie dotyczy użytkowników środowisk wirtualnych ponieważ tam obraz systemu jest kopiowany podczas inicjalizacji instancji. Jest on jednak bardzo widoczny u dostawców serwerów wirtualnych, ponieważ muszą oni zapewnić fizyczne serwery zdolne uruchamiać kolejne instancje serwerów wirtualnych. Dobrze działający proces tworzenia obrazów i instalacji serwerów jest podstawą ich działalności.

10.2.3 Konfiguracja systemu

Po poprawnej instalacji systemu konieczne jest jeszcze wykonanie jego ostatecznej konfiguracji. Musimy nadać mu adres IP, przypisać nazwę, dołączyć poprawnie do systemu monitoringu oraz założyć konta użytkowników (o ile nie były umieszczone w obrazie systemu).

Wszystkie te operacje są specyficzne dla każdej infrastruktury, dlatego też mogą wymagać innych metod realizacji. Są one jednak stosunkowo proste, dlatego też ich automatyzacja nie powinna sprawić problemu.

10.2.4 Instalacja i konfiguracja aplikacji

Dzięki wszystkim poznanym do tej pory zadaniom stawianym przed systemami automatyzacji uzyskaliśmy działający gotowy do użycia serwer. Posiada on jednak jedynie bazowy system operacyjny oraz konfigurację podstawowych usług wspólnych dla wszystkich maszyn w naszej infrastrukturze. Kolejnym krokiem jest, więc umieszczenie na nim naszej aplikacji (lub innej wymaganej przez nią usługi takiej jak baza danych). Zadanie to może być realizowane na wiele sposobów, jednak będzie o wiele prostsze w realizacji gdy pomyślimy o nim już na etapie wytwarzania aplikacji.

Dokładnie zdefiniowane zależności, dynamiczna konfiguracja aplikacji czy jej bezstanowość pozwolą na szybkie i bezproblemowe wprowadzenie automatyzacji. Brak planowania automatycznej instalacji oraz konfiguracji już na etapie projektowania aplikacji może znacznie skomplikować jej późniejsze utrzymanie. Dobre rezultaty przynosi przynajmniej częściowe połączenie firmowych zespołów odpowiedzialnych za wytwarzanie i utrzymanie oprogramowania. Pozwoli to wesprzeć proces automatyzacji już na samym początku wytwarzania aplikacji.

Do poprawnej konfiguracji aplikacji wykorzystywane są systemy wykrywania usług (z j. ang service discovery). Pozwalają one każdej działającej aplikacji na zarejestrowanie oferowanych przez nią zasobów oraz informacji potrzebnych do uzyskania do nich dostępu. Serwer bazy danych może wykorzystać ten mechanizm do poinformowania aplikacji o swojej lokalizacji. Dzięki temu aplikacja nie musi posiadać tej informacji w swoich plikach konfiguracyjnych co upraszcza jej instalację i uruchomienie. Sama aplikacja może zarejestrować swoją obecność tak aby serwery odpowiedzialne za dystrybucję ruchu (z j. ang load balancer) zaczęły kierować do niej żądania od użytkowników. System monitoringu może stąd również czerpać informacje o tym co powinien monitorować na poszczególnych serwerach.

10.2.5 Przydzielanie aplikacji do puli zasobów w zależności od obciążenia

Gdy nasz system automatyzacji posiada wszystkie wymienione wcześniej funkcje możemy rozpocząć integrowanie go z systemem monitoringu. Dzięki temu jesteśmy w stanie automatycznie dodawać lub odejmować serwery z puli zasobów w zależności od aktualnego obciążenia. W przypadku fizycznych serwerów możemy dzięki temu zmniejszyć zapotrzebowanie na energię

elektryczną i moc chłodniczą. Przekłada się to pośrednio na koszt utrzymania infrastruktury. W przypadku rozwiązań wirtualnych w których płacimy za czas pracy instancji, wyłączenie serwera powoduje natychmiastowe zmniejszenie kosztów.

Poza tym nie musimy już zastanawiać się kiedy uruchomić kolejny serwer i czy w nocy nie pojawi się dodatkowy ruch. Posiadając kompletny i przetestowany system automatyzacji mamy pewność, że nasza infrastruktura dostosuje się do aktualnych potrzeb. Dodatkowo problemy związane z awariami poszczególnych serwerów będą rozwiązywane automatycznie. Wychwycenie awarii przez system monitoringu spowoduje uruchomienie kolejnego serwera realizującego to samo zadanie.

10.3 Systemy kontroli wersji

Nieodłączną częścią automatyzacji są systemy kontroli wersji. Nazwę tę noszą wszystkie narzędzia oferujące programistom możliwość śledzenia i komentowania zmian w kodzie źródłowym. Każda zmiana kodu może zostać opisana i zwersjonowana tak aby nawet po długim czasie możliwy był powrót do poprzedniej wersji. Poza wyżej wymienioną funkcjonalnością ułatwiają one także pracę zespołową. Każdy członek zespołu umieszcza swoje zmiany w systemie kontroli wersji, gdzie są one łączone w jedną spójną całość. Możemy także sprawdzić jakie zmiany wykonali inni członkowie zespołu oraz przeczytać umieszczone przez nich komentarze dotyczące tych zmian.

Systemy kontroli wersji dzielą się na dwa podstawowe rodzaje: centralne oraz rozproszone. W pierwszym przypadku każda wersja programu musi być natychmiast wysłana na centralny serwer. Programiści nie mają możliwości wersjonowania swoich zmian bez dostępu do serwera. Do typowych przedstawicieli należy zaliczyć takie programy jak SVN czy Perforce. W drugim przypadku każdy programista posiada swoją kopię całego repozytorium kodu. Daje to możliwość pracy nawet w przypadku braku dostępu do sieci. Członkowie zespołu mogą w każdej chwili wymienić się zmianami aby uzyskać spójny kod aplikacji. Typowymi przedstawicielami są Git oraz Mercurial. Każde z podejść ma swoje wady oraz zalety dlatego przed ostatecznym wyborem powinniśmy przetestować obydwa podejścia i wybrać to które lepiej pasuje do stylu pracy naszego zespołu.

10.4 Ciągła integracja

Elementem łączącym systemy kontroli wersji z automatyzacją utrzymania jest proces ciągłej integracji (z j. ang continuous integration – CI). Jest to proces w którym każda odpowiednio oznaczona zmiana kodu w systemie kontroli wersji może wywołać zdefiniowaną wcześniej akcję. Akcją może być na przykład zbudowanie aplikacji na podstawie aktualnego kodu źródłowego, przetestowanie jej działania za pomocą testów automatycznych i w przypadku ich powodzenia umieszczenie nowej wersji aplikacji na serwerach. Podejście takie znacznie skraca czas potrzebny do wprowadzenia zmiany widocznej na serwerach. Dzięki temu zmiany są mniejsze i dotyczą

zazwyczaj jednej konkretnej funkcjonalności naszej aplikacji. Pozwala to na dokładniejsze śledzenie wpływu zmiany na zachowanie aplikacji. Upraszcza również ewentualne wycofanie zmiany, wystarczy przebudować aplikację z użyciem poprzedniej wersji kodu pobranej z systemu kontroli wersji.

Takie podejście wymaga jednak odpowiedniego przygotowania aplikacji. W przypadku działania na wielu serwerach musimy pamiętać o tym aby umożliwić działanie starej i nowej wersji jednocześnie. Dla przykładu zmiana schematu bazy danych wymagana przez nową wersję aplikacji może spowodować błędy w działaniu starej wersji. Problemów tego typu może być o wiele więcej dlatego istotne jest odpowiednie przygotowanie aplikacji oraz stworzenie środowiska testowego w którym będziemy mogli sprawdzić nową wersję aplikacji przed umieszczeniem jej na serwerach produkcyjnych.

10.5 Podsumowanie

Dzięki informacjom zawartym w tym rozdziale dowiedzieliśmy się jaki wpływ na bezpieczeństwo ma automatyzacja instalacji. Poznaliśmy również obszary naszej infrastruktury których może ona dotyczyć. Dowiedzieliśmy się również dlaczego istotne jest planowanie automatyzacji już na samym początku procesu wytwarzania oprogramowania. Dopracowane i przetestowane systemy automatyzacji pozwalają znacząco ograniczyć zasoby potrzebne do utrzymania aplikacji. Oszczędności z tego tytułu możemy, więc przeznaczyć na dalsze doskonalenie swojego produktu.

Poznaliśmy również systemy kontroli wersji pozwalające na dokładne śledzenie zmian wprowadzanych w kodzie naszej aplikacji. Poza przedstawionymi już funkcjami pozwalają one także na łatwe wdrożenie kontroli kodu. Osoby odpowiedzialne za bezpieczeństwo mogą w prosty sposób prześledzić zmiany wprowadzone w aplikacji i ocenić ich wpływ na bezpieczeństwo.

10.6 Pytania kontrolne do rozdziału dziesiątego

Pyt 1. Jak automatyzacja zarządzania infrastrukturą wpływa na jej bezpieczeństwo?
Pyt 2. Do czego służą systemy kontroli wersji?
Pyt 3. Wymień zalety i wady procesu ciągłej integracji.

11. Co to jest audyt bezpieczeństwa i po co się go wykonuje.

W ostatnim rozdziale przedstawione zostaną pojęcia związane z audytem bezpieczeństwa oraz normą ISO opisującą standardy zarządzania bezpieczeństwem IT. Będziemy się tu skupiać głównie na aspekcie formalnym zarządzania bezpieczeństwem, czyli procedurach i procesach związanych z infrastrukturą informatyczną.

Istnieje wiele różnych zbiorów procedur związanych z bezpieczeństwem IT. Dlatego nie ma konieczności wdrażania od razu wszystkich zaleceń normy ISO. Można to robić stopniowo, obejmując coraz to nowe obszary naszej działalności. Warto jednak zaznajomić się z zaleceniami normy już na samym początku. Pozwoli to uzmysłowić sobie wiele potencjalnych problemów na które możemy natrafić w przyszłości.

Posiadanie takiego zbioru procedur pozwala na sprawne przeprowadzenie audytu bezpieczeństwa. Audyt taki pozwala sprawdzić czy nasze procedury są stosowane w rzeczywistości oraz czy przynoszą spodziewany efekt. Dzięki tym informacjom jesteśmy w stanie zidentyfikować obszary wymagające zmian.

11.1 Norma ISO 27001

Najnowsza wersja normy jest oznaczona jako ISO 27001:2013. Składa się ona z dziesięciu działów opisujących różne etapy procesu zarządzania ryzykiem w kontekście bezpieczeństwa. Poza tym dołączony jest do niej obszerny dodatek który zawiera listę kontrolną wszystkich procedur i dokumentów które powinny być wdrożone w celu uzyskania certyfikatu zgodności z normą.

Na podstawie tego dokumentu powinniśmy być w stanie zaprojektować i wdrożyć system zarządzania bezpieczeństwem informacji (z j. ang: ISMS - information security management system). System taki ma za zadanie standaryzację obsługi incydentów związanych z bezpieczeństwem danych, zebranie statystyk i wskaźników dotyczących incydentów oraz kontroli dostępu do danych. Zebrane statystyki i wskaźniki pozwolą na dalsze udoskonalanie procedur i związanych z nimi systemów bezpieczeństwa.

Uzyskanie przez firmę lub instytucję certyfikacji w zakresie normy ISO27001 jest wyraźnym sygnałem, że dba ona o bezpieczeństwo danych. Może być to także wymagane przez naszych klientów, chcących składować dane wrażliwe w naszych systemach. Dane medyczne jak wiadomo są uznawane za wrażliwe i poufne. Dlatego konieczne jest zapewnienie odpowiedniej kontroli dostępu do nich oraz zabezpieczenia przed utratą oraz kradzieżą. Uzyskanie certyfikacji ISO27001 pokazuje, że problem ten nie jest nam obcy, a dążenie do jego rozwiązania jest częścią naszej działalności.

11.2 Procedury wewnętrzne

Niezależnie od tego czy zdecydujemy się wprowadzać procedury związane z normą ISO, czy też nie, w naszej firmie znajdzie się prawdopodobnie szereg procesów mających wpływ na bezpieczeństwo danych. Zidentyfikowanie i opisanie tych procesów może odkryć wiele braków i niedociągnięć których nie byliśmy wcześniej świadomi.

Procesem w tym przypadku może być na przykład przyjmowanie nowego pracownika. Powinniśmy określić kto występuje o utworzenie nowego konta w naszych systemach informatycznych, kiedy to następuje, jak szeroki dostęp jest wymagany, w jaki sposób zostaną przekazane dane potrzebne do zalogowania w systemie, itp. Są to kwestie istotne ponieważ termin przekazania nowego konta lub przyznania konkretnych uprawnień może zależeć od tego czy nowy pracownik ukończył już szkolenie związane z obsługą systemu. Opisanie wszystkich tych kroków w jednym dokumencie da nam kompletny obraz tego procesu. Pozwoli także na utworzenie list kontrolnych dzięki którym szybko sprawdzimy, czy żaden z wymaganych kroków nie został pominięty. Podobny opis powinniśmy wytworzyć dla wszystkich procesów związanych z obsługą danych w naszej firmie. Poza już wymienionym do najbardziej typowych należy zaliczyć: odejście pracownika, wykonywanie kopii zapasowych, odtwarzanie danych z kopii zapasowych, przyznawanie i odbieranie dostępu do poszczególnych usług, konfiguracja nowej stacji roboczej, i tym podobne. Zbiór wszystkich tych procedur pozwoli nam na sprawne przeprowadzenie audytu wewnętrznego i wskazanie elementów wymagających poprawy.

11.3 Przebieg audytu

Audyty bezpieczeństwa możemy podzielić na dwa podstawowe rodzaje: wewnętrzne oraz zewnętrzne. W pierwszym przypadku audyt jest przeprowadzany we własnym zakresie, najczęściej przez pracowników firmy. Ma on na celu sprawdzenie czy opisane procedury są stosowane poprawnie oraz czy są w dalszym ciągu aktualne. Audyt zewnętrzny wykonywany jest przez zewnętrzną firmę, nie związaną w żaden sposób z naszą działalnością. Jego celem może być chęć sprawdzenia naszych procedur przez inne osoby. Podejście takie pozwala na wychwycenie braków i niedociągnięć które zostały (celowo lub nie) przeoczone przez naszych pracowników. Takie krytyczne spojrzenie może wnieść wiele nowych cennych uwag do istniejących procedur. Innym celem audytu zewnętrznego może być chęć uzyskania wspomnianej wcześniej certyfikacji zgodności z normą.

Niezależnie od rodzaju i celu audytu ma on podobny przebieg. Pierwszym krokiem jest zazwyczaj zaplanowanie przebiegu audytu. Jest to istotne ponieważ testowanie procedur związanych na przykład ze składowaniem i odzyskiwaniem danych może mieć bezpośredni wpływ na wydajność oraz dostępność naszych systemów. Dlatego istotne jest odpowiednie rozplanowanie testów tak aby w jak najmniejszym stopniu wpływały one na normalną pracę firmy. Należy także określić w jakim celu dany audyt zostanie przeprowadzony oraz oraz jakich obszarów naszej

działalności będzie dotyczył. Pozwoli to na oszacowanie czasu koniecznego do przeprowadzenia audytu i opracowania wyników.

Kiedy znany jest już zakres i cel audytu, audytor na podstawie dokumentacji sporządza listy kontrolne zawierające wykaz wszystkich elementów wymagających sprawdzenia. Układając listy kontrole audytor powinien kierować się dwoma zasadniczymi pytaniami: „Co powinienem/powinnam sprawdzić?” oraz „Dlaczego powinienem/powinnam to sprawdzić?” („Jaką informację uzyskam?”). Dobrze ułożona lista pozwoli na szybkie oraz sprawne przeprowadzenie audytu. Dodatkowo usprawni ona proces opracowywania wyników audytu.

Właściwy audyt rozpoczyna się od spotkania mającego na celu przedstawienie audytowanej organizacji planu, zakresu oraz metodyki audytu. Konieczne jest tutaj upewnienie się, że audytowana organizacja rozumie cel i zakres wykonywanego audytu oraz, że zatwierdza jego plan. Jest to także okazja do ustalenia wszelkich kwestii technicznych związanych z przeprowadzanym audytem. Mogą to być sprawy związane z dostępem do pomieszczeń, czy dokumentów wymaganych do prawidłowego przeprowadzenia audytu.

Po spotkaniu otwierającym audytor przystępuje do sprawdzenia wszystkich elementów z przygotowanych wcześniej list kontrolnych. Może to obejmować zarówno sprawdzenie czy organizacja posiada opracowaną procedurę jak również przetestowanie działania danej procedury w rzeczywistości.

Gdy odpowiedzi na wszystkie zawarte na listach kontrolnych pytania zostaną już zebrane następuje etap analizy danych oraz przygotowywania raportu końcowego. Raport taki powinien przedstawiać wnioski w sposób jasny oraz zrozumiały ponieważ zazwyczaj adresowany jest do zarządu danej organizacji, a więc osób niekoniecznie zaznajomionych z technologiami informatycznymi. Powinien on zawierać wykaz niezgodności i odstępstw od normy lub wewnętrznych procedur. Każda niezgodność powinna odnosić się do konkretnego punktu normy lub procedury wewnętrznej organizacji. Dokument ten ma na celu wskazanie obszarów wymagających poprawy. Dlatego też opis problemu powinien być konkretny i dokładny. Pozwoli to na łatwe i szybkie określenie koniecznych do wykonania działań korygujących.

11.4 Podsumowanie

Dokładne sprecyzowanie oraz opisanie procedur i procesów występujących w naszej organizacji daje nam wiedzę na temat tego w jaki sposób zarządzamy danymi. Pozwala także na wprowadzenie pewnych standardów dotyczących dostępu do danych, ich przetwarzania oraz składowania. Dzięki systematycznym audytom możemy się upewnić, że wytworzone procedury są stosowane poprawnie oraz przynoszą zamierzony efekt. Odkrywamy także obszary wymagające wytworzenia nowych procedur lub aktualizacji już istniejących.

Pomimo, że wszystkie opisane w tym rozdziale zagadnienia dotyczą wyłącznie procedur oraz dokumentacji to mają one bezpośredni wpływ na zastosowane rozwiązania techniczne oraz ich konfigurację. Dlatego też wszystkie znaczące zmiany w infrastrukturze powinny nieść ze sobą konieczność sprawdzenia aktualności procedur np. poprzez wykonanie audytu wewnętrznego.

Dzięki temu będziemy w stanie sprawdzić wpływ wprowadzonej zmiany na bezpieczeństwo danych.

11.5 Pytania kontrolne do rozdziału jedenastego

Pyt 1. Jaka norma definiuje zagadnienia związane z bezpieczeństwem informacji?
Pyt 2. Co to jest audyt bezpieczeństwa i w jakim celu się go wykonuje?
Pyt 3. Opisz w punktach przebieg typowego audytu bezpieczeństwa.

Literatura

- Kryptografia dla praktyków. Protokoły, algorytmy i programy źródłowe w języku C., Bruce Schneier, ISBN 83-204-2678-2
- Cisza w sieci, Michal Zalewski, ISBN 83-7361-659-4
- Splątana sieć. Przewodnik po bezpieczeństwie nowoczesnych aplikacji WWW, Michał Zalewski, ISBN 978-83-246-4477-3
- Sztuka podstępu, Kevin Mitnick, William Simon, ISBN 978-83-246-2795-0
- Linux. Bezpieczeństwo. Receptury, Daniel J. Barrett, Richard E. Silverman, Robert G. Byrnes, ISBN 83-7361-249-1
- Podstawy systemów operacyjnych, Abraham Silberschatz, Peter B. Galvin, Greg Cagne, ISBN 83-204-3215-4

Dodatek 1 – wykaz linków

1) http://www.renesys.com/2013/11/mitm-internet-hijacking/ (sprawdzony 6.1.2015)
2) http://googleonlinesecurity.blogspot.com/2011/08/update-on-attempted-man-in-middle.html (sprawdzony 6.1.2015)
3) https://www.openssl.org/docs/ (sprawdzony 6.1.2015)