POLITECHNIKA SZCZECIŃSKA

WYDZIAŁ INFORMATYKI

Rozprawa doktorska

Wyznaczenie bezpieczeństwa sieci komputerowej poprzez analizę czasową zdarzeń

mgr inż. Grzegorz Śliwiński

Promotor:

prof. dr hab. inż. Jerzy Korostil

Szczecin – 2008

Spis treści

1. Wstęp

W pracy sformułowano nową metodę wyznaczenia bezpieczeństwa sieci komputerowej z wykorzystaniem autorskich współczynników: odporności, otwartości i przeciążalności systemu oraz zaproponowaną zmodyfikowaną metodę analizy ryzyka sieciowego za pośrednictwem drzew binarnych (drzew zdarzeń i błędów), która pozwala na analizę ryzyka w czasie rzeczywistego działania systemu. Zgodnie z tematem pracy tj. „Wyznaczenie bezpieczeństwa sieci komputerowej poprzez analizę czasową zdarzeń" zaproponowano nowe pojęcie ***poziomu gwarancji bezpieczeństwa****, którego wartość wyznaczona może być na podstawie analizy bezpieczeństwa i ryzyka systemu.*

Bezpieczeństwo i ryzyko sieci komputerowej to dwa podstawowe i ściśle związane ze sobą parametry opisujące podstawowe cechy dowolnej sieci lub jej elementu składowego. Nie wyobrażamy sobie dostępu do dowolnego punktu w sieci bez odpowiednich zabezpieczeń i uprawnień. Staramy się chronić swoje informacje lub udostępniać je wyłącznie w sposób kontrolowany.

Każdy system sieciowy a nawet każda sieć komputerowa zmienia swoje parametry w czasie działania. Zmiana warunków działania również wpływa na zmianę bezpieczeństwa i ryzyka. Naturalnym zatem staje się oczekiwanie od dowolnego elementu sieci gwarancji poprawności i pewności działania.

Na podstawie ogólnych, wcześniej zaprezentowanych sformułowań, głównym **celem pracy jest**:

Opracowanie metody wyznaczenia poziomu gwarancji bezpieczeństwa w sieciach komputerowych w czasie rzeczywistym.

Metoda naukowa to określona procedura, która powinna być stosowana w procesie pozyskiwania lub tworzenia rzetelnej wiedzy naukowej. Zasadniczym dla niej jest kryterium falsyfikowalności[1]. Metoda naukowa jest też zbiorem zasad, na podstawie których przyjmuje się lub odrzuca analizowane hipotezy lub opisy zjawisk. Przykładanie miary tych zasad do określonych teorii czy opisów decyduje o tym, czy zostaną one uznane za rzetelną wiedzę naukową.

Metoda jest to również zespół teoretycznie uzasadnionych zabiegów koncepcyjnych i instrumentalnych obejmujących na ogół całość postępowania badacza zmierzającego do rozwiązania określonego problemu naukowego. Metoda jest pojęciem najszerszym w stosunku do techniki i narzędzia badawczego. Termin „metoda" można stosować w dwóch znaczeniach. W znaczeniu szerszym jest to próba ogólnego określenia charakteru i zakresu danych badań np. metoda monograficzna, metoda historyczno-porównawcza. W węższym znaczeniu metoda służy do określani powtarzalnych sposobów rozstrzygania konkretnych zagadnień, związanych z realizacją badań przez określoną metodę. Jest to całokształt czynności określających sposób zbierania danych i ich opracowania.

W pracy przyjęto, że możliwe jest gromadzenie informacji na temat wykonywanych działań w systemach sieciowych oraz samej sieci i możliwe jest przetwarzanie tej informacji. Przyjęte założenie pozwala na prowadzenie odpowiednich działań pozwalających na wykonanie niezbędnych wyliczeń do wyznaczenia podstawowych składowych rozpatrywanego systemu. Założenia są możliwe do wykonania ze względu na zdolności do monitorowania zdarzeń w sieciach i systemach sieciowych w odpowiednich dziennikach zdarzeń (rejestrach systemowych).

[1] Falsyfikacja (łac. falsum - fałsz) - jest to odmiana jednego z rozumowań zwanego sprawdzaniem.

Przyjęto zatem tezę, że:

Na podstawie analizy ryzyka i wartości bezpieczeństwa możliwe jest określenie poziomu gwarancji bezpieczeństwa sieci komputerowej.

Do realizacji tezy potrzeba wyznaczyć dwie składowe (bezpieczeństwo i ryzyko) oraz określić podstawy do wyznaczenia poziomu gwarancji bezpieczeństwa. W drugim rozdziale pracy zaprezentowano obecne osiągnięcia i zdefiniowano podstawowe pojęcia niezbędne do zrozumienia zakresu opisywanej problematyki. Zostały tam poruszone podstawowe problemy obecnych sieci komputerowych, zagrożenia i metody ochrony przed nimi. Głównym aspektem podsumowującym podjętym w tym rozdziale jest stwierdzenie, że dostępne środki bezpieczeństwa bazują jedynie na przygotowaniu systemu sieciowego do uruchomienia w środowisku sieciowym i nie uwzględniają czynników powstałych w czasie eksploatacji tego systemu.

Autorskie rozwiązanie tego zagadnienia zostało zaprezentowane w rozdziale trzecim, który opisuje warunki i możliwości wykonania niezbędnych obliczeń oraz prezentuje zależności matematyczne pozwalające wyznaczeń wartości bezpieczeństwa, ryzyka oraz poziomu gwarancji bezpieczeństwa dowolnej sieci. Zaproponowane w podrozdziale 3.1 nowatorskie podejście do wyznaczenia bezpieczeństwa na podstawie przyjętych trzech współczynników (tj. odporność, otwartość, przeciążalność) pozwala na pełne, a zarazem jasne określenie składowych bezpieczeństwa. Kolejnym etapem było przystąpienie do opisu ryzyka oraz zaproponowanie możliwości wyznaczenia jego wartości. Opisana metoda z podrozdziale 3.2 reprezentacji drzew zdarzeń i błędów stosowana jest we wielu dziedzinach (również z zakresie sieci komputerowych) jednak nowym

podejściem jest zastosowanie analizy zdarzeń w czasie rzeczywistym oraz możliwość dalszej rozbudowy proponowanego rozwiązania o procesy automatyzacji tworzenia i wyznaczania wartości ryzyka systemu sieciowego.

W ostatnim z podrozdziałów tj. 3.3 przedstawiono podstawę wyznaczenia wartości poziomu gwarancji bezpieczeństwa. Jest to podejście nowatorskie do zagadnienia bezpieczeństwa sieci komputerowych. Autor rozszerza w następnym rozdziale tj. 4.3 tą tematykę stosując prawdopodobieństwo i interpolację Lagrange'a do przewidywania zachowania się systemu, a tym samym wyznaczenia poziomu gwarancji bezpieczeństwa nie tylko w czasie rzeczywistym ale również w czasie który nastąpi z określonym poziomem ufności zależnym od wielkości czasu przewidywanego działania systemu sieciowego.

W przytaczanym czwartym rozdziale zaprezentowano przykładowe możliwości wyznaczenia opisywanych w trzecim rozdziale współczynników poprzez zastosowanie autorskich narzędzi prezentowanych w kolejnym piątym rozdziale.

Autor nie wyczerpał całego zagadnienia i w pracy zostały przedstawione główne nurty pozwalające na określenie całości problemu i sposobu jego rozwiązania. Zaprezentowane badania zostały zrealizowane w warunkach laboratoryjnych z wykorzystaniem jako bazy informacji rzeczywistego systemu sieciowego działającego w dużej strukturze firmowej.

2. Bezpieczeństwo systemów - wprowadzenie

W drodze do społeczeństwa informacyjnego, do którego niewątpliwie zmierzamy, wspierają nas nowe technologie i nowe wynalazki. Wspomnieć tu należy o powszechnej informatyzacji codziennego życia, o rozwoju wszechobecnego Internetu i gospodarce elektronicznej. Powszechne stały się telefonia komórkowa, bankowość elektroniczna oraz karty płatnicze, a jak wiadomo ich podstawą są zaawansowane technologie teleinformatyczne.

Rozwój tych technologii przynosi wszystkim znaczące korzyści, jednak wiąże się z nimi również szereg niebezpieczeństw – temu rozwojowi towarzyszą, bowiem całkiem nowe formy zagrożeń dla jednostek, instytucji, a nawet dla całych społeczeństw.

Im intensywniej pragniemy czerpać korzyści z teleinformatyki, tym bardziej musimy otworzyć na świat nasze systemy. Czyniąc to bez odpowiednich środków, narażamy się na przeróżne niebezpieczeństwa. Znaczące uzależnienie się instytucji od eksploatowanych w niej systemów teleinformatycznych wprowadza dla jej procesów biznesowych nowe kategorie zagrożeń - zagrożenia specyficzne dla teleinformatyki.

Z faktu istnienia wielu zagrożeń, jakie niosą dla nas wszystkich nowoczesne technologie teleinformatyczne nie wynika wcale, że powinniśmy z nich rezygnować, musimy jedynie nauczyć się czerpać z nich korzyści, ograniczając ryzyko wystąpienia niekorzystnych zjawisk, a nawet strat.

Informacja staje się zasobem strategicznym, ale nie każda, tylko taka, która: niesie w sobie istotne treści, jest dostarczana na czas, do właściwego adresata, w formie nienaruszonej i niezawodnie – po prostu bezpiecznie.

Z tym strategicznym charakterem informacji zaczynają się oswajać państwa, przeróżne instytucje, a nawet pojedynczy człowiek.

Skoro informacje mają coraz większą wartość, to również coraz więcej pojawia się związanych z nimi nadużyć, a nawet przestępstw. Przestępczość komputerowa posiada międzynarodowy zasięg, organa ścigania i prawodawstwo w większości krajów nie są w zasadzie przygotowane do jej zwalczania. Użytkownicy w chwili obecnej obawiają się najbardziej:

- utraty poufności swoich danych osobowych,
- ujawnienia numeru swojej karty płatniczej,
- niefachowych porad lekarskich udzielanych poprzez Internet,
- zawirusowania systemu,
- otrzymania z sieci fałszywych wiadomości, w tym pogłosek dotyczących notowań giełdowych.

2.1. Aspekty bezpieczeństwa

Polityka bezpieczeństwa jest to szeroko rozumiany plan lub sposób działania, przyjęty w celu osiągnięcia wysokiego poziomu bezpieczeństwa systemu informatycznego oraz ochrony danych, realizowany jako skutek podjętych decyzji administracyjnych. Określa ona zbiór reguł i warunków dotyczących sposobu, w jaki zarządza się informacją, chroni ją i rozdziela. Możliwe są tu dwa podejścia projektowe:

- to, co nie jest zabronione, jest dozwolone,
- to, co nie jest dozwolone, jest zabronione.

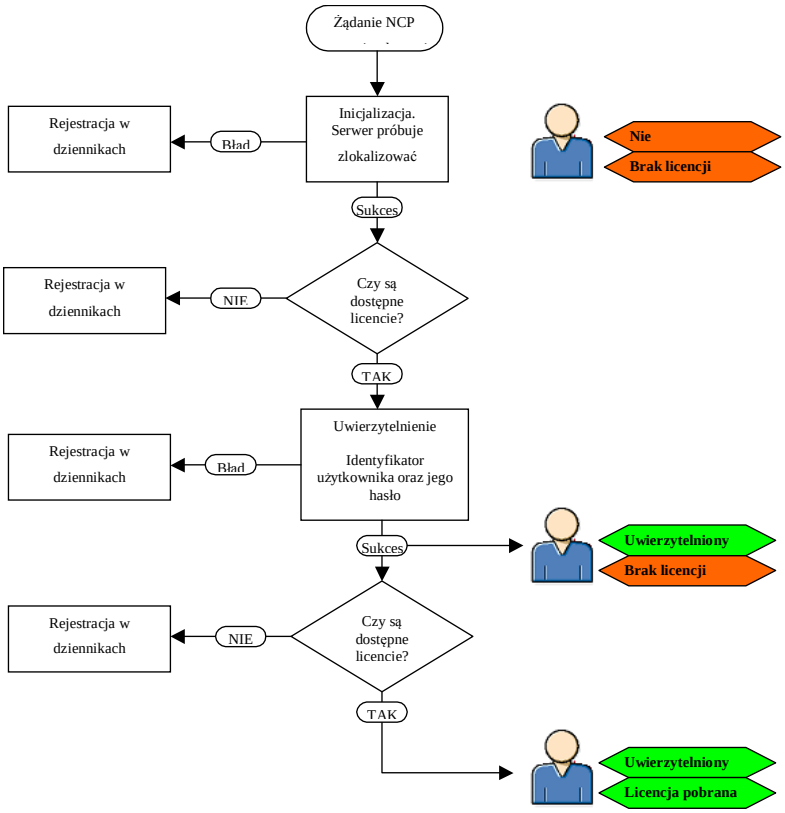
Druga forma projektowa jest bardziej restrykcyjna – co za tym idzie bardziej bezpieczna – jednak wymaga od projektującego bardzo dobrego

rozeznania (w fazie założeń) środowiska, dla jakiego dany projekt zostanie stworzony [14]. Jest to proces bardzo trudny. Kwestie bezpieczeństwa można podzielić na dwa różne punkty widzenia. Z jednej strony mówić możemy o Identyfikacji, Uwierzytelnieniu i Autoryzacji, z drugiej zaś będzie to Poufność, Integralność i Dostępność. Różne rodzaje bezpieczeństwa w sposób sformalizowany, precyzyjny i jednoznaczny wyrażają te aspekty polityki bezpieczeństwa, których realizacja jest wymagana w systemach informatycznych. Rzeczywistość zaś opisują w terminach jednostek organizacyjnych, podmiotów i obiektów. Pod pojęciem podmiotu rozumie się użytkownika lub proces działający w jego imieniu. Obiektem jest element bierny, który można wybrać i korzystać z niego lub manipulować nim [18] (na przykład: pliki, foldery, urządzenia wejścia/wyjścia, itp.).

Identyfikacja polega na stwierdzeniu tożsamości podmiotu lub obiektu. Identyfikator to niepowtarzalna nazwa lub numer przypisany podmiotowi lub obiektowi. Identyfikator podaje tylko niepotwierdzoną tożsamość. Identyfikatory są konieczne dla rozliczeń i autoryzacji, ale nie mogą być używane bez dodatkowego uwierzytelnienia podmiotów, jeśli w systemach pożądany jest jakikolwiek stopień bezpieczeństwa. Uwierzytelnienie podmiotów ma bezpośredni wpływ na bezpieczeństwo systemów [21] (Rysunek 1).

Uwierzytelnienie jednostki (podmiotu lub obiektu) służy sprawdzeniu, czy jednostka jest tym, za kogo się podaje, czy jest autentyczna. Gdy jednostka J1 zostanie uwierzytelniona wobec jednostki J2, to mówimy także, że nastąpiła identyfikacja jednostki J1 przez jednostkę J2. Oprócz uwierzytelnienia jednostek w systemach komputerowych mamy do czynienia z uwierzytelnieniem danych. Polega ono na potwierdzeniu autentyczności pochodzenia danych ze względu na określone źródło

danych i ich integralność. Dane uwierzytelnia się za pomocą podpisu cyfrowego, funkcji skrótu z kluczem kryptograficznym lub znacznika danych, który jest szyfrowany razem z danymi [45].



Rysunek 1 Przykład uwierzytelnienia użytkownika

Autoryzacja jest pozwoleniem wydanym przez instancję nadrzędną (autorytet). Polega na otrzymaniu dostępu do obiektów systemu informatycznego w oparciu o identyfikator. Autoryzacja jest czymś innym niż uwierzytelnienie. W procesie uwierzytelnienia sprawdza się jedynie tożsamość jednostki, ale nie mówi się nic o jej uprawnieniach dostępu do obiektów [61]. Jednostkę można identyfikować za pomocą protokołu uwierzytelnienia, a w najprostszym przypadku poprzez sprawdzenie jej identyfikatora (z punktu widzenia bezpieczeństwa systemu jest to sytuacja najgorsza, polegająca na udzieleniu dostępu na przykład do systemu bez

żadnej dodatkowej weryfikacji, bez sprawdzenia choćby hasła użytkownika. Z tego powodu wynika, że w odniesieniu do podmiotu:

- identyfikator jest niezbędny do rozliczeń i autoryzacji,
- w procesie uwierzytelnienia sprawdza się tożsamość podmiotu,
- po uwierzytelnieniu podmiotu może on korzystać z podmiotów lub obiektów, do których uzyskał uprawnienia od podmiotu nadrzędnego.

Bezpieczeństwo danych polega na zabezpieczeniu ich przed nieuprawnionym lub nieprawidłowym, przypadkowym bądź umyślnym ujawnieniem, modyfikacją lub zniszczeniem [61][21]. Wyróżniamy trzy podstawowe aspekty bezpieczeństwa danych:

- poufność,
- integralność,
- dostępność.

Poufność oznacza niedostępność treści zawartej w danych dla wszystkich podmiotów nieuprawnionych do jej odczytania. Danym, których naruszenie poufności byłoby szczególnie niewskazane i kosztowne, przypisujemy odpowiednio wysoki poziom bezpieczeństwa (poufne, tajne, ściśle tajne). Bezpośrednim sposobem zapewnienia poufności jest szyfrowanie danych. Procedury, ograniczenia uprawnień dostępu czy ograniczenie fizycznego dostępu do systemu komputerowego są środkami pośrednimi, pozwalającymi na osiągnięcie tego celu. Pomimo stosowania różnego rodzaju środków zapewniających poufność, istnieje niebezpieczeństwo przypadkowego lub celowego jej naruszenia. W związku

z tym system ochrony powinien nie tylko zapewniać poufność, lecz także gwarantować możliwość wykrycia prób i przypadków jej naruszenia [63].

Integralność danych oznacza, że dane nie zostaną w żaden nieuprawniony sposób zmienione, a tym samym ich stan pozostanie zgodny z wymaganym i oczekiwanym stanem właściwym. Integralność danych może być naruszona przez nieupoważnionego użytkownika, błędy i zaniedbania popełnione przez użytkownika upoważnionego, a także w wyniku awarii, zakłócenia w transmisji, błędów w oprogramowaniu, działania wirusów, itp. Nieupoważniona modyfikacja nie musi wiązać się z naruszeniem poufności danych. Podobnie jak poufność, integralność musi być zapewniona podczas przetwarzania, przechowywania i przesyłania informacji. Integralność danych zapewnia się stosując funkcje skrótu, a w pewnym stopniu także kody wykrywające i korygujące błędy [63][51][21]. W sposób pośredni można przyczynić się do osiągnięcia tego celu poprzez stosowanie procedur uwierzytelnienia, ograniczanie uprawnień dostępu, ograniczanie fizycznego dostępu do systemu komputerowego, stosowanie metod zwiększających niezawodność sprzętu i tolerancję na błędy. Konieczne może być także zagwarantowanie możliwości wykrycia każdego przypadku lub próby naruszenia integralności danych.

Dostępność oznacza niczym nieograniczoną możliwość korzystania z danych przez uprawnionych do tego użytkowników. Dostępność danych może być naruszona przez nieuprawnionego użytkownika, błędy popełnione przez użytkownika upoważnionego, a także w wyniku awarii, zakłóceń w transmisji, błędu oprogramowania, przeciążenia systemu. Wstrzymanie przez nieupoważnionego użytkownika dostępu do zasobów może stanowić wstęp do ataku na poufność i integralność danych. Pożądane może być więc zapewnienie możliwości wykrycia każdego przypadku nieuzasadnionej odmowy dostępu do danych. Dostępność

zapewnia się przez stosowanie odpowiednio zabezpieczonych systemów operacyjnych, stały nadzór nad stopniem wykorzystania zasobów, stosowanie systemów sterowania ruchem sieciowym i obciążeniem serwerów, stosowanie metod zwiększających niezawodność sprzętu i tolerancję na błędy [78][21][79][69][81].

Spójność danych odnosi się przede wszystkim do baz danych i oznacza konieczność spełniania przez każdy stan bazy danych zbioru warunków sformułowanych w definicji bazy danych. Warunki te, zwane warunkami spójności, dzielą się na warunki statyczne[2] i dynamiczne[3]. Zaistnienie warunków spójności jest warunkiem koniecznym poprawności (niesprzeczności) bazy danych. Naruszenie spójności bazy danych może nastąpić na skutek semantycznie niepoprawnych operacji, niewłaściwej synchronizacji działania transakcji współbieżnych, a także w wyniku awarii systemu. Jednym z zasadniczych celów systemów zarządzania baz danych jest więc czuwanie nad spójnością bazy danych, a środkiem osiągnięcia tego celu jest właściwe zarządzanie transakcjami (w tym współbieżnymi) oraz odtwarzanie bazy danych w przypadku awarii.

Istnieje związek między integralnością i spójnością danych, ale nie są to pojęcia tożsame. Naruszenie integralności, a więc nieuprawniona modyfikacja danych, nie musi naruszyć spójności bazy danych. Jednocześnie jednak użytkownik, który ma uprawnienia do modyfikacji danych, a więc działania jego nie naruszają postulatu integralności, może naruszyć spójność bazy danych i tym samym spowodować odrzucenie tych modyfikacji przez system zarządzania bazą danych [14][18][26][22].

---

[2] określają zależności między danymi w każdym poszczególnym stanie bazy danych.

[3] definiują zależności między danymi z różnych stanów bazy danych, a więc tym samym określają reguły zmiany stanów.

2.2. Zagrożenia

Wśród zagrożeń wyróżnić można dwie grupy. Po pierwsze, zagrożenia wynikające z celowego działania nieuprawnionego użytkownika. Po drugie, takie, które nie są skutkiem celowego działania. Do tej drugiej grupy zaliczamy: awarie sprzętu, zaniki zasilania, błędy użytkowników, skutki działania czynników zewnętrznych, takich jak ogień, woda, trzęsienia ziemi, zewnętrzne pole elektromagnetyczne. Każde z zagrożeń może spowodować naruszenie jednego lub kilku aspektów bezpieczeństwa danych.

Największą grupę problemów związanych z bezpieczeństwem systemów stanowią problemy stwarzane przez ludzi. To ludzie włamują się do systemów, podsłuchują, niszczą dane, wprowadzają wirusy, zaniedbują swoje obowiązki lub w sposób nieświadomy przyczyniają się do obniżenia poziomu bezpieczeństwa. Przestępstwa popełniane są najczęściej przez osoby posiadające pewne uprawnienia w systemie komputerowym lub osoby, które posiadały takie uprawnienia w przeszłości. Zagrożenia osobowe można podzielić na dwie kategorie: zewnętrzne i wewnętrzne. Do zewnętrznych zalicza się: agentów obcych wywiadów, terrorystów, kryminalistów, agentów wywiadu gospodarczego i hakerów – inaczej mówiąc atakujący jest osobą spoza kręgu użytkowników systemu komputerowego [130][134][138]. Kategoria zagrożeń wewnętrznych obejmuje: pracowników, dostawców, konsultantów, klientów, byłych pracowników, stażystów – osoby, które w jakiś sposób są powiązane z systemem informatycznym. Większość incydentów związanych z bezpieczeństwem ma swoje źródło w wewnętrznej strukturze organizacji. Wśród zagrożeń wewnętrznych wyróżnić możemy:

- akty wewnętrznego sabotażu,
- kradzież informacji,

- kradzież usług – np. wykorzystanie zasobów do celów prywatnych,
- błędy użytkowników,
- niedbalstwo,
- nieprawidłowe stosowanie mechanizmów bezpieczeństwa.

Często występującym zagrożeniem są błędy popełniane przez użytkowników: błędy we wprowadzanych danych, błędy w użytkowaniu systemu, przypadkowe usunięcia plików. Do przykładów niedbalstwa użytkowników i administratorów systemów zaliczyć można brak odpowiedniej ochrony antywirusowej i brak lub niepełność zapasowych kopii plików. Najbardziej jaskrawym przykładem niewłaściwego stosowania mechanizmów bezpieczeństwa przez użytkowników są nieprawidłowości w korzystaniu z systemów haseł. Często zdarza się, że użytkownicy ujawniają swoje hasła innym osobom, stosując identyczne hasła do różnych systemów i aplikacji, posługują się hasłami krótkimi, łatwymi do odgadnięcia, zapisują hasła w miejscach ogólnie dostępnych [24][42][47][60].

Włamanie się do systemu przez użytkownika nieuprawnionego odbywa się najczęściej przez przechwycenie identyfikatora i hasła użytkownika uprawnionego. Stosowanych w tym celu jest wiele metod:

- zmiana oryginalnego programu, którym użytkownicy rejestrują się w systemie – podstawienie konia trojańskiego,
- wykorzystanie rezydentnych programów kontrolujących klawiaturę – zapisywanie sekwencji naciskanych klawiszy,
- podsłuch łącza, którym transmitowane są dane uwierzytelniające,
    - rozgałęźnik na kablu klawiatury i urządzenie rejestrujące transmisję z klawiatury,

    - podsłuch w lokalnej sieci komputerowej,
    - podsłuch w sieci rozległej,
- atak słownikowy,
- przeszukiwanie wyczerpującą metodą prób i błędów – atak brutalny,
- metody fizyczne – patrzenie na ręce użytkownika rejestrującego się w systemie,
- metody inżynierii społecznej – nakłonienie użytkownika do udostępnienia lub zmiany hasła.

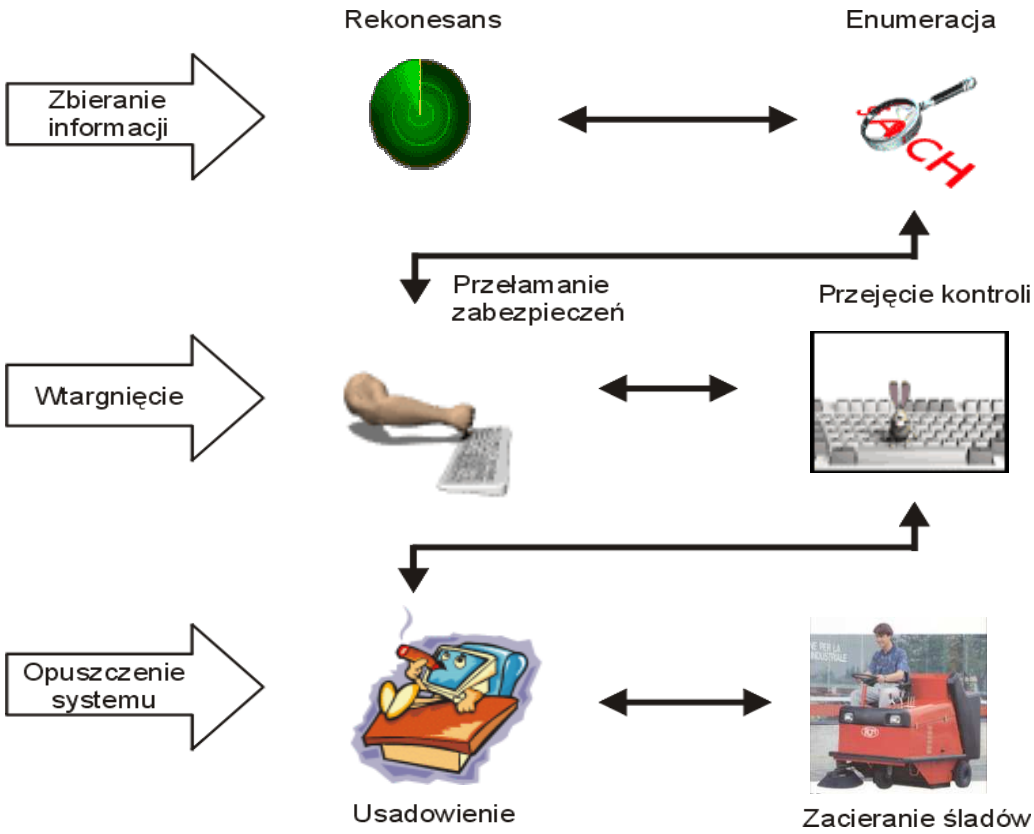
Atak słownikowy polega na podstawianiu w miejsce hasła ciągów znaków sekwencyjnie pobieranych z pewnego słownika [51][91]. Obroną przed atakiem słownikowym jest stosowana często w systemach operacyjnych funkcja automatycznego blokowania konta po kilkukrotnym podaniu błędnego hasła. Jednak ta sama funkcja może być wykorzystana przez intruza w celu pozbawienia dostępu uprawnionego użytkownika. Jeżeli plik z hasłami zostanie przejęty przez potencjalnego włamywacza, to atak słownikowy może być przeprowadzony poza systemem, w którym dane hasła są stosowane.

Przeszukiwanie wyczerpujące jest zbliżone do ataku słownikowego. Jednak w tym przypadku w miejsce szukanego hasła podstawia się nie ciągi znaków z ograniczonego zbioru, lecz wszystkie możliwe ciągi. Ze względu na zwykle dużą liczbę kombinacji metoda ta wymaga zastosowania znacznych mocy obliczeniowych.

Ponadto włamania można dokonać wykorzystując:

- programy zmieniające hasła,
- wyłączenie kontroli haseł przy rejestrowaniu,
- chwilową nieobecność użytkownika,

- tzw. haki pielęgnacyjne, czyli niejawne instrukcje umożliwiające łatwą obsługę i rozwój oprogramowania,
- błędy w systemach operacyjnych i oprogramowaniu użytkowym.



Rysunek 2 Przebieg przeprowadzenia ataku

Atak na system informatyczny poprzedzany jest zwykle etapem wstępnym (Rysunek 2) – zbieraniem informacji o atakowanym systemie [100]. Istnieje wiele sposobów zdobywania informacji ułatwiających wykonanie ataku:

- zbieranie dokumentów publicznie dostępnych (niekiedy publicznie udostępniane są informacje o używanych systemach operacyjnych i środkach ochrony, stosowanym sprzęcie komputerowym),

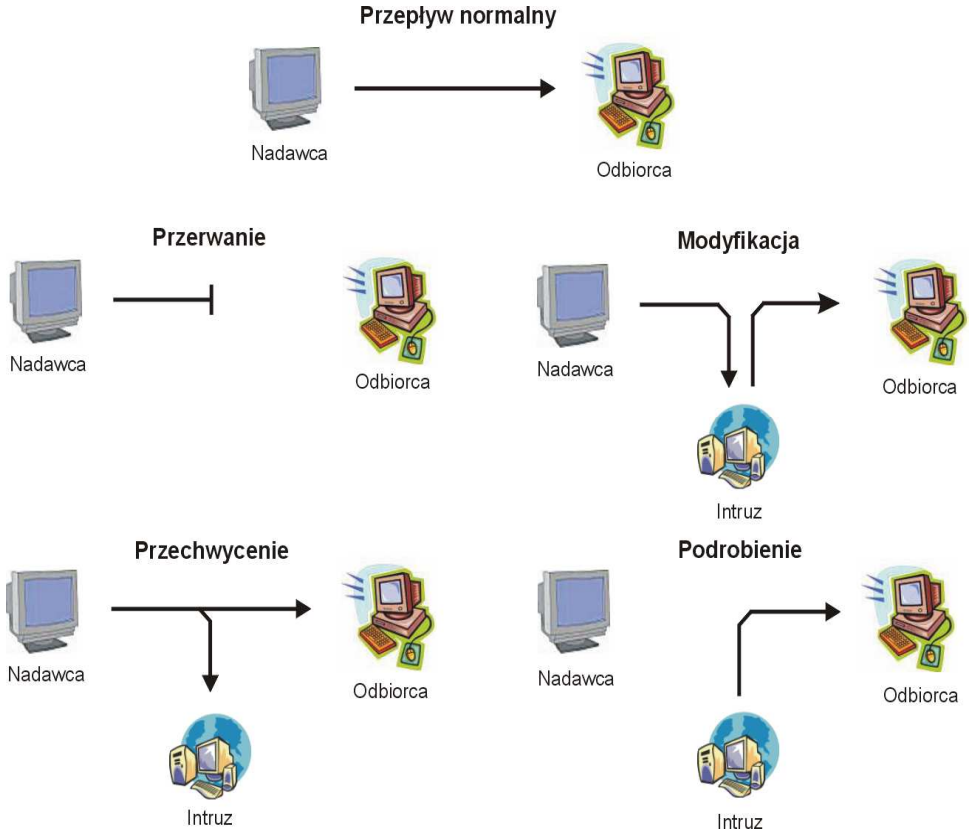
- podstęp – udawanie pracownika, dostawcy (osobiście, w rozmowie telefonicznej, w korespondencji), w celu uzyskania informacji, o które się zabiega,
- wymuszenia, szantaż,
- przeszukiwanie odpadków w siedzibie instytucji i poza nią (często wśród wyrzucanych odpadków i makulatury znaleźć można notatki, informacje o klientach, telefonach wewnętrznych, dokumentację techniczną, zapisane nośniki danych).

Do ujawnienia poufnych informacji może dojść także, jeżeli w systemie istnieją tzw. ukryte kanały. Są to kanały komunikacyjne pozwalające przesłać informację pomiędzy użytkownikami z pominięciem kontroli dostępu i innych mechanizmów zabezpieczających. Ukryty kanał powstaje, jeżeli użytkownik uprawniony (nadawca) i użytkownik nieuprawniony (odbiorca) współdzielą pewne zasoby. Użytkownik uprawniony lub działający w jego imieniu program może manipulować zasobem, a odbiorca informacji może odczytywać stan zasobu, na przykład nazwy plików.

Innym problemem związanym z poufnością danych jest niebezpieczeństwo wnioskowania. Do naruszenia poufności informacji może dojść, jeżeli na podstawie dostępnych danych nieuprawniony użytkownik może wyciągać wnioski co do danych poufnych. Podczas wnioskowania o danych poufnych można wykorzystać następujące techniki:

- pośredni dostęp do danych – formułowanie zapytań do udostępnionej części danych ujawniających informacje o danych tajnych,
- analiza odpowiedzi systemu na próby wprowadzania nowych danych,

- wykorzystywanie arytmetycznych i logicznych zależności między danymi jawnymi i tajnymi,
- analiza odpowiedzi, w których część danych nie jest ujawniana,
- korzystanie z funkcji statystycznych.



Rysunek 3 Ataki na bezpieczeństwo przepływu informacji

Innym przykładem zagrożeń mogą być wirusy komputerowe. Do najczęstszych należą programy zakłócające normalne działanie systemów komputerowych. Wywoływane zakłócenia mogą polegać na: niszczeniu lub zmianie zawartości plików i dysków, blokowaniu systemu, obciążaniu procesora dodatkową pracą, generowaniu dźwięku, wyświetlaniu nieoczekiwanych tekstów lub obrazów. Niektóre z tego typu programów są wykorzystywane do ataków na systemy informatyczne. Ich autorzy

korzystają z coraz bogatszego zbioru mechanizmów tworzenia i rozpowszechniania, między innymi z języka Java, aplikacji ActiveX, języków makro poleceń [24][[84][. Istnieje kilka grup takich programów. Są to:

- wirusy,
- bakterie,
- robaki,
- bomby logiczne,
- konie trojańskie.

Największą liczebnie grupę stanowią wirusy komputerowe. Ocenia się, iż dotąd pojawiło się ponad 40 000 wirusów komputerowych, przy czym szacuje się, że liczba ta zwiększa się o setki wirusów miesięcznie. Termin „wirus komputerowy" oznacza program, który potrafi się rozmnażać i dopisywać oraz ukrywać wewnątrz plików zawierających kod wykonywalny (wirusy plików) lub wewnątrz systemowych sektorów na dyskach (wirusy sektorów systemowych). Wirusy plików uaktywniają się w momencie uruchomienia zainfekowanego programu – następuje zarażenie innego pliku z kodem wykonywalnym i/lub wywołanie zakłóceń w działaniu systemu. Wirus doczepiony do pliku w komputerze przedostaje się wraz z plikiem do innego komputera (przy przenoszeniu programu na nośniku danych lub w czasie transmisji siecią komputerową).

Rozwój systemów antywirusowych spowodował pojawienie się nowych, trudniejszych do wykrycia typów wirusów. Do tej grupy należą między innymi wirusy typu stealth i wirusy polimorficzne. Wirusy typu stealth potrafią ukryć przed programem antywirusowym zmianę pliku, który został

zarażony. Wirusy polimorficzne utrudniają wykrycie przez mutacje, czyli modyfikacje swojej własnej zawartości przy każdej kolejnej infekcji.

Oddzielną grupę stanowią makrowirusy tworzone za pomocą języków makro poleceń, dostępnych między innymi w edytorach tekstu i arkuszach kalkulacyjnych. Makrowirusy rozprzestrzeniają się wraz z dokumentami (na przykład tekstami napisanymi za pomocą edytora MS Word), do których są doczepione [22]. Ponieważ transmisja dokumentów jest prowadzona na znacznie szerszą skalę niż transmisja programów, makrowirusy rozprzestrzeniają się znacznie szybciej aniżeli wirusy tradycyjne. Ponadto tworzenie i modyfikowanie makro wirusów jest prostsze.

Bakterie są samodzielnymi programami powielającymi się, których niszczycielskie działanie polega na zużywaniu zasobów systemu, takich jak moc procesora, pamięć operacyjna i pamięć dyskowa. Duża szybkość rozmnażania się bakterii powoduje bardzo szybkie wyczerpanie zasobów i blokadę systemu.

Działanie robaków jest podobne do bakterii. Jednak w odróżnieniu od bakterii ich polem działania nie jest pojedynczy system, lecz sieć komputerowa. Robak rozprzestrzenia się w sieci, zmniejsza wydajność serwerów lub je całkowicie blokuje.

Programowa bomba logiczna jest ukrytym i nieudokumentowanym fragmentem innego programu, zaprojektowanym w taki sposób, że w określonych warunkach następuje aktywacja bomby i realizacja zaprogramowanych czynności. Warunkiem aktywacji może być: określona data, liczba uruchomień programu nosiciela, umieszczenie lub usunięcie pewnej informacji w systemie (pliku, rekordu, wartości w bazie danych).

Terminem koń trojański oznacza się realizujący pożyteczne zadanie program, w którego wnętrzu celowo umieszczono i ukryto nieudokumentowany fragment kodu wykonujący niepożądaną i szkodliwą

czynność. Konie trojańskie są często stosowane przez niepowołanych użytkowników do wykonywania pewnych czynności w systemie, do którego nie mają oni bezpośredniego dostępu, w celu ukrycia aktywności hakera w systemie, zainfekowania systemu wirusem komputerowym.

Urządzenia elektroniczne są źródłami emitowanego na zewnątrz promieniowania elektromagnetycznego. Promieniowanie to może być rejestrowane i analizowane przez specjalne urządzenia. Zjawisko niepożądanej emisji sygnałów elektromagnetycznych nazywane jest emisją ujawniającą, a występowanie sygnału emisji ujawniającej – elektromagnetycznym przenikaniem informacji. Wyróżnia się emisję ujawniającą promieniowaną i przewodzoną. Emisja ujawniająca promieniowana powstaje przez indukowanie pola elektromagnetycznego w przestrzeni otaczającej urządzenie. Emisja ujawniająca przewodzona polega na przekazywaniu sygnałów elektrycznych w sieci zasilającej, w uziemieniach i w obwodach sygnałowych.

2.3. Metody ochrony danych

Za przestrzeganie zasad ochrony w systemie informatycznym odpowiedzialny jest system kontroli dostępu. Elementami takiego systemu są: podmioty – użytkownicy, procesy, obiekty – dane, programy, operacje – czytanie, zapisywanie, tworzenie, usuwanie itp.

Polityka bezpieczeństwa określa bazę kontroli dostępu, między innymi sposób zarządzania regułami dostępu i sposób formułowania tych reguł. Ponieważ część naruszeń systemu bezpieczeństwa jest skutkiem zaniedbań i nieświadomych działań uprawnionych użytkowników, w polityce bezpieczeństwa należy uwzględnić konieczność sprawdzania wiedzy użytkowników, osobistą odpowiedzialność za zaniedbania, systematyczne szkolenia [19][34][41].

Personalną odpowiedzialność za naruszenia zasad bezpieczeństwa zapewnia stosowanie indywidualnych, osobistych uprawnień w systemie – nawet jeżeli pewna grupa użytkowników cechuje się identycznymi przywilejami. Dzięki takiej indywidualizacji uprawnień możliwe jest:

- ustalenie indywidualnej odpowiedzialności,
- nadzorowanie aktywności każdego użytkownika osobno,
- uniknięcie konieczności zmiany hasła grupy w przypadku zmiany uprawnień jednego z jej członków.

Z punktu widzenia polityki bezpieczeństwa wyróżnia się systemy otwarte i zamknięte. W systemie otwartym reguły dostępu formułowane są w postaci zakazów (to, co nie jest zabronione, jest dozwolone). Podmiot nie otrzymuje dostępu do obiektu tylko wtedy, gdy do reguł dostępu należy odpowiedni zakaz.

W systemie zamkniętym reguły dostępu formułowane są w postaci przywilejów (to, co nie jest dozwolone, jest zabronione). Podmiot nie otrzymuje dostępu do obiektu wtedy, gdy do reguł dostępu nie należy odpowiedni przywilej. System zamknięty pozwala spełnić jedną z podstawowych zasad formułowania reguł dostępu – zasadę minimum koniecznego [46][47][51]. Zgodnie z tą zasadą podmiotowi przyznaje się najbardziej ograniczający zestaw uprawnień, niezbędnych do realizacji obowiązków. Podobną zasadą jest udostępnianie uprawnień jedynie podczas okresów, które są wymagane do realizacji obowiązków. Ewentualne naruszenia systemu bezpieczeństwa wynikające z przypadkowego lub celowego działania są w takim wypadku ograniczone. Systemy zamknięte charakteryzują się wyższym poziomem bezpieczeństwa i w praktyce są częściej stosowane od systemów otwartych.

Ważnym aspektem polityki bezpieczeństwa jest sposób zarządzania regułami dostępu [51]. W zależności od liczby uprawnionych do zarządzania administratorów wyróżniamy systemy scentralizowane i rozproszone. W przypadku zarządzania scentralizowanego pojedynczy, uprawniony administrator definiuje reguły dostępu. Wśród rozwiązań systemów rozproszonych wyróżniamy:

- systemy hierarchiczne – centralny administrator nadaje uprawnienia administratorom niższego szczebla,
- systemy oparte o posiadanie obiektu – podmiot tworzący obiekt staje się jego administratorem,
- systemy kooperacyjne – reguły dostępu są definiowane za zgodą grupy administratorów.

W celu uproszczenia zarządzania w procesie realizacji polityki bezpieczeństwa grupuje się podmioty o tych samych uprawnieniach i obiekty o tych samych klasach bezpieczeństwa. Do rozwiązania w tym wypadku są problemy przynależności jednego podmiotu do wielu grup jednocześnie, problemy związane ze zmianą przynależności. Grupowanie odbywa się często z wykorzystaniem hierarchicznej klasyfikacji podmiotów i obiektów. Tego typu klasyfikacje są stosowane powszechnie w tworzonych do celów militarnych wielopoziomowych systemach bezpieczeństwa. Wyróżnia się dwa sposoby realizacji kontroli dostępu: obowiązkową i uznaniową [51][61][64][67][103].

Obowiązkowa kontrola dostępu obejmuje środki ograniczania dostępu do obiektów, oparte o etykiety bezpieczeństwa i scentralizowane zarządzanie regułami dostępu. Każdemu podmiotowi i obiektowi systemu nadaje się klasę bezpieczeństwa wyznaczoną na podstawie stopnia ochrony (np. jawne, poufne, tajne, ściśle tajne) i kategorii (obszaru zastosowań).

Klasyfikacja podmiotów odzwierciedla stopień zaufania dla podmiotu i obszar działania podmiotu. Do klasyfikowania obiektów wykorzystuje się ich wrażliwość, czyli miarę ważności przypisaną do informacji zawartych w obiekcie; innymi słowy – wielkość potencjalnych szkód spowodowanych ujawnieniem informacji. Podmiot otrzymuje dostęp do obiektu, jeżeli spełnione są odpowiednie relacje pomiędzy klasą bezpieczeństwa podmiotu a klasą bezpieczeństwa obiektu.

Uznaniowa kontrola dostępu to środki ograniczania dostępu do obiektów, oparte o identyfikację użytkowników, przywileje i rozproszone zarządzanie regułami dostępu. Każdemu podmiotowi nadaje się przywileje (czytania, modyfikacji, usuwania itp.) w stosunku do określonych obiektów. W systemie takim podmiot może decydować o przywilejach innych podmiotów w stosunku do określonych obiektów. Podmiot otrzymuje dostęp do obiektu w wybranym trybie (odczyt, zapis, usunięcie itp.), jeżeli posiada odpowiedni przywilej w stosunku do tego obiektu [103][104].

Przywileje podmiotów wobec obiektów mogą istnieć w postaci zorientowanej na bilety lub w postaci zorientowanej na listy. W pierwszym przypadku każdy podmiot przechowuje niefałszowalną listę wzorców bitowych, nazywanych „biletami", po jednym dla każdego obiektu, do dostępu, do którego jest uprawniony. W drugim – każdy chroniony obiekt ma listę wszystkich podmiotów uprawnionych do dostępu do tego obiektu.

Stosowane w systemie reguły dostępu są niekiedy rozszerzane o dodatkowe predykaty wyrażające ograniczenia nałożone na przywileje podmiotu. Ograniczenia takie mogą być związane z wartościami obiektów, datą i czasem dostępu, sposobem uzyskiwania dostępu (zdalnie czy lokalnie), historią operacji realizowanych wcześniej. W systemie z dodatkowymi ograniczeniami podmiot otrzymuje dostęp do obiektu w wybranym trybie, jeżeli posiada odpowiedni przywilej w stosunku do tego

obiektu i spełnione są wszystkie predykaty związane z danym dostępem. Kontrolowaniem predykatów zajmują się systemy operacyjne i systemy zarządzania bazami danych.

Za wdrażanie polityki bezpieczeństwa odpowiedzialni są ludzie operujący na odpowiednich procedurach opracowywych w tym celu przy pomocy mechanizmów zabezpieczenia, które realizują określone zabezpieczenie za pomocą sprzętu i oprogramowania. Mechanizmy te spełniają role prewencyjne i detekcyjne. Możemy wśród nich wyodrębnić mechanizmy zewnętrzne i wewnętrzne. Do pierwszej grupy należą administracyjne i materialne środki zabezpieczeń przed dostępem do pomieszczeń, urządzeń oraz ochrona przed awariami, katastrofami. Celami są: minimalizacja możliwych naruszeń systemu ochrony, minimalizacja konsekwencji wynikających z takich naruszeń i zagwarantowanie możliwości odtworzenia stanu po naruszeniu systemu ochrony. Do wewnętrznych mechanizmów zabezpieczenia należą:

- mechanizmy identyfikacji i uwierzytelniania,
- mechanizmy kontroli dostępu – systemy uprawnień,
- mechanizmy audytu.

Wśród metod uwierzytelniania, czyli sprawdzania tożsamości użytkownika systemu komputerowego, wyróżnia się trzy grupy:

- metody oparte o wiedzę użytkownika,
- metody oparte o identyfikatory materialne,
- metody biometryczne.

Użytkownik pragnący uzyskać dostęp do chronionego systemu komputerowego musi wykazać się pewną wiedzą lub posiadaniem obiektu

identyfikującego albo posiadaniem pewnych cech fizycznych. Ponieważ każda z powyższych metod obarczona jest pewnymi wadami, wysoki stopień bezpieczeństwa osiąga się stosując jednocześnie więcej niż jedną z metod uwierzytelniania [120][130][47].

Uwierzytelniony w systemie użytkownik nie powinien pozostawiać dostępnych mu zasobów systemu bez nadzoru. Przed każdym opuszczeniem miejsca pracy niezbędne jest dokonanie czynności wymuszających konieczność ponownego uwierzytelnienia przed kontynuacją pracy w systemie.

Do metod opartych o wiedzę użytkownika zaliczymy najbardziej rozpowszechnioną metodę, w której proces uwierzytelnienia polega na sprawdzeniu znajomości poufnego hasła. Odmianą tej metody jest sprawdzenie znajomości pewnych faktów. Podstawową zaletą systemu opartego na hasłach użytkowników jest prostota implementacji.

Zapewnienie odpowiedniego poziomu bezpieczeństwa wymaga rygorystycznego przestrzegania zasad bezpiecznego stosowania haseł. Do najważniejszych czynników wpływających na bezpieczeństwo systemu hasłowego zaliczymy:

- moc alfabetu, w którym tworzone są hasła,
- długość haseł,
- okres ważności,
- sposób generacji,
- złożoność haseł,
- metody przechowywania i przesyłania haseł w systemie.

Poziom bezpieczeństwa hasła zależy od liczby znaków w alfabecie, w którym tworzone jest hasło. Od liczności alfabetu i długości haseł zależy

liczba wszystkich możliwych haseł, a jednocześnie złożoność ataku poprzez przeszukiwanie wyczerpujące. Liczba haseł o długości *l*, które można zbudować z alfabetu zawierającego *n* znaków jest równa n^l [45].

Ponieważ zwiększanie długości haseł wiąże się ze zwiększeniem prawdopodobieństwa popełnienia błędu przy wprowadzaniu i zwiększeniem trudności zapamiętania, zamiast pojedynczego ciągu znaków stosuje się niekiedy frazy hasłowe – zrozumiałe sekwencje wyrazów o długościach przekraczających maksymalną długość hasła. Fraza taka jest przekształcana przez funkcję skrótu do rzeczywistego hasła o akceptowalnej długości. Funkcją skrótu nazywamy taką funkcję jednokierunkową *h* przekształcającą wiadomość *m* o dowolnej długości w r-bitowy skrót *h(m)*.

Skutki ujawnienia hasła można zmniejszyć skracając okres jego ważności. Po upływie okresu ważności lub w przypadku podejrzenia ujawnienia hasło powinno być zmienione. Najbardziej efektywną metodą minimalizowania niebezpieczeństwa jest stosowanie systemu haseł jednorazowych.

Hasło może być tworzone przez użytkownika, który będzie się nim posługiwał, przez ręczny generator haseł jednorazowych lub automatycznie, losowo przez system. Jeżeli użytkownik ma możliwość tworzenia hasła, to system (administrator systemu) powinien korzystać z mechanizmów zabezpieczających przed stosowaniem haseł łatwych do złamania przy użyciu ataku słownikowego.

Hasła generowane automatycznie przez system muszą posiadać cechy ułatwiające ich zapamiętanie. Łatwość zapamiętania zmniejsza skłonność do zapisywania hasła. Jednocześnie dobrze jest, gdy nie jest znana metoda generowania haseł w systemie [33][47][61][75].

Metody przesyłania i przechowywania haseł użytkowników powinny zapewniać wysoki poziom ochrony przed ujawnieniem. Do transmisji haseł należy wykorzystywać łącza komunikacyjne zabezpieczone fizycznie przed podsłuchem biernym i aktywnym. W czasie transmisji i przechowywania hasła powinny być zaszyfrowane.

Coraz częściej stosowane są rozwiązania polegające na jednokrotnym uwierzytelnieniu użytkownika. Rozwiązania tego typu pozwalają użytkownikowi na dostęp do wielu rozproszonych zasobów po jednokrotnym uwierzytelnieniu się wobec centrum uwierzytelniania – zaufanej trzeciej strony. Zadaniem centrum uwierzytelniania jest przechowywanie i zarządzanie hasłami użytkowników. Użytkownik, który chce się uwierzytelnić w celu otrzymania dostępu do pewnego zasobu, przesyła swój identyfikator i hasło do centrum uwierzytelniania. Po uwierzytelnieniu trzecia strona udostępnia użytkownikowi tzw. bilety pozwalające skorzystać z chronionych zasobów. Bilety są zaszyfrowane za pomocą kluczy kryptograficznych poszczególnych, chronionych zasobów. Do tego typu rozwiązań należą: Kerberos i Passport [74][75][[92].

Passport jest protokołem pozwalającym użytkownikowi na dostęp do wielu różnych stron WWW po jednorazowym uwierzytelnieniu się wobec serwera uwierzytelniania ustalonego dla grupy stron WWW (np. sklepów internetowych). Bilet uprawniający do dostępu jest przechowywany na dysku użytkownika w postaci tzw. ciasteczka i przesyłany do serwera wraz z żądaniem HTTP.

W bardzo wielu mechanizmach bezpieczeństwa wykorzystuje się różne formy haseł. Kwestia poprawnego stosowania haseł jest jedną z ważniejszych – bardzo duża liczba problemów związanych z bezpieczeństwem (można szacować na ponad 50%) wynika ze stosowania nieodpowiednio wybranych lub źle chronionych haseł.

Przestrzeganie niektórych zasad poprawności stosowania haseł można wymusić, korzystając z narzędzi do kontroli złożoności i wymiany haseł. Złożoność haseł można zagwarantować stosując programy mające wbudowane słowniki wyrazów, nazw użytkowników, popularnych wyrażeń. Za pomocą pewnych reguł (na przykład zapisywania wyrazów od tyłu) programy takie generują zbiory niedozwolonych haseł. Mogą one działać w sposób aktywny lub pasywny. Aktywne sprawdzanie haseł polega na okresowym uruchamianiu programu i wyszukiwaniu haseł łatwych do odgadnięcia. Pasywne sprawdzanie haseł jest realizowane w momencie wyboru nowego hasła przez użytkownika. Wybranie hasła łatwego do odgadnięcia powoduje odrzucenie go przez program. Wielokrotne używanie tego samego hasła można uniemożliwić tworząc rejestr historii haseł.

W niektórych systemach i aplikacjach producenci instalują domyślne hasło fabryczne – często identyczne dla partii produktów. Użytkownik systemu z hasłem fabrycznym powinien je zmienić i posługiwać się wyłącznie własnym, prywatnym hasłem.

Użytkownik może udowodnić swoją tożsamość, okazując identyfikator fizyczny, najczęściej w postaci karty. Proces uwierzytelnienia polega na odczytaniu informacji zawartej w identyfikatorze. W bardziej zaawansowanych systemach mikroprocesor wbudowany w tzw. identyfikator inteligentny dodatkowo przetwarza informacje.

Komunikacja między identyfikatorem a systemem jest realizowana za pomocą czytnika kontaktowego lub bezkontaktowego. Czytniki kontaktowe wykorzystują do komunikacji styki elektryczne, odczyt magnetyczny lub optyczny. W czytnikach bezkontaktowych transmisja odbywa się przy użyciu sprzężeń indukcyjnych, pojemnościowych, fal podczerwonych lub radiowych.

Największym zagrożeniem dla systemów, w których identyfikacja odbywa się przy użyciu identyfikatorów fizycznych, jest możliwość kradzieży, zagubienia i podrobienia. Poziom bezpieczeństwa można zwiększyć poprzez zintegrowanie systemu identyfikatorów z systemem haseł. Tego typu rozwiązanie wykorzystywane jest w systemach uwierzytelniania z żetonami. Żeton jest identyfikatorem generującym jednorazowe hasła. Korzystanie z żetonu polega na uaktywnieniu przez podanie osobistego numeru identyfikacyjnego (PIN) użytkownika, a następnie na wprowadzeniu odpowiedniego ciągu cyfr (na przykład otrzymanego od systemu informatycznego, wobec którego użytkownik chce się uwierzytelnić) – na wyświetlaczu żetonu pojawia się hasło, którym użytkownik posługuje się w celu uwierzytelnienia. Funkcje żetonów mogą być realizowane także przez oprogramowanie instalowane na komputerze użytkownika.

Metody biometryczne wykorzystują fakt unikatowości pewnych fizycznych cech człowieka i jego zachowań [14][26]. Do wykorzystywanych w procesach uwierzytelniania cech należą:

- linie papilarne,
- kształt twarzy, dłoni,
- rysunek tęczówki oka,
- głos,
- podpis odręczny,
- sposób pisania na klawiaturze.

Uwierzytelnianie za pomocą biometrii jest procesem kilkuetapowym. Rozpoczyna się od pomiaru cech użytkownika. Czujnik pomiarowy (często uzupełniony o przetwornik analogowo-cyfrowy) przekazuje do systemu

sygnały cyfrowe reprezentujące wynik pomiaru. Sygnały te są następnie przetwarzane do odpowiedniego formatu. Następną fazą jest weryfikacja, czyli porównanie wyniku pomiaru z zapamiętanymi wzorcami. Wynik może być porównany z wszystkimi zapamiętanymi wzorcami lub wzorcem jednego użytkownika. W tym drugim przypadku konieczne jest dodatkowe zidentyfikowanie użytkownika, na przykład przez wpisanie nazwiska lub identyfikatora. Ze względu na częstą niedokładność wynik pomiaru uprawnionego użytkownika rzadko jest identyczny z zapamiętanym wzorcem tego użytkownika. Konieczne jest stosowanie dobrze wybranego zakresu tolerancji dopasowania między wzorcem a bieżącym pomiarem. Zbyt mały zakres tolerancji będzie powodował wysoki wskaźnik błędnych odrzuceń; duży zakres tolerancji może przyczynić się do akceptowania nieautoryzowanych użytkowników.

Wzorce cech użytkowników oraz wyniki pomiarów powinny być chronione podczas przesyłania i przechowywania w sposób podobny jak hasła użytkowników.

Jeden z wewnętrznych mechanizmów bezpieczeństwa stanowią mechanizmy audytu bezpieczeństwa. Ich zadanie polega na bieżącym monitorowaniu aktywności użytkowników i rejestrowaniu zdarzeń. Wystąpienie rejestrowanego zdarzenia powoduje umieszczenie zapisu audytorskiego w rejestrze kontrolnym. Zapis audytorski zawiera informację o podmiotach i obiektach uczestniczących w zdarzeniu. Zdarzeniami zapisywanymi w rejestrze kontrolnym mogą być, na przykład: zarejestrowanie się użytkownika, odczyt i zapis pliku, zmiana parametru systemu bezpieczeństwa (np. zmiana hasła użytkownika). Informacje z rejestru kontrolnego są wykorzystywane do śledzenia wypadków

(incydentów) związanych z zabezpieczeniem lub do rekonstrukcji danych, które zostały uszkodzone lub zniszczone.

W celu wykrycia prób naruszenia bezpieczeństwa systemu i w celu wprowadzenia w błąd ewentualnego intruza, korzysta się z narzędzi do tworzenia tzw. szczelin pozornych i przynęt, czyli łatwo wykrywalnych z zewnątrz i dobrze monitorowanych luk, stanowiących pułapki na nieuprawnionych użytkowników. Przynęta może służyć także do zbierania informacji o metodach działań intruzów. Informacje te są następnie wykorzystywane do udoskonalania systemów ochrony. Istnieje kilka metod tworzenia systemów przynęt:

- komputer ze standardowym systemem operacyjnym podłączony do Internetu w celu przyciągnięcia intruzów,
- oprogramowanie emulujące funkcje różnych rzeczywistych systemów,
- system tworzący złudzenie istnienia w systemie informatycznym wielu luk bezpieczeństwa.

Dwa podstawowe wymagania wobec takiego systemu to wbudowana możliwość śledzenia wszystkich działań intruzów oraz gwarancja nieujawnienia rzeczywistego celu, w jakim system został stworzony.

Coraz częściej standardowe mechanizmy audytu w systemach operacyjnych są uzupełniane przez specjalizowane programy śledzące aktywność użytkowników. Programy takie spełniają wiele funkcji:

- rejestrują informacje o stronach WWW odwiedzanych przez użytkowników i poświęcanym na to czasie,
- dokonują analizy treści poczty elektronicznej wysyłanej i odbieranej przez użytkowników,

- blokują dostęp do wybranych miejsc w Internecie,
- rejestrują różne formy aktywności użytkowników (korzystanie z gier komputerowych, wykorzystywanie oprogramowania firmy do celów prywatnych).

Kolejnym rodzajem ochrony danych jest system antywirusowy. Najskuteczniejszą metodą walki z wirusami komputerowymi jest odpowiednia profilaktyka – niedopuszczanie, by wirusy dostały się do systemu. Mimo iż zapewnienie pełnego bezpieczeństwa w ten sposób jest niemożliwe, to pozwala ograniczyć liczbę wirusów dostających się do systemu. Podstawowym elementem profilaktyki jest uruchamianie w systemie komputerowym wyłącznie programów pochodzących z bezpiecznych źródeł i przetestowanych za pomocą programów antywirusowych oraz otwieranie wyłącznie przetestowanych dokumentów. Testowanie powinno obejmować także wszelkie nośniki danych i pliki transmitowane sieciami komputerowymi [54]. Bardzo ważnym elementem ograniczającym skutki niszczycielskiego działania wirusów jest tworzenie i przechowywanie zapasowych kopii plików. Programy antywirusowe wykorzystują wiele metod wykrywania wirusów:

- poszukiwanie sygnatur – zawartość testowanego pliku jest porównywana ze zbiorem sygnatur (charakterystycznych ciągów bajtów) różnych wirusów, w bardziej zaawansowanych programach wykrywanie jest wspomagane przez takie rozwiązania, jak sieci neuronowe czy systemy ekspertowe,
- sprawdzanie integralności – aktualne cechy charakterystyczne (na przykład długość, wartość sumy kontrolnej) testowanego pliku są porównywane z wartościami zapamiętanymi w bazie danych w czasie, gdy plik był „czysty",

- analiza heurystyczna – badanie zachowania się programów i poszukiwanie prób infekowania systemu (na przykład wywołania przerwań systemowych).

Wiele programów antywirusowych łączy w sobie kilka różnych metod działania, w celu zwiększenia efektywności. Programy antywirusowe mogą działać w trybie rezydentnym lub na żądanie. Programy pracujące w trybie rezydentnym są uruchamiane przy starcie systemu i automatycznie testują kopiowane do systemu pliki oraz umieszczane w komputerze dyski. Programy uruchamiane na żądanie kontrolują pamięć operacyjną i aktualnie zapisane na dyskach pliki.

Postępy technologii generowania wirusów i technologii zabezpieczeń antywirusowych zachodzą w zbliżonym tempie. Zarówno wirusy, jak i programy antywirusowe stają się coraz bardziej złożone i wyrafinowane. Systematycznie rośnie liczba wirusów, brak natomiast narzędzia potrafiącego wykryć i unieszkodliwić wszystkie wirusy. Ochrona przed znaczną częścią istniejących wirusów wymaga jednoczesnego korzystania z kilku różnych programów antywirusowych. Ze względu na pojawianie się nowych wirusów konieczna jest stała aktualizacja baz danych wykorzystywanych przez programy antywirusowe.

Przenikanie elektromagnetyczne (emisja ujawniająca) stanowi kolejny problem ochrony danych. W celu ochrony przez ujawnieniem informacji podejmuje się następujące działania:

- instalowanie i wykorzystywanie wyłącznie urządzeń charakteryzujących się obniżonym poziomem emisji ujawniającej (jednym ze standardów tego typu jest amerykański TEMPEST),

- tłumienie emisji przez instalowanie urządzeń (komputerów, drukarek) w specjalnych obudowach lub w pomieszczeniach ekranowanych, stosowanie ekranowanych kabli i łączówek lub światłowodów,
- ograniczanie do minimum długości kabli,
- filtrowanie zasilania,
- właściwe uziemianie urządzeń,
- stosowanie specjalnych zestawów znaków do wyświetlania tekstu,
- maskowanie elektromagnetyczne – wprowadzanie sygnałów zakłócających lub sygnałów o strukturze podobnej do sygnałów emisji ubocznej.

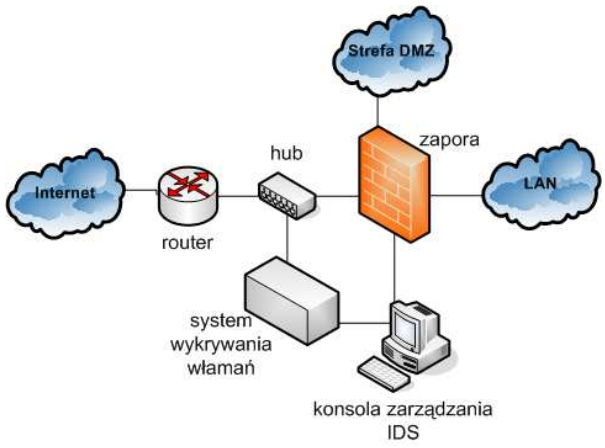
Ważnym elementem ochrony jest badanie rzeczywistej odległości, z której informacja zawarta w sygnale emisji ujawniającej może być odebrana – wyniki takich badań powinny być ściśle chronione [45][63].

Kolejnym elementem ochrony może być śluza bezpieczeństwa, czyli połączenie systemu proxy i firewall – czasami również zawierają w sobie funkcjonalność NAT (ang. *Network Address Translation*) oraz VPN (ang. *Virtual Private Networks*). Śluza bezpieczeństwa jest blokadą chroniącą prywatną, wewnętrzną sieć komputerową przed podsłuchem, penetracją i atakiem z sieci Internet. Obecnie śluzy bezpieczeństwa stają się nieodzownym elementem systemu informatycznego połączonego z Internetem. Są stosowane także do separowania różnych podsieci w ramach jednej wewnętrznej sieci korporacyjnej. Osobiste śluzy bezpieczeństwa chronią komputery prywatne użytkowników korzystających

z Internetu [110][107]. Do podstawowych funkcji śluzy bezpieczeństwa należą:

- przechwytywanie wszystkich informacji przekazywanych między siecią prywatną a Internetem i decydowanie, czy pakiet danych lub połączenie ma być przepuszczone,
- uwierzytelnianie użytkowników,
- utajnianie adresów IP komputerów sieci prywatnej,
- przetwarzanie poczty elektronicznej,
- śledzenie i rejestrowanie zdarzeń,
- tworzenie wirtualnych sieci prywatnych z szyfrowaniem transmisji między węzłami,
- generowanie alarmów.

Funkcję przechwytywania pakietów spełnia firewall. Na podstawie definiowanego przez administratora ciągu reguł filtrowania zatrzymuje lub przepuszcza pakiet. Reguły filtrowania składają się z czynności i kryteriów



Rysunek 4 Schemat systemu bezpieczeństwa z wykorzystaniem ścian ogniowych (firewall), DMZ oraz systemu wykrywania włamań

wyboru. Czynność (blokuj lub przepuść) określa działanie podejmowane w razie spełnienia kryteriów wyboru. Kryteriami wyboru mogą być źródłowe i docelowe adresy IP, typ protokołu, numery portów źródłowych i docelowych, kierunek transmisji (przychodzący czy wychodzący) [112].

Dla każdego pakietu następuje analiza ciągu kolejnych reguł filtrowania. Jeżeli pakiet spełnia kryterium reguły, to jest wykonywana czynność zapisana w regule. Jeżeli nie – następuje sprawdzenie następnej reguły. Ostatnią regułą w ciągu powinno być odrzucanie wszystkich pakietów. Wyróżnia się filtry pasywne i aktywne. Filtry pasywne analizują każdy pakiet w oderwaniu od kontekstu, a filtry aktywne przechowują kontekst sesji, dzięki czemu możliwe jest filtrowanie także pakietów przesyłanych za pomocą protokołów bezpołączeniowych, takich jak UDP.

Uwierzytelnianiem użytkowników po obu stronach śluzy zajmuje się brama aplikacyjna z serwerami proxy (dla każdego typu aplikacji wymagany jest oddzielny serwer). Serwer proxy kontroluje wymianę danych pomiędzy dwiema sieciami, komunikując się w imieniu użytkownika sieci z serwerami na zewnątrz tej sieci.

Rysunek 5 Hosty pośredniczące – brama aplikacyjna

Użytkownik chcący skorzystać z usług Internetu po drugiej stronie śluzy rejestruje się w serwerze proxy. Po uwierzytelnieniu i potwierdzeniu odpowiednich uprawnień otrzymuje dostęp do aplikacji po drugiej stronie śluzy [114][115].

Śluza bezpieczeństwa nie zapewnia pełnej ochrony prywatnej sieci. Bezpieczeństwo sieci prywatnej może być naruszone przez użytkownika wewnętrznego, przez połączenie z Internetem zrealizowane z pominięciem śluzy. Ponadto, w celu ochrony przed wirusami, konieczne jest uzupełnienie śluzy o dodatkowe oprogramowanie antywirusowe.

Śluza bezpieczeństwa zabezpiecza dostęp do danych, jednak nie operacje na tych danych. Do prowadzenia polityki bezpieczeństwa w szerokim tego słowa znaczeniu niezbędne będą jeszcze systemy wykrywania włamań IDS (ang. *Intrusion Detection Systems*). Systemy wykrywania włamań służą do wykrywania prób ataków na systemy informatyczne z zewnątrz, a także prób nadużycia zasobów systemu przez uprawnionych użytkowników. Działają one w tle, w sposób ciągły i informują administratora o wszelkich budzących wątpliwości zdarzeniach. Wiele spośród systemów tego typu działa na zasadzie analizy zapisów audytorskich dokonywanych przez systemy operacyjne. Ponadto systemy takie mogą realizować własne dodatkowe procedury audytu. W zależności od obszaru analizy rejestrów audytorskich wyróżnia się trzy rodzaje systemów wykrywania włamań:

- jednostanowiskowe – analizie poddawane są rejestry audytu z pojedynczego hosta,
- wielostanowiskowe – analizie poddawane są rejestry audytu z wielu hostów,
- sieciowe – analizie poddawane są informacje o ruchu w sieci oraz rejestry audytu z pojedynczego lub z wielu hostów.

Istnieją dwa modele wykrywania włamań: model z wykrywaniem anomalii i model z wykrywaniem nadużyć [44][48][100][107].

W systemach wykorzystujących model z wykrywaniem anomalii włamania są wykrywane poprzez poszukiwanie pojedynczych zdarzeń i ciągów zdarzeń, które odbiegają od normalnego profilu użytkowania systemu. Profil taki wyznaczany jest metodami statystycznymi na podstawie analizy audytu oraz innych informacji, takich jak obciążenie procesora, dysku, pamięci operacyjnej, aktywność użytkowników, liczba otwieranych sesji. Należy dodać, iż profil taki musi być stale aktualizowany.

W systemach korzystających z modelu wykrywania nadużyć porównuje się sekwencje zdarzeń z zapamiętanym zbiorem sygnatur znanych technik ataków. Skuteczność tego typu systemów jest ograniczona do włamań dokonywanych za pomocą metod standardowych.

Szczególne znaczenie posiadają narzędzia do testowania bezpieczeństwa i wykrywania luk w zabezpieczeniach. Narzędzia tego typu tworzą bardzo liczną i różnorodną grupę [69][74]. Wyróżnić tu można:

- proste analizatory wersji oprogramowania i pakietów naprawczych,
- uniwersalne skanery zabezpieczeń,
- skanery portów,
- skanery połączeń modemowych,
- analizatory rejestrów systemowych,
- analizatory list kontroli dostępu,
- analizatory zabezpieczeń baz danych,
- programy do analizy bezpieczeństwa haseł użytkowników,
- skanery bezpieczeństwa dla źródłowych wersji oprogramowania,

- generatory pakietów do testowania zabezpieczeń.

System informatyczny podlega ciągłym zmianom. Na przykład dodawani są nowi użytkownicy, z których wielu korzysta z dostępu zdalnego – istnieje niebezpieczeństwo popełniania błędów przy konfigurowaniu systemu. Przy stosowaniu narzędzi do testowania należy pamiętać, by testy zabezpieczeń były przeprowadzane systematycznie. Nie wystarczy w tym przypadku przeprowadzenie testu po instalacji systemu lub po większej rekonfiguracji. Z drugiej strony częste wykonywanie testów zabezpieczeń (na przykład raz dziennie) jest trudne.

Baza danych skanera zawiera systematycznie aktualizowaną listę potencjalnych zagrożeń. Skanery zabezpieczeń szukają w systemie luk stanowiących zagrożenie. Niektóre skanery automatycznie analizują bezpieczeństwo na granicy między chronionym systemem, a światem zewnętrznym oraz bezpieczeństwo wewnętrzne. Testy mogą obejmować zagrożenia w takich typach oprogramowania, jak: systemy operacyjne, systemy sieciowe, przeglądarki WWW, śluzy bezpieczeństwa, pakiety naprawcze. Efektem analiz jest zapamiętanie wyników w bazie danych, przygotowanie raportów (w postaci pliku HTML, w formie graficznej, tekstowej, w formacie LaTeX lub innym), ocena ryzyka i wskazanie sposobów wyeliminowania znalezionych luk w zabezpieczeniach. Wykryte luki mogą być automatycznie eliminowane przez zainstalowanie odpowiedniego pakietu naprawczego. Bardziej zaawansowane narzędzia tego typu są w stanie testować wiele systemów jednocześnie, w celu wykrycia potencjalnych zagrożeń, które same w sobie nie stanowią niebezpieczeństwa, natomiast łącznie z wieloma innymi potencjalnymi zagrożeniami mogą wywołać poważne problemy.

Dokonujący szczegółowej analizy skaner generuje ogromne ilości danych (lista potencjalnych zagrożeń może zawierać setki lub tysiące elementów), stąd ważnym aspektem działania jest sposób prezentacji wyników analiz. Ponieważ nie można założyć, że wynik testów będzie zawsze analizowany przez specjalistę od zabezpieczeń, istotne jest, by raport miał postać zrozumiałą i ułatwiającą interpretację administratorowi systemu informatycznego. Znalezione zagrożenia powinny być uporządkowane od najpoważniejszych do błahych.

2.4. Zasady stosowania środków ochrony

Należy mieć świadomość tego, iż żadna z istniejących metod ochrony nie jest doskonała i nie eliminuje całkowicie zagrożeń. Zapewnienie pełnego bezpieczeństwa w praktyce nie jest możliwe. Ważnym czynnikiem wpływającym na bezpieczeństwo jest przestrzeganie zasad poprawnego stosowania środków ochrony [1][21][30][24]:

- uwzględnianie funkcji ochrony w całym procesie tworzenia, wdrażania i eksploatacji systemu informatycznego,
- korzystanie z systemów i mechanizmów sprawdzonych, certyfikowanych (np. zgodnie z TCSEC, ITSEC, CCITSE) przez niezależne instytucje,
- standaryzacja rozwiązań,
- ciągłość stosowania,
- poprawność implementacji,
- poprawność (stała kontrola) konfiguracji,
- akceptowalność przez użytkowników - brak akceptacji użytkowników może prowadzić do nieprawidłowości,

- maksymalnie duża automatyzacja – zwolnienie użytkowników i administratorów z obowiązku wykonywania dodatkowych procedur związanych z ochroną; ponieważ źródłem większości błędów jest człowiek, należy, tam gdzie jest to możliwe (np. przy wykonywaniu kopii zapasowych), ograniczać do minimum jego ingerencję w przetwarzanie – odpowiednie oprogramowanie i sprzęt powinny zapewniać pracę z niewielkim udziałem człowieka,
- integrowanie funkcji bezpieczeństwa z innymi elementami systemów informatycznych (ochrona antywirusowa w serwerze poczty elektronicznej lub w śluzie bezpieczeństwa, mechanizmy kryptograficzne w systemach operacyjnych),
- ścisła współpraca między różnymi mechanizmami bezpieczeństwa (np. system wykrywania włamań po wykryciu próby ataku komunikuje się ze śluzą bezpieczeństwa, która blokuje źródło ataku),
- nadzór nad sposobami korzystania ze środków bezpieczeństwa (np. kontrola złożoności haseł użytkowników),
- aktualizacja środków ochrony – konieczność uwzględniania nowych zagrożeń, zmian organizacyjnych,
- każde nieprawidłowe, nieoczekiwane, odbiegające od normy działanie systemu informatycznego powinno być poddane analizie w celu wykrycia przyczyny,
- opracowanie dokumentacji systemu zabezpieczeń,
- systematyczna kontrola stanu bezpieczeństwa – wykonywanie testów penetracyjnych, symulacji włamań,

- opracowanie i stosowanie polityki bezpieczeństwa – wymuszanie odpowiednich zachowań na użytkownikach i odpowiedzialność za popełniane błędy i niedbałość.

3. Rozszerzenie podstawy określenia wartości bezpieczeństwa sieci

3.1. Podstawowe współczynniki określające wartość bezpieczeństwa

Podając systemowi analizującemu pewne parametry, dzięki którym będzie w stanie określić aktualnie potrzebny poziom zabezpieczeń oraz zweryfikować już działający, będziemy mogli określić wartość bezpieczeństwa systemu w określonym momencie. Weryfikację aktualnie działającego sytemu zabezpieczeń można przeprowadzić na podstawie trzech parametrów: odporność (ξ), otwartość (η), przeciążalność (μ). Parametry te są na tyle ogólne, że można je zastosować niemal do wszystkich dostępnych elementów zabezpieczających [4].

Systemy komputerowe sieci korporacyjnych w chwili obecnej narażone są w sposób ciągły na próby zmiany stanu równowagi pomiędzy bezpieczeństwem, a ryzykiem systemu chronionego poprzez różnego rodzaju nieuprawnione działania osób niepowołanych w tym systemie. Korporacje nie mogą sobie pozwolić na ograniczanie czasowe dostępu do informacji poprzez zamykanie biur i wyłączanie serwerów o określonych porach oraz zamykanie swoich zasobów informatycznych przed klientem, w czasach kiedy działalność i wyniki korporacji kształtowane są przez prawa rynku.

Rozpatrzmy korporację, która udostępnia swoje strategiczne dane w sposób ciągły poprzez automatyczne usługi typu WWW, FTP, terminale itp. i jest otwarta dla klientów przez większość dnia (załóżmy 12 godzin dziennie).

W sytuacji takiej można dostrzec wiele zagrożeń i możliwości niepowołanego ingerowania w prawidłowe i bezpieczne działanie sieci korporacji. Z danych, uzyskanych na podstawie bazy danych działania systemu monitorowania dostępu do zasobów informatycznych jednej z uczelni wynika, że na przykład stanowiska komputerowe były użytkowane sumarycznie przez okres 806 dni 9 godzin i 44 minut (dane z działania w okresie od 20.03.2001 do 06.05.2001, wygenerowanych wpisów w bazie: 19 625, łączny czas w minutach: 1 161 224). W stosunku do pojedynczej maszyny daje to okres nieprzerwanego działania (24 godziny na dobę) średnio przez ponad 5 dni. W tym czasie do systemu sieciowego zalogowało się około 3 500 osób, gdyż tyle liczyła sieciowa baza użytkowników. W czasie tym zarejestrowano próby wpływania na QoS[4] (ang. *Quoality of Service*) oraz nieliczne próby złamania haseł, w tym administracyjnego [8][9][10].

Opisując system sieciowy i jego złożoność można natrafić na wiele problemów. Jednym z podstawowych jest niekompatybilność oprogramowania, a w szczególności systemowego. Różnice w zachowaniu się systemu operacyjnego utrudniają możliwości opisania zagrożeń, a zwłaszcza samej wartości bezpieczeństwa. Problem różnorodności systemów wynika z praw rynkowych i jest zjawiskiem jak najbardziej pożądanym (konkurencyjność firm). W celu przeprowadzenia dostatecznych analiz systemowych i programowych należy uprościć różnorodne parametry do jak najbardziej wspólnych cech dla każdego systemu. Miarę odporności systemu można przedstawić za pomocą wzoru (1).

[4] Miara jakości i płynności dostarczanych usług.

$$\xi = \frac{\sum_{i=0}^{n} k_i}{t(u)} \qquad (1)$$

gdzie:

k_i – jednostkowa przewidywana próba niepowołanego dostępu,

t – czas obsługi danego procesu użytkownika u.

Pod stopniem trwałości[5] środka obrony będziemy rozumieć ilość ataków, którym przeciwdziałać będzie rozpatrywany element obrony, w czasie obsługi przez użytkownika systemu zdalnego dostępu. Jak wynika z określenia (1), trwałość metody obrony może być ustalona na postawie badań eksperymentalnych, albo w procesie eksploatacji. Jest oczywiste, że wartość początkowa tego parametru nie powinna być równa zeru. Dlatego, przed włączeniem do systemu obrony oddzielnych elementów obrony, każdy element powinien być poddany próbie na trwałość ilości zagrożeń. Naturalnie w takich próbach należy symulować te zagrożenia, które są charakterystyczne dla systemu opisywanego.

Systemy obrony składają się z rzędu oddzielnych podsystemów, które związane są ze sobą. Dlatego istnieją możliwości wykorzystania różnych środków obrony z jednego podsystemu do drugiego, zmieniając przy tym tajne komponenty. Doprowadza to do tego, że ten sam środek obrony, jeżeli wykorzystuje się go w różnych podsystemach ochranianego systemu, posiada różny stopień otwartości. Ocena otwartości elementów obrony jest ważną charakterystyką, o ile obciążenie tych środków jest jedną z szeroko stosowanych metod ataków. Wartości, jakie przechowują systemy nadzoru,

[5] Trwałość – odporność w czasie, odporność na możliwe ataki.

mogą być różne. Istotna jest ilość informacji wspólnych pomiędzy ośrodkami chroniącymi. Prawie każda aplikacja bazuje na wewnętrznych tablicach podręcznych (ang. *cache*). Zapisane tam informacje służą głównie przyspieszeniu procesu weryfikacji zabezpieczeń. Jeżeli możliwe będzie złamanie (nieuprawniony dostęp) elementu zabezpieczającego na przykład poprzez dostęp do danych przechowywanych w pamięci operacyjnej poprzez opisywany element obrony, możliwe również będzie wykorzystanie tam zapisanych informacji. Działanie takie pozwoli na próby wydobycia ważnych parametrów, na przykład cech użytkownika takich jak nazwa konta i hasło oraz wykorzystanie poznanych informacji w innym etapie zabezpieczeń. Opisywany przykład jest najbardziej oczywistym w wykorzystaniu do nieuprawnionych działań i sam nasuwa dalsze możliwości jego wykorzystania. Parametry przechowywane przez podprocesy w systemie bezpieczeństwa mogą być różne. Żaden z podsystemów nie musi przechowywać informacji najbardziej oczywistych, jak opisywane wcześniej hasło i nazwa konta użytkownika. Mogą to być złożone formy przepływu informacji pomiędzy lokalnymi ośrodkami zabezpieczeń. Zakładając wcześniej przytaczany wspólny element w różnych etapach procesu bezpieczeństwa i możliwe przełamanie zabezpieczeń informacji przetrzymywanych w pamięci podręcznej, możemy sobie wyobrazić potencjalne wykorzystanie zgromadzonych informacji w pośrednich elementach procesu zabezpieczeń z możliwością modyfikacji danych przetwarzanych (przecież udało się nam je poznać, więc możemy również je modyfikować).

Fakt ten może zobrazować zależność (2), która mówi, że miarą otwartości środka obrony jest stosunek ilości otwartych komponentów środka obrony do ogólnej ilości ukrytych komponentów w danym środku obrony [8][9][10].

$$\eta = \alpha * \sum_{i=1}^{n} \left(\frac{M}{m} \right)_i \quad \textbf{(2)}$$

gdzie:

m – ilość otwartych ukrytych komponentów,

M – całkowita ilość ukrytych komponentów,

i – ilość środków obrony,

α - współczynnik doprowadzenia stopnia otwartości do jednostki pomiaru.

Większość elementów zabezpieczających posiada jako podstawowy mechanizm bezpieczeństwa proces uwierzytelniający, który bazuje na identyfikatorze obiektu i haśle przypisanym do tego identyfikatora. Opisywany współczynnik jest elementem trudnym do wyznaczenia ze względu na konieczność poznania architektury wewnętrznej procesu chroniącego pod kątem posiadanych (gromadzonych) danych. W wyniku doświadczeń stworzono opis matematyczny, który pozwala wyliczyć wartości M i m potrzebne do prawidłowego przeliczenia współczynnika η (2).

$$M_{t_i} = N * \gamma \quad \textbf{(3)}$$

gdzie:

N – maksymalna ilość haseł w systemie

γ – współczynnik określony wzorem

$$\gamma = \frac{\sum_t \beta_t}{\varepsilon} \quad \textbf{(4)}$$

gdzie:

ε – ilość zalogowanych na moment t_i użytkowników

β – średnia długość hasła wyrażona wzorem

$$\beta = \frac{l_{\max} - l_{\varepsilon_i}}{l_{\max} - l_{\min}} \tag{5}$$

gdzie:

l_{max} – maksymalna długość hasła

$l_{\varepsilon i}$ – długość hasła użytkownika ε_i

l_{min} – minimalna długość hasła

W przypadku wyznaczenia m musimy uwzględnić każdy proces P_i, który rozpoczyna się prawidłowym logowanie i może być przez system zabezpieczeń traktowany jako atak na moment t_i. Dla wyznaczenia m należy wówczas sumować każdy opisywany proces P_i zgodnie z zależnością:

$$m = \sum_{i=1}^{n(t_i)} P_i \tag{6}$$

gdzie:

P_i – i-ty proces

n – ilość zdarzeń

Kolejnym charakterystycznym parametrem systemu chronionego niezależnie od jego rodzaju jest obciążalność. Obciążającą zdolnością elementów obrony informacji nazywa się wielkość, charakteryzującą ilość usług, które mogą być przetwarzane środkami obrony w danym okresie czasu. W sposób matematyczny opisać ją można przy pomocy wzoru (7):

$$\mu = \frac{n * u_k}{\sum_{i=0}^{m} (z_i * t_i(u_k))} \tag{7}$$

gdzie:

u_k – k-ta usługa,

n – ilość funkcjonalnych działań obsługi,

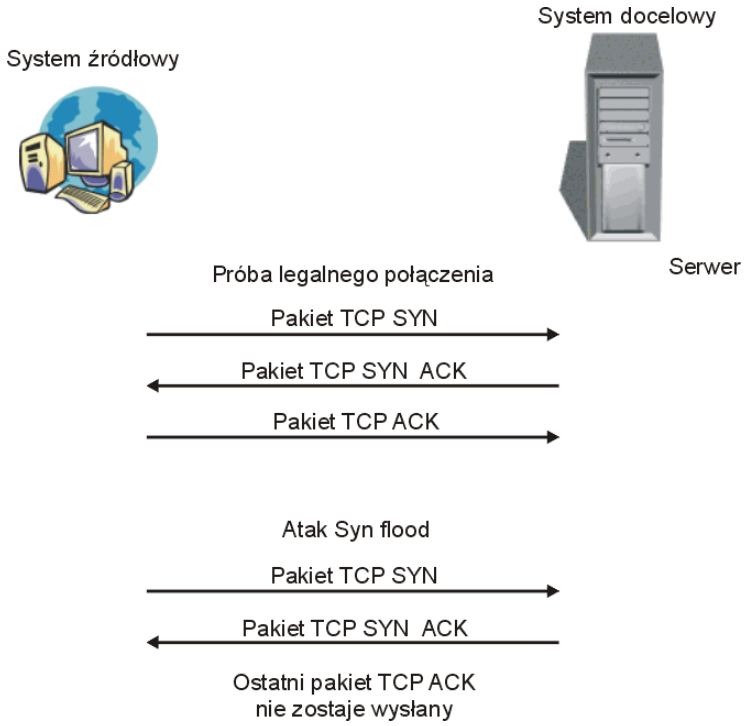
z_i – i-te zapotrzebowanie,

$t_i(u_k)$ – i-ty czas obsługi k-tej usługi

Niedostateczna możliwość przeciążenia może doprowadzić do większej wrażliwości na jakiekolwiek ataki spowodowane rozległymi możliwościami dostępu do k-tej usługi. DoS lub dDoS jest obecnie najpopularniejszą formą ataków na sieci korporacyjne [23][24]. Obrona przed takimi rodzajami ataków w stosunku do sieci komputerowej nie może być prowadzona poprzez system sieciowy.

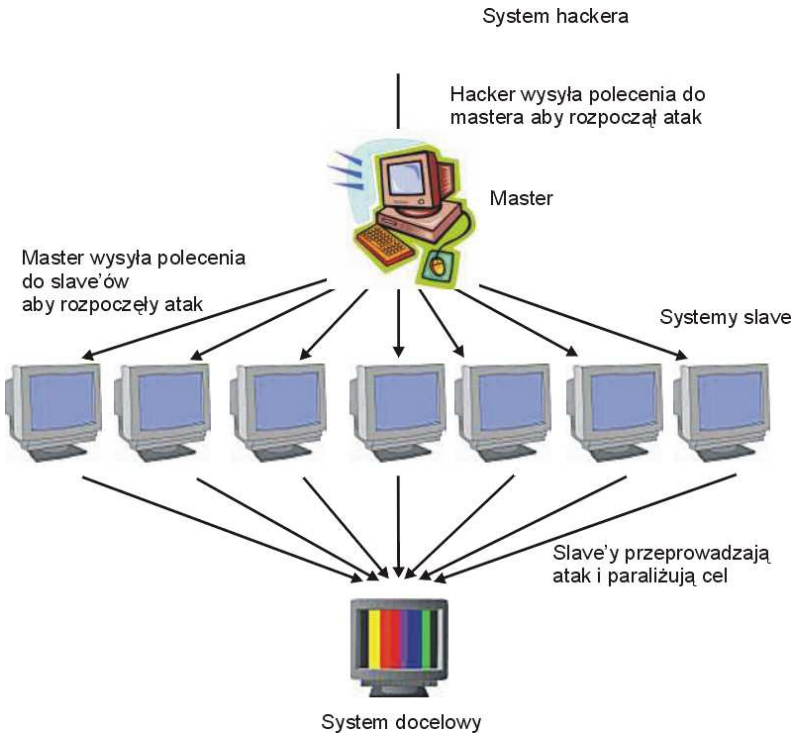
Każdy chyba już zdaje sobie sprawę z możliwości nadmiernego przeciążenia swojego systemu domowego poprzez zaalokowanie dostępnych zasobów danej jednostki komputerowej na przykład przez zajęcie dostępnej przestrzeni pamięci operacyjnej lub mocy obliczeniowej procesora. Wiemy również, że w czasie przeciążenia inne aplikacje lub podsystemy sprzętowe wymagające dostępu do przeciążonych elementów (procesor, pamięć) mogą działać w sposób niezadowalający lub w ekstremalnych warunkach odmówić współpracy. Na takich właśnie problemach opierają się ataki na przeciążenie.

Obecnie stosowane metody ataku przeciążeniowego opisywane są ogólnie jako DoS lub Distributed DoS. Różnica polega na tym, że proces DoS (rysunek 6) opiera się na jednym lub małej ilości hostów atakujących, a dDoS (rysunek 7) wykorzystuje do tego celu duże zorganizowane ośrodki lub wiele pojedynczych stacji komputerowych podłączonych do sieci Internet i zarządzanych pośrednio przez włamywacza [23][24][[35][45].



Rysunek 6 Standardowy atak DoS

Tego typu atak jak DoS jest oczywiście znany i istnieją odpowiednie metody ochrony przed nim. Jakość obrony zależy wyłącznie od skali ataku i właśnie parametrów przeciążalności założonych w systemie chronionym.



Rysunek 7 Atak typu Distributed DoS

Głównym celem włamywaczy są systemy rozwiązywania nazw (ang. *name resolving*) takie jak DNS (ang. *Domain Name System*). Jednostki serwerowe tego typu lub można powiedzieć inaczej, jednostki serwerowe świadczące usługę DNS są tam mocno narażone na atak tego typu ze względu zmiany relacji nazwa adres IP i przekierowanie całego ruchu sieciowego (internetowego) do stacji włamywacza. Niestety, nie tylko usługi takie jak opisywany system DNS mogą zostać zaatakowane w ten sposób. Dotyczy to wszystkich znanych usług dostępnych dla użytkowników sieci. Doprowadzenie do przeciążenia danej usługi właśnie podatnej na takie działanie doprowadzić może do spowolnienia lub w szczególnym przypadku braku odpowiedzi od innych usług działających na takim atakowanym hoście. Przeciążenie mocy procesora lub zasobów pamięci operacyjnej poprzez nieodporne usługi jest najczęstszym problemem systemów bezpieczeństwa. Niestety, nie musi to być spowodowane przez

wadliwe działanie systemów bezpieczeństwa. Czasami sytuacja jest bardziej prozaiczna. Częsty powód stanowi niewłaściwe ulokowanie usług na serwerach. Przykładem może być umieszczenie wszystkich kluczowych usług w sieci na jednym serwerze, łącznie z możliwością pracy na przykład terminalowej również użytkowników tej sieci. Można wówczas liczyć się z sytuacją, że nawet dobrze przygotowany system bezpieczeństwa odporny na przeciążenia napływające z sieci w powiązaniu z obciążeniem generowanym przez użytkowników na tym samym serwerze doprowadzić może do katastrofy [34][51]. Zaproponować można prosty kod programowy pozwalający na powielanie własnej kopii w systemie operacyjnym i w ten sposób rezerwujący coraz większe zasoby systemowe (przykładowy kod poniżej).

```
main()
     {
     while (1) fork();
     }
```

Ten prosty kod programu wykonany przez użytkownika może spowodować dużo zamieszania w systemie. Ważnym staje się tu właśnie pojęcie ochrony kont użytkowników, nie tylko kont administracyjnych. Wcześniej prezentowane parametry pozwalają właśnie na opisanie całości zagadnienia. Dostęp do kont zwykłych użytkowników serwera może pominąć całą mozolną pracę włożoną w proces zabezpieczania przed przeciążeniem poszczególnych komponentów obrony, ponieważ na przykład poprzez wykorzystanie tak prostego kodu programowego możliwe jest spowodowanie awarii całego serwera.

Różne metody złamania systemu bezpieczeństwa nie ograniczają się jedynie do tych już przytoczonych wcześniej. Istnieje wiele równoważnych metod lub problemów, na przykład:

- ataki dyskowe,
- ataki na przestrzeń wymiany (tmp, swap),
- ataki na system alokacji dyskowej (i-node),
- przepełnienie bufora kodu programistycznego,
- nadpisywanie zmiennych systemowych,
- podmiana dowiązań symbolicznych,
- ataki przez pocztę elektroniczną,
- konie trojańskie i wirusy,
- ataki przeprowadzane kanałem poleceń (ang. *command-channel attacks*),
- ataki wykorzystujące dane (ang. *data-driven attack*),
- ataki na usługi trzecie (ang. *third-party attack*),
- sfałszowanie uwierzytelniania klientów (ang. *false authentification*),
- inżynieria społeczna.

Wszystkie wymienione przykłady możliwe są do wykonania w sposób zdalny lub lokalny, gdzie zdalny rozumie się jako działanie prowadzone na danych napływających ze środowiska sieciowego i operacjach na usługach, a lokalne jako działania wykonywane na systemie operacyjnym danego serwera sieciowego. Praktycznie wszystkie możliwe są do poddania analizie prowadzonej na zaproponowanych współczynnikach opisujących wartości systemu bezpieczeństwa, poza jednym (inżynieria społeczna). Przytoczony problem dotyczy tzw. „czynnika ludzkiego". Bardzo popularne stwierdzenie na usprawiedliwienie błędów działania człowieka w czasie współpracy z systemem elektronicznym. Niemniej jednak należy pamiętać, że żadne

współczynniki i liczby nie pomogą dobrze chronić swoich zasobów elektronicznych o ile nie będziemy, łącznie z wprowadzaniem kolejnych metod bezpieczeństwa w odpowiedni sposób dbali o zwiększanie świadomości i doświadczenia personelu przeznaczonego do opieki nad systemami sieciowymi, jak również samych użytkowników tychże systemów [30].

Przedstawiane tu problemy mają głównie na celu uświadomienie złożoności problemu zabezpieczeń, a także analizy ryzyka konkretnego celu (na przykład pojedynczego serwera).

Bezpieczeństwo sytemu będzie opisem sumy wszystkich składowych bezpieczeństwa i można przedstawić je za pośrednictwem zależności (8):

$$B(S) = \sum_{i=0}^{n} \xi_i + \sum_{j=1}^{m} \eta_j + \sum_{r=0}^{k} \mu_r \qquad \textbf{(8)}$$

Zależność przedstawiona w taki sposób może okazać się niewystarczająca w opisie złożoności systemu chronionego, jednak na podstawie przeprowadzonych badań jej dokładność w stosunku do raczej nieskomplikowanych pojedynczych systemów nadzoru sieciowego ukierunkowanych do wydzielenia dedykowanych rozwiązań serwerowych do poszczególnych usług, jest dość wysoka i zadowalająca. Wartość powstała z procesu sumy, jaką jest *B(S)* będzie główną wartością bezpieczeństwa systemu [8][9][10]. Wartość ta jednak nie uwzględnia głównego czynnika jakim jest czas. Zaprezentowana dalej wartość *B(S)* będzie już rozpatrywana pod kątem czasu.

3.2. Opracowanie modeli ryzyka gromadzenia i przesyłania informacji

3.2.1. Bezpieczeństwo a ryzyko

Bezpieczeństwo systemu chronionego to wartość jaka możliwa jest do wyznaczenia na podstawie badania aktualnie wdrożonych systemów bezpieczeństwa. Wartość ta określa niewrażliwość na próby nieautoryzowanego dostępu i nie określa czy takie próby (nieautoryzowanego dostępu) były wykonywane. Ryzyko opisuje chęć lub bardziej prawdopodobieństwo zaistnienia próby nieautoryzowanego dostępu. Innymi słowy można powiedzieć, że bezpieczeństwo jest miarą odporności na takie próby, a ryzyko opisuje prawdopodobieństwo wystąpienia takich prób oraz zwiększenie ryzyka systemu następuje również na podstawie analizy ilości takich nieuprawnionych prób dostępu. Ryzyko rośnie wraz z przyrostem takich prób.

Trudno jest obecnie określić, przy pomocy jakiej metody wyznaczyć wartość ryzyka chronionego systemu, czyli miarę podatności tego systemu na włamania lub zakłócenia. Wartość ta jest niezmiernie potrzebna, jednak problemem są niezliczone metody włamań opisywane wcześniej oraz możliwości ich wykrycia w odpowiednio skończonym czasie. Do tej pory ryzyko wyznaczane jest na drodze analiz informacji zgromadzonej w systemach chronionych oraz rejestracji potencjalnych włamań do takiego systemu i ich analizie celem odparcia kolejnych prób. Do tego celu wykorzystuje się różnego rodzaju systemy IDS zaopatrzone w heurystyki analizy ryzyka i w znane prototypy włamań. Przykładem takiego postępowania może być opracowanie dr. Andrzeja Białasa polegające w uproszczeniu na opisaniu wszystkich udokumentowanych procedur bezpieczeństwa i przypisanie im odpowiednich wartości procentowych na

podstawie zbierania opinii od osób odpowiedzialnych za dane procedury, jak również administratorów sieciowych i prowadzenia obliczeń uśredniających. Metody te są niestety niewystarczające, jednak do chwili obecnej nie ma lepszych metod.

W poprzednim rozdziale zaprezentowane zostały współczynniki, które pozwalają na stwierdzenie aktualnego bezpieczeństwa systemu. Niestety posiadanie jedynie tej informacji nie odnosi się w żaden sposób do rzeczywistości. Wszyscy zdają sobie sprawę, że każdy element w sieci może być inaczej zagrożony w zależności od tego jaką informację przenosi. Poznanie zatem samej wartości bezpieczeństwa mówi jedynie o jakiejś liczbie w pojęciu matematycznym nie mającej punktu odniesienia do stanu faktycznego. Aby można kompleksowo rozważać ochronę konkretnego systemu należy również oszacować ryzyko. Według normy PN-I-02000 [140] ryzyko to:

„… prawdopodobieństwo, że określone zagrożenie wykorzysta określoną podatność systemu przetwarzania danych".

A według normy PN-I-13335-1:1999 [141]:

„… ryzyko jest prawdopodobieństwem określającym możliwość wykorzystania określonej podatności przez dane zagrożenie w celu spowodowania straty lub zniszczenia zasobu lub grupy zasobów, a przez to negatywnego bezpośredniego lub pośredniego wpłynięcia na instytucję".

Ryzyko jest nieodzownym parametrem bezpieczeństwa i całkowicie zależnym od niego. Ryzyko jest wartością przewidywaną, czyli musi być opisywane przez prawdopodobieństwo. Jednak oszacowanie ryzyka[6] w stanie początkowym działania systemu jest niewystarczające. Przykładem może być fakt, że system po samym uruchomieniu, nie posiadający żadnych

[6] W literaturze można spotkać jeszcze inne określenie – ocena ryzyka.

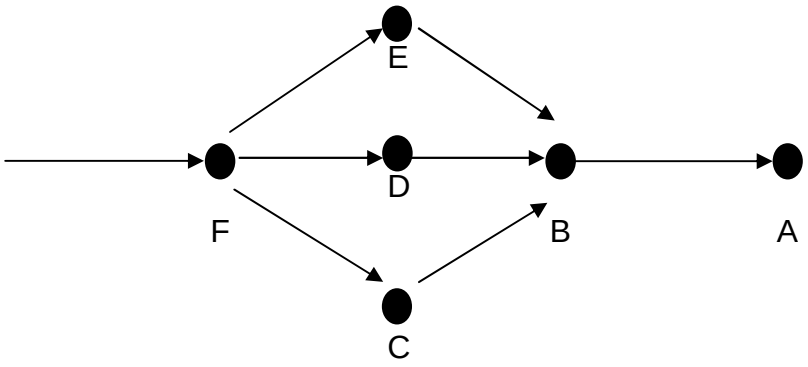
danych obarczony jest mniejszym ryzykiem niż ten sam system w niezmienionej postaci ale na przykład z krytycznymi danymi dla jakiejś teoretycznie rozważanej korporacji. Ujawniająca się w tym przykładzie relacja z czasem jest podstawą do konieczności wprowadzenia takich samych założeń jak dla wyznaczenia bezpieczeństwa w stosunku do wyznaczania ryzyka.

Gromadzenie informacji o bezpieczeństwie pozwoli zatem na dokładniejsze wyznaczanie samego ryzyka systemu. Poprzez zastosowanie reprezentacji drzew błędów i zdarzeń zaprezentowanych w następnym rozdziale, które pozwalają opisać złożoność zależności poszczególnych metod włamania oraz stany systemu chronionego w określonej jednostce czasu, można przeprowadzić odpowiednie matematyczne działania, które doprowadzą do wyznaczenia wartości ryzyka systemu zabezpieczanego [7].

3.2.2. Wyznaczanie ryzyka

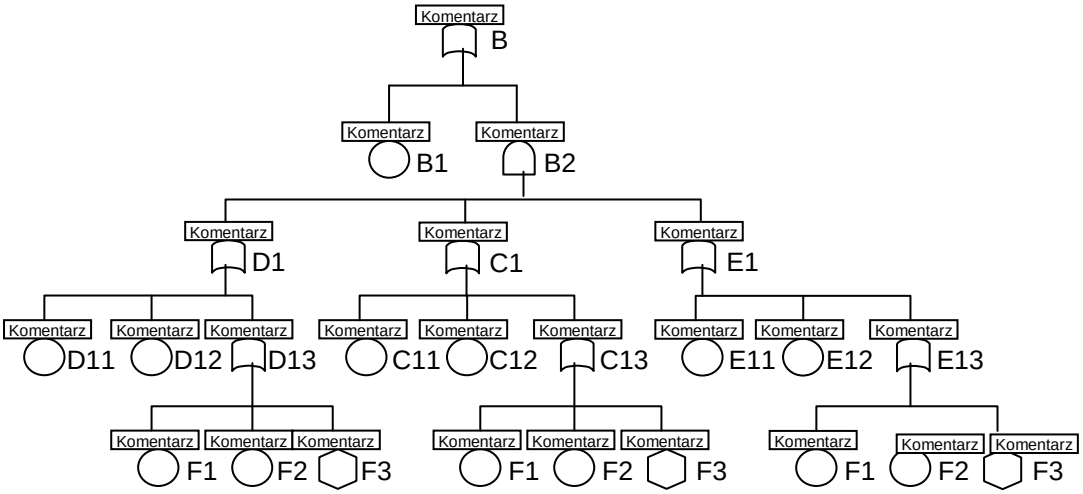
Każdy chroniony system jest zabezpieczony przez wiele podsystemów bezpieczeństwa, które razem stanowią system zabezpieczeń. Poszczególne części składowe systemu zabezpieczeń w określonej chwili czasu znajdują się w jakimś stanie zależnym od prowadzonych przez system lub użytkownika operacji. Wyznaczenie tych stanów możliwe jest z wykorzystaniem metody drzew logicznych. Podstawowymi elementami niezbędnymi do przeprowadzenia tej metody są sygnały wejściowe, które bezpośrednio wpływają na wybrany do analizy podsystem bezpieczeństwa. Sygnały te mogą występować samodzielnie lub być powiązane ze sobą w określonej zależności [6][7].

Załóżmy, że zajście zdarzenia *A* zależy od zdarzenia *B*. Zaś zdarzenie *B* zależeć będzie od *C, D, E*. Wtedy zależność będzie miała postać:



Rysunek 8 Opis zależności zdarzeń

Każde ze zdarzeń może mieć kilka zupełnie różnych wyników. Niech *B* posiada możliwości *B1*, *B2* z których *B2* zależy od *D*, *C* oraz *E*. Zdarzenia *D*, *C* oraz *E* mogą być zależne od warunków *D1*, *C1*, *E1*, a te od *D11*, *D12*, *D13*, *C11*, *C12*, *C13*, *E11*, *E12*, *E13*, zaś *D13*, *C13*, *E13* odpowiednio od *F1*, *F2*, *F3*. Wtedy drzewo błędów przyjmie następującą postać:



Rysunek 9 Opis zależności zdarzeń w drzewie błędów

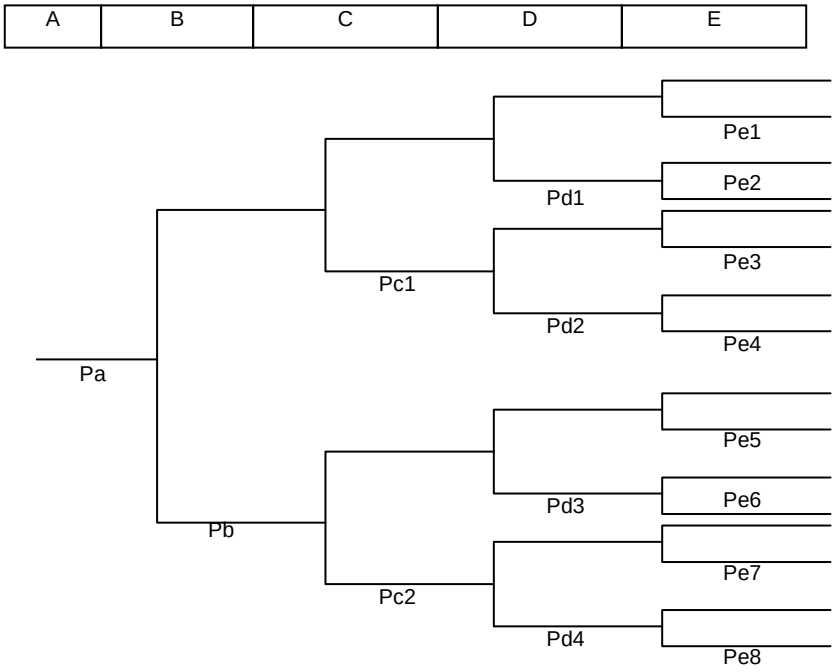
Do budowy drzewa stanów można używać następujących funkcji: AND, OR, NOT AND, NOT OR, warunkowy AND. Drzewo stanów odwzorowuje zbiór minimalnych zdarzeń. Metoda ta pozwala określić tablicę stanów niezbędną do budowania drzewa zdarzeń. Prezentowane na rysunku 9 drzewo można zapisać następującym wzorem:

$$\begin{aligned} B &= B1 + B2 = B1 + (D1 * C1 * E1) = B1 + [(D11 + D12 + D13) * \\ &* (C11 + C12 + C13) * (E11 + E12 + E13)] = \\ &= B1 + \{[D11 + D12 + (F1 * F2 * F3)] * \\ &[C11 + C12 + (F1 * F2 * F3)] * [E11 + E12 + (F1 * F2 * F3)]\} \end{aligned} \quad \textbf{(9)}$$

Drzewa zdarzeń są logicznym przedstawieniem znaczących reakcji na początkowe zdarzenia. Każde drzewo zdarzeń składa się z diagramu (grafu) i tablicy stanów podsystemów bezpieczeństwa. Diagram buduje się od lewej strony zaczynając od zdarzenia końcowego, a następnie przechodząc w prawo opisuje się pośrednie możliwe stany. Stany pośrednie obrazowane są poprzez dwie linie: górną i dolną. Linia górna reprezentuje zajście określonej sytuacji, która doprowadza do zdarzenia po stronie lewej

w sposób niemożliwy do rejestracji w stanie obserwowanym. Linia dolna to prawdopodobieństwo zajścia zdarzenia z lewej strony w stanie obserwowanym.

Dla opisywanego wcześniej drzewa błędów może przyjąć ono postać przedstawioną na rysunku 10.



Rysunek 10 Drzewo zdarzeń z tablicą stanów A, B, C, D, E

Powstałe w ten sposób drzewo pozwala na wyznaczenie prawdopodobieństwa zajścia zdarzenia w dowolnym stanie systemu bezpieczeństwa, zależnie od potrzeb. Przejście systemu bezpieczeństwa do stanu A, B, C, D, lub E obliczyć można na podstawie logicznych zależności sygnałów w drzewie błędów. Prawdopodobieństwo $P_{i,j}$ zmienia się w trakcie działania systemu w zależności od zdarzeń, które powstają w systemie bezpieczeństwa. Indeksy i oraz j prawdopodobieństwa P to odpowiednio oznaczenie stanu, w którym obserwowane jest określone prawdopodobieństwo (np.: w stanie A będzie to zapis P_a), zaś drugi indeks

pozwala opisać składowe prawdopodobieństwa *Pi* w określonym stanie, gdy ich ilość jest większa od 1 (np.: w stanie D posiadamy 4 składowe prawdopodobieństwa *Pd* i oznaczone zostały poprzez Pd_1, Pd_2, Pd_3, Pd_4). Przykładem zilustrowania sposobu opisu zajścia zdarzeń zgodnie z drzewem zdarzeń przedstawianym na rysunku 10 może być zależność:

$$P_a * P_b * P_{c2} * P_{d4} * P_{e8} \tag{10}$$

Opisuje ona iloczyn prawdopodobieństw, jaki zachodzi w najniższej z gałęzi prowadzącej do zdarzenia, które zachodzi w stanie A.

Wyznaczanie *R(S)* (ryzyko systemu zabezpieczanego) należy rozpocząć od obliczenia prawdopodobieństwa Q_i jako średniego prawdopodobieństwa na podstawie prawdopodobieństw cząstkowych z drzewa zdarzeń, prowadzącego do określonego zdarzenia. Jeżeli określimy przez *RIR* (ang. *Risk Increase Ratio*) wartość przyrostu ryzyka wzorem przedstawionym poniżej

$$RIR = \frac{F(1)}{F(Q)} \tag{11}$$

gdzie:

F(1) – częstość występowania zdarzenia w przypadkach nie zarejestrowanych przez podsystem bezpieczeństwa w określonym stanie,

F(Q) – częstość występowania zdarzenia z prawdopodobieństwem *Q* zarejestrowanych przez podsystem bezpieczeństwa.

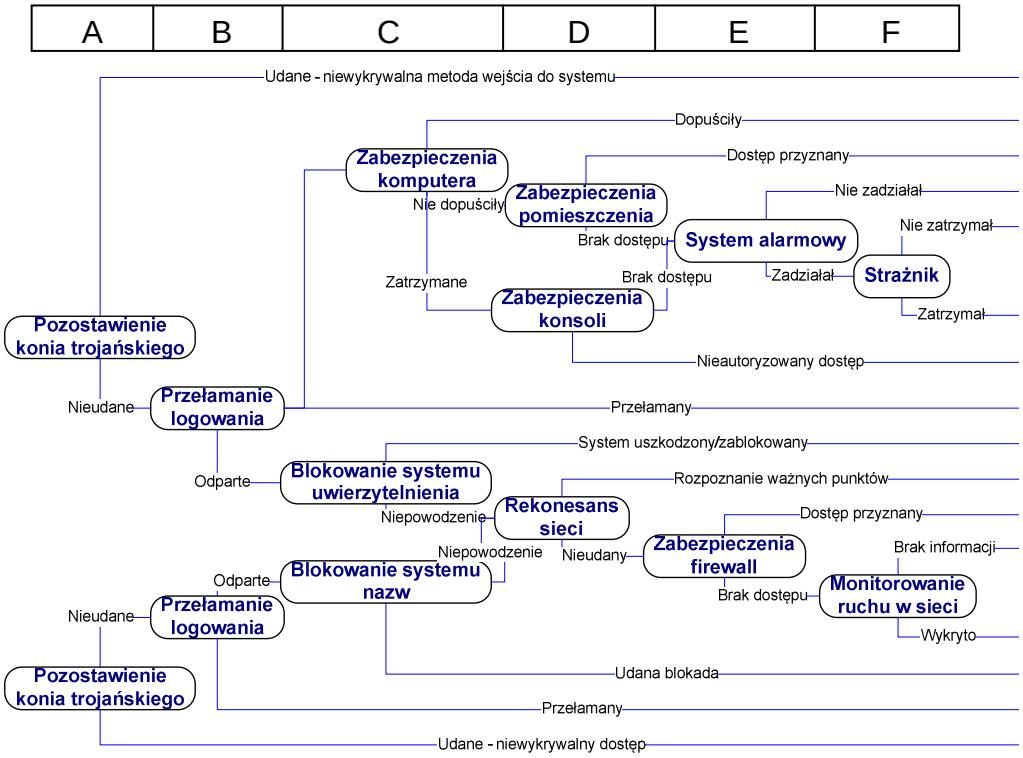
to wartość ryzyka systemu na chwilę *t* opiszemy przy pomocy zależności

$$R(S)_t = \sum_{i=1}^{n} \frac{RIR(Q_i)_t}{n} \tag{12}$$

gdzie:

n – ilość możliwych początkowych stanów zajścia zdarzenia końcowego.

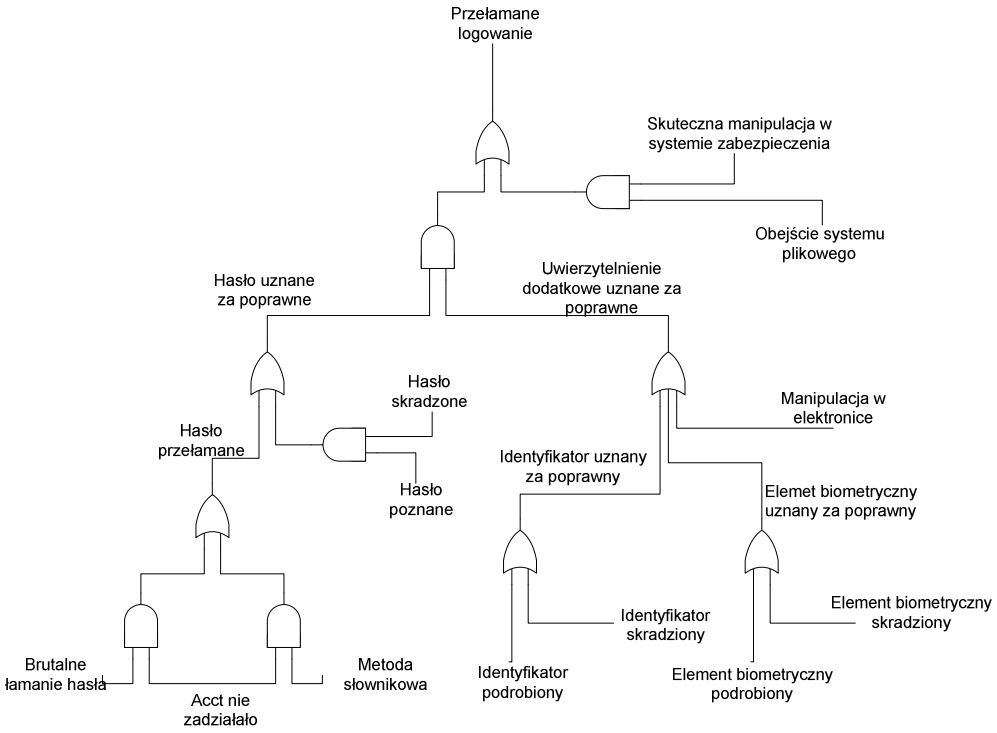
Głównym problemem tworzenia opisywanych drzew może być złożoność procesów opisywanych. Nawet najprostszy z punktu widzenia proces uwierzytelnienia zależy od stanu innych podprocesów, które on wykorzystuje. Dla przykładu proces uwierzytelnienia może być powiązany z systemem accountingu, który odpowiada za przeprowadzanie działań w chwili prawidłowego, jak i nieprawidłowego procesu uwierzytelnienia.



Rysunek 11 Przykładowe drzewo zdarzeń w procesie włamania sieciowego i lokalnego

Zależeć może również od podsystemu przekazywania sekretów takich jak hasło, czyli na przykład systemów PAM (ang. *Pluggable Authentication Module*), którymi są zestawy narzędzi systemu porównywania hasła przyklejone do systemu operacyjnego. Skomplikowanie zależności rośnie

wraz z ilością podsystemów zabezpieczających, jak również możliwościami przełamania ich. Przykładowe zależności przedstawiono na rysunku 11 i 12.



Rysunek 12 Drzewo błędu dla zdarzenia „przełamanie logowania"

Parametr, jakim jest *R(S)* czyli ryzyko systemu należy do najistotniejszych. To właśnie ta wartość doprowadziła do powstania systemów bezpieczeństwa, a obecnie całkowicie się z nim splotła. Nie można rozpatrywać rozdzielnie tych parametrów. Istnienie jednego jest całkowicie zależne od drugiego. Systemy sieciowe, komputery i inne urządzenia oraz procedury przekazywania danych muszą być rozpatrywane pod względem obu parametrów. W następnej części zaprezentowane zostanie powiązanie i podstawowe zależności pomiędzy tymi wyznacznikami bezpieczeństwa aktualnych sieci i stacji komputerowych [6][7][8].

3.3. Wyznaczanie poziomu gwarancji bezpieczeństwa

Wszystkie obecnie znane metody prowadzą do wyznaczenia dwóch wartości: bezpieczeństwa i ryzyka. Metody ich wyznaczenia mogą być różne i abstrahując od faktu ich statycznej wartości (określonej w danej chwili bez ciągłego wyznaczania w czasie działania systemu), mówią one głównie o bezpieczeństwie. Istnieją również próby połączenia tych wartości w jedną spójną całość. Przykład taki opisywany jest przez Krzysztofa Lidermana, który stara się powiązać ryzyko z bezpieczeństwem przyjmując pojęcie „ryzyka szczątkowego" jako wartości równicy pomiędzy idealnym bezpieczeństwem (byłby to poziom 100%), a wyznaczonym w procesie weryfikacji. Sam autor stwierdza jednak, że nie jest możliwym osiągnąć wartość idealną.

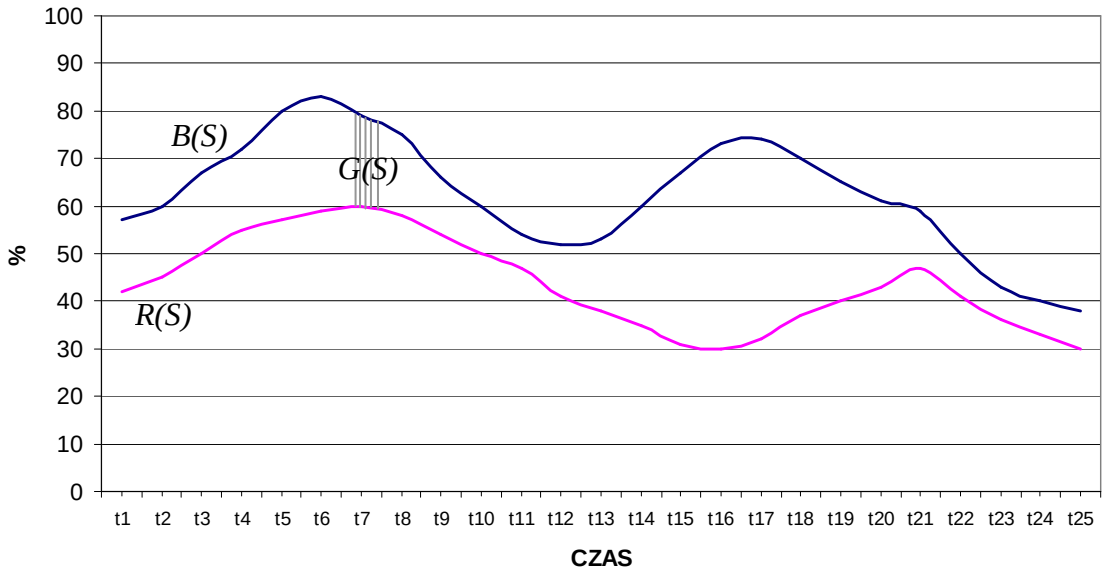
W związku z powyższym należy przyjąć, że wartość bezpieczeństwa powinna dążyć do wartości idealnej, ale nigdy jej nie osiągnie. Wartość ryzyka zaś powinna być zatem redukowana do zera, ale również jest to nie osiągalne, ponieważ ryzyko jest przeciwieństwem bezpieczeństwa, a bezpieczeństwo – jak założyliśmy – nie jest w stanie osiągnąć wartości idealnej. Jeżeli istnieje zawsze ryzyko (nawet przy wartościach szczątkowych większych od zera) i może być wyznaczane niezależnie od bezpieczeństwa (zbieranie informacji do wyznaczenia bezpieczeństwa będzie jedynie elementem uzupełniającym poprawność wyznaczenia ryzyka) to należy przyjąć inny parametr, który przedstawi aktualny stan rozpatrywanego systemu. W tym celu proponuję przyjąć parametr jakim będzie poziom gwarancji bezpieczeństwa.

W celu przejścia do opisu poziomu gwarancji bezpieczeństwa należy w pierwszej kolejności przedstawić matematyczny obraz bezpieczeństwa

systemu $B(S)_t$ zależność (8). Kolejnym krokiem jest reprezentacja w tych samych zakresach czasowych wartości ryzyka $R(S)$ przedstawionego za pośrednictwem wzoru (12). Poszczególne składowe zależności (12), jakimi są $RIR(Q_i)$ opisanymi za pomocą wzoru (11), są cząstkowymi wartościami zmiany współczynnika zwiększenia ryzyka obliczanego dla poszczególnych aplikacji (usług) w różnych kombinacjach (jeżeli takie zachodzą). Na podstawie tych dwóch wartości analizowanych w czasie t można wyznaczyć zależność:

$$G_i(S,t) = B_i(S,t) - R_i(S,t) \qquad \textbf{(13)}$$

Wartość $G_i(S,t)$ można przedstawić w postaci pola ograniczonego krzywą $B(S,t)$ i $R(S,t)$. Przykładową zależność przedstawiono na rysunku 13.



Rysunek 13 Wartość gwarancji bezpieczeństwa w czasie

Do prawidłowego działania systemu informatycznego wartość *B(S)* powinna być większa od wartości *R(S)* w chwili t_i. Optymalnym rozwiązaniem byłoby uzyskanie wartości min. $G_i(S,t)$, tak aby nakład kosztów związanych z wdrożeniem systemów bezpieczeństwa odpowiadał odpowiedniemu ryzyku działania systemu informatycznego. Wartość $G_i(S,t)$ nie powinna być jednak równa zeru ze względu na niewystarczający margines bezpieczeństwa. Margines bezpieczeństwa, czyli minimalna wartość $G_i(S,t)$ powinna być ustalana osobno dla każdego systemu informatycznego, zależnie od jego przeznaczenia [2][3][4][5].

3.4. Podsumowanie

Autorska propozycja współczynników jakimi są: odporność, otwartość, przeciążalność pozwalają w zrozumiały sposób określić niezbędną wartość bezpieczeństwa sieci korporacyjnej. Zaprezentowane w tej kwestii odpowiednie wzory zostały sprawdzone w warunkach laboratoryjnych i udowodniły słuszność rozpatrywanej analizy bezpieczeństwa.

Dodatkowo zaprezentowano autorską metodę wyznaczenia ryzyka bazując o reprezentację drzew zdarzeń i błędów. Metoda dokładniej została opisana w następnym rozdziale. Dla tej metody wykonano również badania eksperymentalne, który pozwoliły zweryfikować słuszność założeń.

Głównym elementem rozdziału jest nowatorskie podejście do wyrażenia bezpieczeństwa poprzez poziom gwarancji bezpieczeństwa. Obecnie nie rozpatrywano takiej możliwości. Zaprezentowana przez autora opisywana wartość w znaczny sposób powinna się przyczynić do prostszego spojrzenia na aspekty związane z bezpieczeństwem i ryzykiem dla przeciętnego użytkownika sieci, który oczekuje jasnego i zrozumiałego parametru pozwalającego mu wybrać bezpieczną pracę z systemem sieciowym.

4. Badanie dynamicznych modeli zabezpieczenia poziomu bezpieczeństwa

4.1. Badanie parametrów określających współczynniki bezpieczeństwa

W poprzednim rozdziale przedstawione zostały trzy współczynniki, za pośrednictwem których można w sposób wystarczający przetestować bezpieczeństwo, a w szczególności składowe bezpieczeństwa, jakimi są poszczególne elementy chroniące system sieciowy. Każdy z parametrów był testowany na przygotowanych do tego celu systemach wydzielonych z rzeczywistego środowiska pracy. W warunkach laboratoryjnych – bo o takich w chwili obecnej tu mowa – zostały przygotowane testy z wykorzystaniem znanych metod włamań i zaburzeń systemów sieciowych, celem sprawdzenia prawidłowości teorii. Testowaniu podlegały główne aspekty systemów zgodnie z przyjętymi wcześniej elementami składowymi sprawdzanych współczynników.

W przypadku *odporności*[7] testowano dokładność i jakość informacji rejestrowanych przez podsystemy bezpieczeństwa w dziennikach zdarzeń systemu lub własnych konkretnych aplikacjach. Wykonano szereg prób polegających na kontrolowanych atakach w określonych jednostkach czasu na wybrane popularne usługi i analizie informacji zapisanych w wyżej wymienionych dziennikach systemowych.

Jedną z testowanych usług były połączenia SSH[8] (ang. *Secure Shell*) i główną uwagę skierowano na informacje pozostawione przez próby

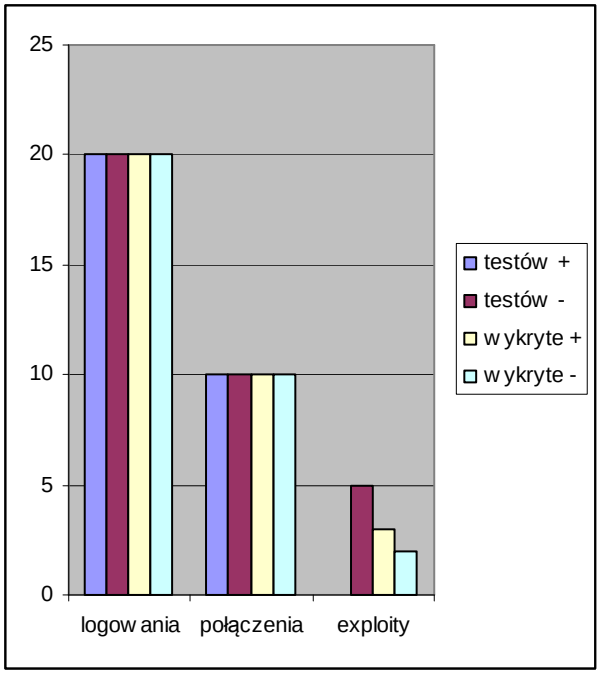
[7] Def.: Odporność – stosunek liczby prób zaburzenia systemu do czasu obsługi procesu klienta.

[8] Połączenie terminalowe do systemów UNIX (szyfrowane).

logowania się do systemu w sposób właściwy oraz sugerujący chęć włamania się na poszczególne konta oraz same fakty wykonywania połączeń do systemu poprawnych i niepoprawnych. Na podstawie zgromadzonych danych w pierwszej kolejności należy stwierdzić, iż sam system usługi SSH dostarcza całkiem dużo informacji niezbędnych do analizy odporności testowanej usługi [8].

Wygenerowano 20 połączeń z prawidłowym logowaniem się na wybrane konta użytkowników systemu oraz dodatkowo 20 prób połączeń na wybrane konta z nieprawidłowymi identyfikatorami, jak również hasłami. Na podstawie zebranych informacji z systemów logów można stwierdzić, że daemon[9] SSH jest w stanie dostarczyć bardzo dokładne informacje o procesach logowania do systemu. Podobną sytuację testową wykonano w przypadku prób nawiązania połączenia, które kończyły się próbą logowania, jak również takie, które nie dobiegały do skutku. Trzecim elementem było sprawdzenie wykrywania połączeń wykonanych przy użyciu istniejącego oprogramowania (exploitów) i w tym przypadku udało się uzyskać rozbieżne informacje od systemu, co widać na zaprezentowanym wykresie. Rozbieżności te wynikały z charakterystyki działania exploitów. Dwa z nich nie spowodowały „właściwych" połączeń do systemu i zostały jedynie zarejestrowane jako próby połączeń do SSH na systemie Firewall (iptables z logowaniem wszystkich połączeń na port 22).

[9] Proces systemowy lub użytkownika działający z odpowiednimi uprawnieniami w celu realizacji określonych zadań.



Rysunek 14 Raport wykrywania zdarzeń SSH

Kolejnym elementem testowanym systemu były usługi FTP[10] (ang. *File Transfer Protocol*), WWW[11] (ang. *World Wide Web*), RDP[12] (ang. *Remote Desktop Protocol*), które w przypadku podobnych testów (bez exploitów) dostarczały również wystarczających danych dla wyznaczenia wartości współczynnika odporności. Testy te obejmowały głównie połączenia do usług oraz stwierdzanie faktów właściwego lub niepoprawnego logowania do systemu.

Na podstawie zgromadzonych informacji przeprowadzono wstępne próby wyznaczenia wartości współczynnika odporności. Wartość tego współczynnika jest miarą ilości nieprawidłowych zdarzeń z punktu widzenia

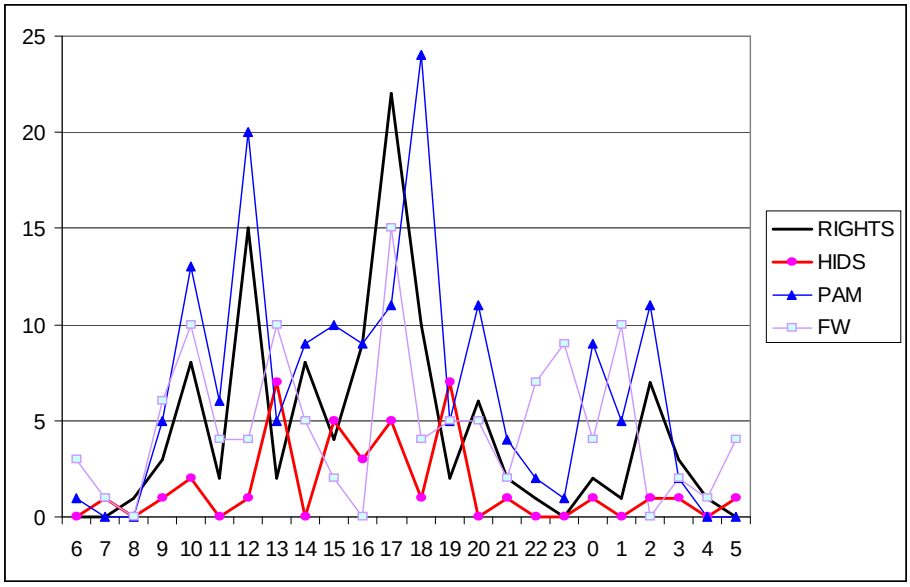
[10] Protokół przesyłania plików.

[11] Przekazywanie informacji graficzno-tekstowej z wykorzystanie protokołu HTTP (ang. *HyperText Transfer Protocol*).

[12] Połączenia terminalowe do systemu Windows.

systemu zabezpieczającego w jednostce czasu. Przeprowadzone badania określiły wartość średnią odporności w przedziale 5-11 zdarzeń na godzinę.

Przedział ten został określony doświadczalnie (rys. 15) poprzez ogólnie dostępne analizatory rejestrów systemowych. Wartości średnie badanych zabezpieczeń (wyniki zaobserwowane w rejestrach) oscylowały właśnie w określonym przedziale i dlatego też autor przyjął prezentowany przedział jako wyjściowy do dalszych badań.



Rysunek 15 Analiza odporności zabezpieczeń

Dane zgromadzone na bazie doświadczeń laboratoryjnych należało potwierdzić w rzeczywistym systemie. I tu okazało się, że o ile jesteśmy w stanie generować różne zdarzenia w warunkach laboratoryjnych, to w systemie rzeczywistym przy dużej rotacji użytkowników sieci (testowana sieć komputerowa składała się z ok. 35 serwerów przeznaczonych do pracy terminalowej, jak i usługowej z kilkoma tysiącami użytkowników) oczekiwane wyniki były bardzo małe. Głównym powodem tego stanu jest

fakt braku aktywności niepożądanej w rozpatrywanym przedziale czasowym. Jednak główny cel został osiągnięty. Udowodniono możliwość wyznaczenia współczynnika odporności na podstawie zebranych informacji w dziennikach systemów sieciowych oraz w laboratoryjnych warunkach możliwość jego przeliczenia [9].

Sam fakt uzyskania ciekawych wyników tylko na podstawie ingerencji własnej w system jest właściwie mało istotny. Ważny jest fakt właśnie możliwości policzenia tego parametru. W chwili obecnej przeliczenia dokonywane były „ręcznie", jednak istnieje możliwość dopasowania do sygnatur pozostawianych w dziennikach systemowych odpowiednich automatycznych narzędzi przeliczających ten współczynnik.

Kolejnym krokiem było przystąpienie do analizy parametru *otwartości*[13] systemów zabezpieczających. Jak zostało to zaprezentowane w poprzednim rozdziale, wyznaczenie tego parametru jest związane z poznaniem struktury informacji przechowywanej przez konkretne rozwiązanie zabezpieczające. W tym przypadku ograniczono się do metody pośredniej pozwalającej skupić się na jednym z dość ważnych podsystemów bezpieczeństwa, jakim jest system uwierzytelnienia. System uwierzytelniający to jeden z podstawowych składników prawie każdego protokołu lub usługi w obecnych systemach komputerowych. Jego rola więc jest dość znaczna w procesie bezpieczeństwa każdego systemu. Nie byliśmy w stanie zaprezentować wielu możliwości z powodu braku dostępu do odpowiednich informacji możliwych do zastosowania w prowadzonym badaniu. Autor zdaje sobie sprawę z bagażu błędu, jakim może być

[13] Otwartość – zależność części zasobu, które na moment t_i są stale dostępne, do całkowitej ilości posiadanych zasobów.

obarczone wyznaczenie potem globalnych wartości, jakimi są *B(S)* czyli bezpieczeństwo systemu oraz *G(S)* czyli poziom gwarancji bezpieczeństwa, jednak wyniki te są jedynie wstępem do dokładnej analizy, która może być prowadzona w dalszych etapach z właściwym uwzględnieniem większej ilości usług bezpieczeństwa po konsultacjach z odpowiednimi producentami danej usługi zabezpieczającej. Ujednolicając wzory 2, 3, 4, 5, 6 otrzymamy postać wzoru:

$$\eta = \alpha * \sum_{i=1}^{n} \left(\frac{\left(N * \frac{\sum_{t} \frac{l_{\max} - l_{\varepsilon_i}}{l_{\max} - l_{\min}}}{\varepsilon} \right)_{t_i}}{\sum_{i=1}^{n(t_i)} P_i} \right)$$

W warunkach laboratoryjnych wykonano testowe badania w celu określenia wstępnych wartości parametru otwartości systemu, a następnie wykonano rzeczywiste pomiary w systemie sieciowym. System ten zawiera 3050 kont użytkowników, z czego 200 kont zawierało domyślne hasło o długości 10 znaków, 2800 kont miało to samo domyślne hasło o długości początkowej 11 znaków. Jest to stan początkowy systemu i dla uproszczenia nie uwzględniono możliwości zmiany haseł. Dla takiego stanu przeanalizujmy w jednym z momentów pracy systemu t_i wartość współczynnika otwarcia, którego parametry miały się następująco:

Tabela 1 Zestawienie parametrów dla wyznaczenia współczynnika otwartości na moment t_i

Całkowita liczba haseł w systemie analizowanym	N	3050
Maksymalna długość hasła rozpoznawana w systemach	l_{max}	14
Minimalna długość hasła możliwa do podania w systemie	l_{min}	3
Średnia długość hasła użytkowników US	$l_{\varepsilon(US)}$	11
Średnia długość hasła użytkowników UD	$l_{\varepsilon(UD)}$	10
Zalogowanych użytkowników klasy US na moment t_i	$\varepsilon(US)$	100
Zalogowanych użytkowników klasy UD na moment t_i	$\varepsilon(UD)$	10
Ilość procesów wykorzystujących uwierzytelnienie	P_i	3
Uśredniona wartość logowań w chwili t_i	$n(t_i)$	55

Przyjęte klasy użytkowników US i UD są to odpowiednio użytkownicy posiadający jako domyślne hasło PESEL oraz użytkownicy z generowanym hasłem o stałej jego długości. Podstawiając odpowiednio do wzorów (2-6) otrzymamy następujące wartości:

Tabela 2 Wyznaczone wartości pośrednie współczynnika otwartości

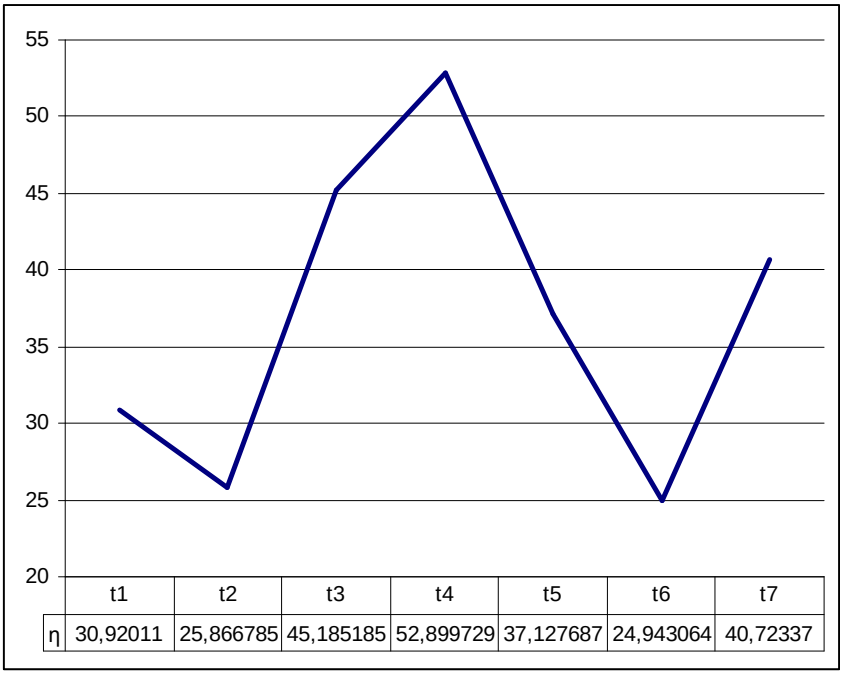
Średnia długość hasła dla klasy US	$\beta_{(US)}$	7,636363636
Średnia długość hasła dla klasy UD	$\beta_{(UD)}$	9,090909091
Wartość współczynnika klasy użytkowników	γ	16,72727273
Całkowita ilość ukrytych komponentów	M_{ti}	51018,18182
Ilość otwartych ukrytych komponentów	m_{ti}	165

Zgodnie z powyższymi wynikami określonymi w tabeli 2 na podstawie założeń z tabeli 1 otrzymamy następującą wartość współczynnika otwartości η na chwilę t_i działania systemu sieciowego:

$$\eta = 30{,}92011[\%]$$

W tym konkretnym przypadku przyjęto współczynnik doprowadzenia stopnia otwartości środka obrony do jednostek pomiaru $a = 0.1$, co zostało ujęte w wyniku prezentowanym powyżej.

Na rysunku 16 przedstawiono zestawienie wyznaczonego parametru otwartości systemu wykonanych w siedmiu kolejnych po sobie przedziałach czasowych momentu t_i rozpatrywanego systemu, przy stałej rotacji współpracujących z systemem użytkowników sieci oraz ich nierównomiernym rozłożeniu (stosunek wartości US do UD), a także stałej liczbie procesów wykorzystujących mechanizm uwierzytelniania i niezmiennej ilości użytkowników.



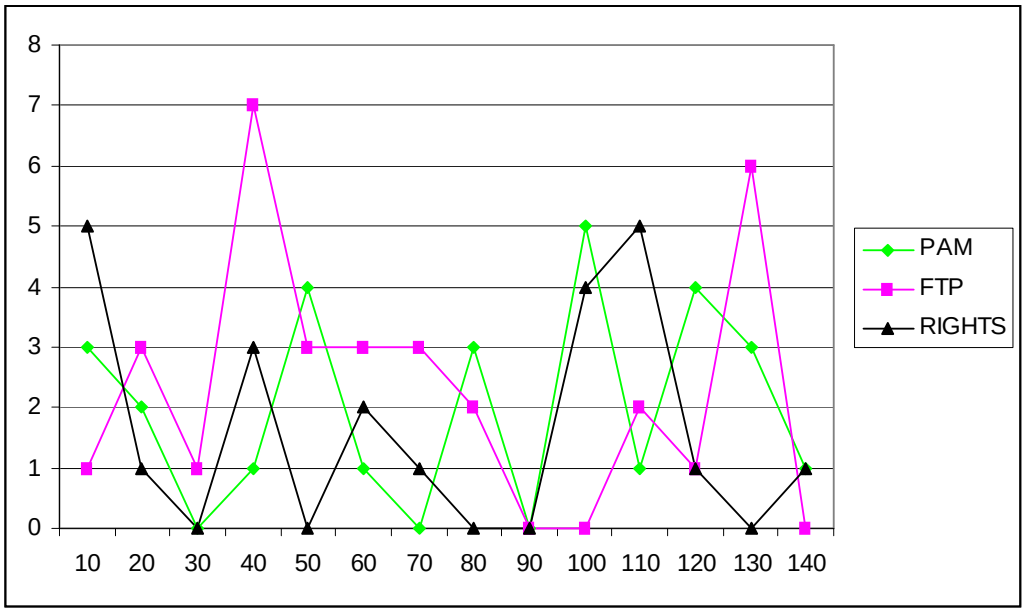
	t1	t2	t3	t4	t5	t6	t7
η	30,92011	25,866785	45,185185	52,899729	37,127687	24,943064	40,72337

Rysunek 16 Wartości współczynnika otwartości w chwilach t_i

Z doświadczeń i badań przeprowadzonych na tym parametrze wynika, że jest on bardzo czuły na zmiany w systemie bezpieczeństwa szczególnie przy wyżej wymienionych założeniach. Nawet bardzo małe

zmiany na przykład w ilości procesów uwierzytelniających lub ilości zgłoszeń logowania do systemu wywołują bardzo burzliwe zachowanie się tego parametru. Niemniej jednak proces wyznaczenia współczynnika otwartości jest w sposób jednoznaczny zależny od budowy wewnętrznej elementarnego modułu bezpieczeństwa, a głównie od ilości przechowywanej (buforowanej) informacji niejawnej niezbędnej w procesie bezpieczeństwa systemu komputerowego [6][8][9][10].

Trzeci z parametrów opisujących bezpieczeństwo systemów informatycznych, jakim jest *przeciążalność*[14] opisuje głównie wydolność przetwarzania informacji w czasie. Współczesne sieci komputerowe narażone są na dość częste ataki typu DoS (ang. *Denial of Service*).



Rysunek 17 Wykorzystanie kanałów obsługi

Blokowaniu poddawane są nie tylko typowe usługi sieciowe: DNS, FTP, SMTP (ang. *Simple Mail Transport Protocol*), ale również usługi

[14] Stosunek dostępnych kanałów usługi do całkowitego zapotrzebowania w czasie.

zabezpieczające [23]. Istnieje zatem zagrożenie, że usługa zabezpieczająca posiadać będzie zbyt mało kanałów obsługi klientów.

Badania w tej kwestii w środowisku doświadczalnym wykazały, że ograniczenia te nie są aż tak duże, jak mogło się wydawać w trakcie precyzowania założeń. Ilość odwołań przypadających na czas obsługi klienta średnio wahał się w okolicy wartości kilku żądań na 10[sekund].

Wartość ta dotyczy większości elementów zabezpieczających, jednak wiele z badanych usług nie podaje maksymalnej ilości kanałów obsługi, co powodowało trudności w oszacowaniu współczynnika z zależności (7).

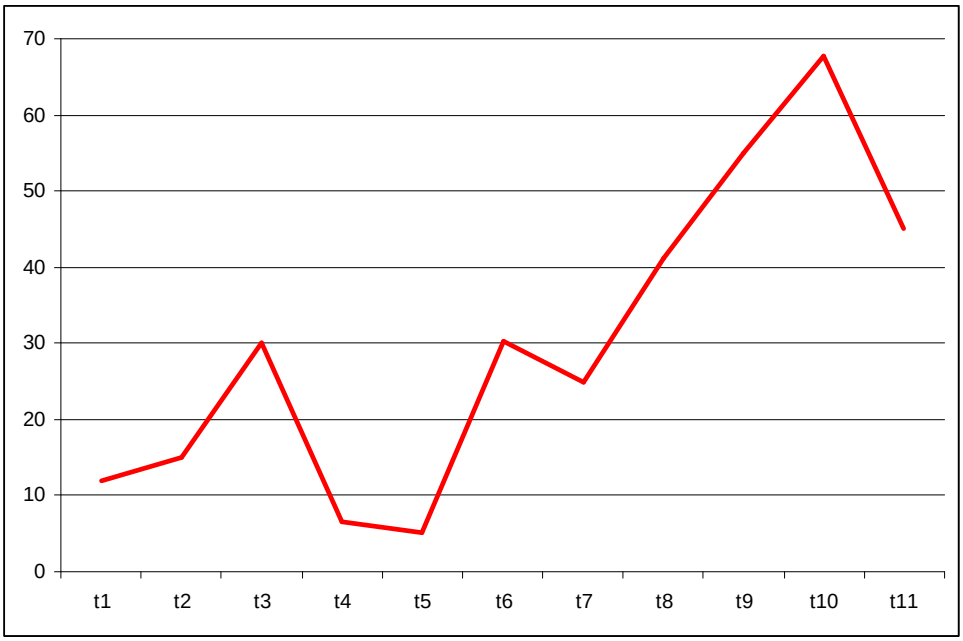
W warunkach rzeczywistych wartości te znacznie uległy zmianie. W systemach mocno uwarunkowanych na połączenia chwilowe (na przykład systemy pocztowe) ilość odwołań jest dość znaczna i przekraczała (w środowisku badanym) kilkadziesiąt do kilkaset na 10-20 sekund pracy badanego serwera. Wykonane badania parametru przeciążalności uwzględniające wartości połączeń zgodnie z rysunkiem 17 i przy poniższych założeniach dały następujące wyniki

Tabela 3 Zestawienie parametrów niezbędnych do wyznaczenia przeciążalności

Ilość funkcjonalnych działań obsługi	n	3
Średni czas obsługi k-tej usługi	$t_i(u_k)$	5
Średnie zapotrzebowanie	z_i	10
Średnia ilość k-tej usługi (zagłębienie, podprocesy)	u_k	2

Dla tak sformułowanych założeń wartość wyznaczonego przeciążenia procesu uwierzytelniania wyniosła $\mu = 12$ [%]. Poniżej na rysunku przedstawiono zebrane dane w procesie działania systemu

zabezpieczanego na podstawie danych dostępnych w dziennikach zdarzeń systemu w założonym przedziale czasowym.

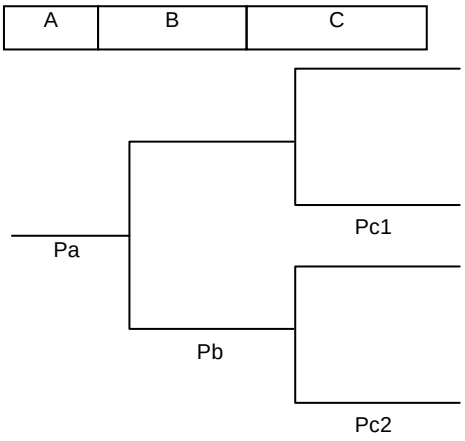


Rysunek 18 Wartości współczynnika przeciążalności w poszczególnych przedziałach czasowych

Badania w systemie rzeczywistym przyniosły kolejne zmiany w poglądach dotyczących istotności tego parametru. Przeciążenia – podobnie jak otwartość – są również dość czułym współczynnikiem. Silnie reaguje na zapotrzebowanie dając natychmiastowe zmniejszenia wartości bezpieczeństwa systemu obrony. Cecha ta jest bardzo mile widziana i w sposób bardzo dobry będzie odzwierciedlać zmiany bezpieczeństwa *B(S)* w czasie rzeczywistej pracy systemu. Dowodzi to słuszności doboru właśnie tych parametrów opisujących badane wartości bezpieczeństwa systemów informatycznych [8][9][10].

4.2. Badanie matematycznego modelu ryzyka

Na podstawie opisywanych drzew przeprowadzono badania rzeczywistego systemu, celem odwzorowania wartości poszczególnych składowych ryzyka systemu. W tym celu stworzone zostało drzewo zdarzeń dla jednego z procesów bezpieczeństwa, jakim jest system uwierzytelniania. Jest to mechanizm pozwalający w sposób dość prosty opisać go i zaprezentować w prowadzonych badaniach. Jest to również mechanizm w większości przypadków zrozumiały i prosty.



Rysunek 19 Drzewo zdarzeń systemu uwierzytelnień

Odpowiednio stany A, B, C zostały przypisane następującym podsystemom bezpieczeństwa: wykrycie podejrzenia włamania (stan A), proces audytu (stan B), weryfikacji haseł (stan C). W każdym ze stanów określono następujące prawdopodobieństwa: *Pa* – prawdopodobieństwo włamania w stanie A, *Pb* – prawdopodobieństwo wykrycia próby łamania haseł w stanie B, Pc_1 – prawdopodobieństwo kolejnych wykrytych błędnych prób uwierzytelnienia następujących w krótkich odstępach czasu po sobie, Pc_2 – prawdopodobieństwo błędnego uwierzytelnienia pomniejszone o czynnik Pc_1.

Prezentowane wyniki badań zostały opracowane z rejestrów systemowych 14 serwerów dydaktycznych zarejestrowanych w ciągu 8 dni tj. od 28.03.2004 do 03.04.2004 roku. Poniżej zostały przedstawione wykresy poszczególnych prawdopodobieństw zgodnie z drzewem na rysunku 19. W podanym okresie stwierdzono zapisy dotyczące następujących usług posługujących się systemem uwierzytelnień: POP3[15] (ang. *Post Office Protocol version 3*), IMAP4[16] (ang. *Internet Message Access Protocol*), Exim[17], ProFTPd[18], SSHd[19]. Zaobserwowano również 9344 odwołania do systemu uwierzytelnień i jednocześnie w tym czasie system zarejestrował 4543 błędne próby uwierzytelnienia. Tak dużą wartość błędów spowodował głównie w tym okresie serwis POP3, co może być błędem użytkownika, albo celową próbą łamania hasła. W tym samym okresie stwierdzono 4 przypadki włamań na podstawie zgłoszeń użytkowników oraz zmian stanu systemu chronionego zarejestrowanych w odrębnych rejestrach zdarzeń systemowych [2][7][8][9][10].

Na podstawie zebranych danych zaobserwowano następujące zależności wpływające na badane parametry i współczynniki:

Tabela 4 Analizowane dane z rejestrów zdarzeń systemowych

Wykryte niepoprawności połączeń w obsłudze usług	***1/n***	4543
Połączenia POP3	***1/n***	2965
Połączenia SSH	***1/n***	420

[15] Protokół i usługa wymiany informacji poczty elektronicznej – odczytywanie wiadomości z przeniesieniem ich na komputer odbiorcy.

[16] Protokół – usługa informacji pocztowej – odczytanie wiadomości na serwerze pocztowym bez ich przenoszenia do stacji odbiorcy.

[17] System pocztowy o nazwie „Exim" – serwer usługi łącznie z elementami zabezpieczającymi.

[18] Serwer usługi FTP.

[19] Serwer usługi połączeń terminalowych SSH.

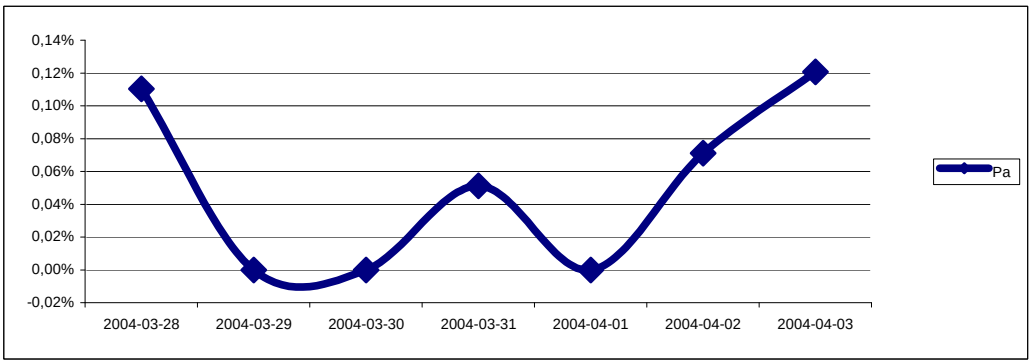
Połączenia FTP	$1/n$	3846
Połączenia SMTP	$1/n$	565
Połączenia IMAP	$1/n$	1109
Wykryte próby naruszenia bezpieczeństwa	$1/n$	111
Udokumentowane włamania	$1/n$	4

Analizując uzyskane dane i przekładając je na drzewo zdarzeń w procedurze uwierzytelnienia użytkownika systemu za pośrednictwem opisywanych wcześniej usług dokonano następujących wyliczeń:

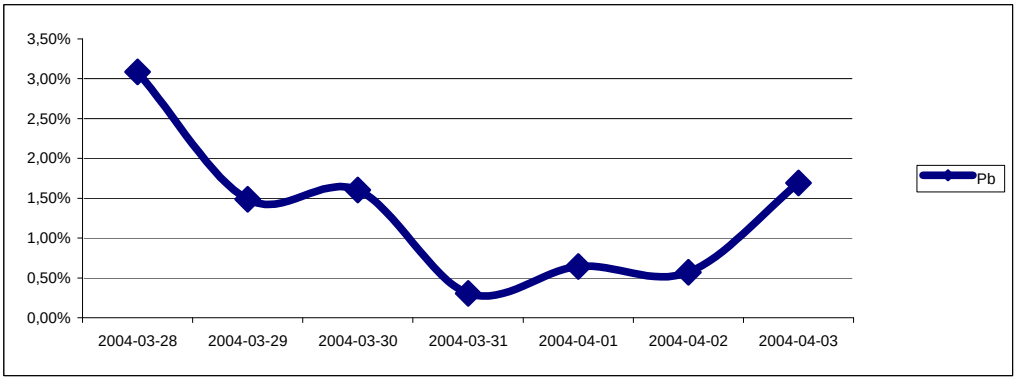
Tabela 5 Wyniki obliczeń wartości pośrednich ryzyka systemu

Całkowite prawdopodobieństwo w stanie A drzewa	P_a	0,0428%
Całkowite prawdopodobieństwo w stanie B drzewa	P_b	1,1879%
Całkowite prawdopodobieństwo w stanie C_1 drzewa	P_{c1}	16,2065%
Całkowite prawdopodobieństwo w stanie C_2 drzewa	P_{c2}	32,4130%
Uśrednione prawdopodobieństwo	Q_i	0,0126%

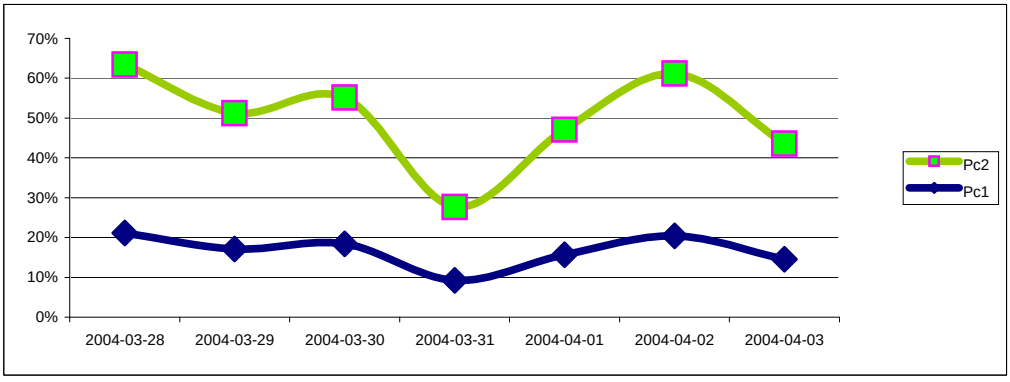
Poniżej zaprezentowano na wykresach wartości składowe poszczególnych prawdopodobieństw w każdym ze stanów rozpatrywanego drzewa zdarzeń. Dane te zostały zgrupowane ze względu na odpowiednie przedziały czasowe, w których dokonano odpowiednich pomiarów.



Rysunek 20 Rozkład prawdopodobieństwa *Pa*



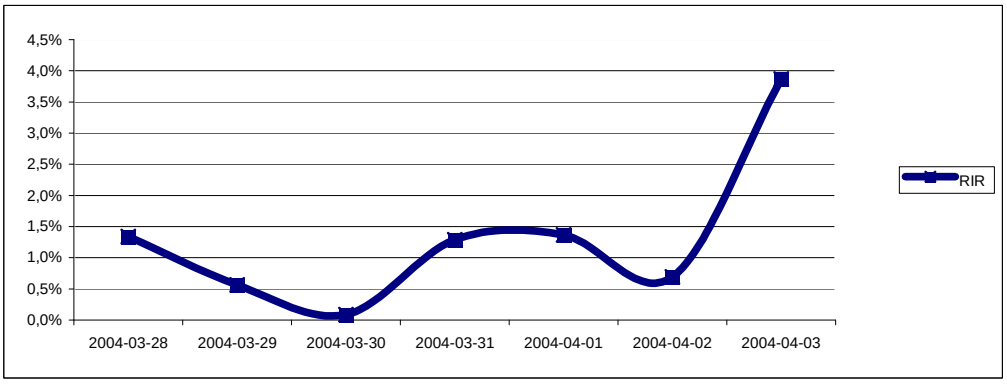
Rysunek 21 Rozkład prawdopodobieństwa *Pb*



Rysunek 22 Rozkład prawdopodobieństw *Pc1* oraz *Pc2*

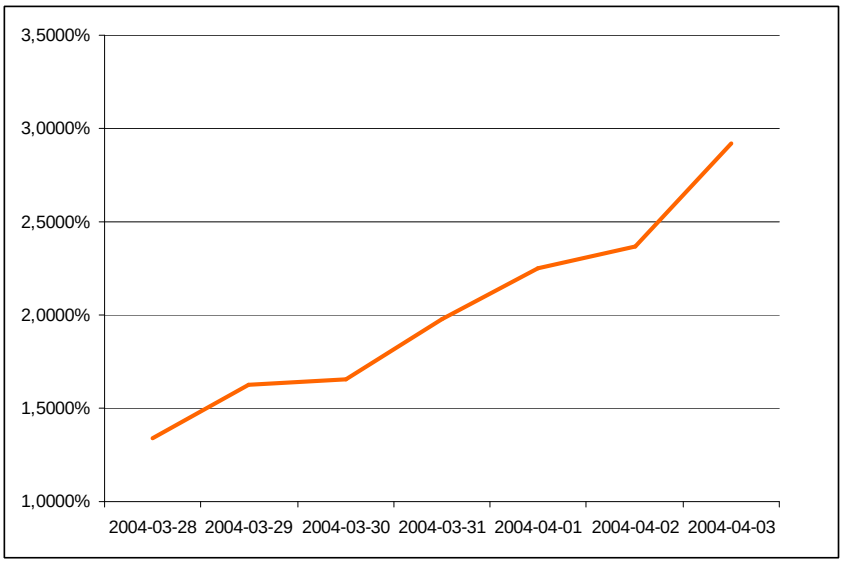
Prezentowane wykresy zostały poddane aproksymacji oraz uproszczono prezentacje danych do zarejestrowanych zdarzeń w ciągu jednego dnia. Informacje uzyskano na podstawie oprogramowania stworzonego samodzielnie do tego celu. Oprogramowanie oraz pozostałe systemy kontroli i analizy dzienników zdarzeń w systemach informatycznych zostało zaprezentowane i omówione w następnym rozdziale.

Na podstawie przeprowadzonych badań wyliczono wartość $RIR(Q_1)$ zgodnie z opisywanymi już wcześniej danymi dotyczącymi zapisów rejestrów systemowych systemu uwierzytelniającego [2][3][4][5].



Rysunek 23 Wartość przyrostu ryzyka *RIR* w czasie działania systemu

Wartość $R(S)_t$ będzie chwilową miarą ryzyka systemu, która po opisaniu na osi czasu pozwoli wyznaczyć zmienność ryzyka elementów systemu bezpieczeństwa dla wszystkich wartości *RIR* w danej jednostce czasu.



Rysunek 24 Prezentacja ryzyka systemu w poszczególnych przedziałach czasowych

Badania ryzyka systemu zostały zaprezentowane jedynie w małym fragmencie analizowanej przestrzeni czasowej ze względu na czytelność formy obliczeń. Prowadzone w tym zakresie badania wykazały fluktuacje krzywej ryzyka w czasie działania i użytkowania systemu w wartościach kilkuprocentowych z losowymi odchyleniami w górę (tj. nagłe zwiększenie obserwowanego parametru), których przypadkowość jest raczej całkowicie spontaniczna. Działania prowadzone w celu ustalenia prawidłowości tych zdarzeń zakończyły się niepowodzeniem, gdyż nie stwierdzono żadnej prawidłowości, która pozwoliłaby przywidzieć kolejne takie skoki. W związku z tym przyjęto, że charakter tych zdarzeń nie może zostać opisany matematycznie i stanowi pewnego rodzaju szum w strukturze analizowanej.

Zaprezentowane wcześniej drzewo dla którego zostały wykonane obliczenia jest tylko wycinkiem dużej zależności, które mogą nie podlegać opisowi na niektórych etapach lub można prościej powiedzieć – mogą Ne

zachodzić w określonym stanie. Przykładem może być cała struktura zależności przedstawiona poniżej.

Przeprowadzone badania pozwoliły wyznaczyć wartości $B(S)_t$ oraz $R(S)_t$, czyli bezpieczeństwa i ryzyka systemu analizowanego w czasie działania tychże systemów. Zostały jednak one wyznaczone przy pomocy autorskich programów[20] oraz systemów detekcji zdarzeń i anomalii. Niestety, zostały one (jak podawano wcześniej) ustalone na podstawie „ręcznych" obliczeń, bez wsparcia zautomatyzowanego oprogramowania.

Doprowadziło to do możliwości wyznaczenia wartości poziomu gwarancji bezpieczeństwa systemu analizowanego.

4.3. Badanie metody oceny poziomu gwarancji bezpieczeństwa poszczególnych fragmentów sieci

Istnieje możliwość wyznaczenia bezpieczeństwa systemu informatycznego *B(S)* oraz wartości ryzyka działania tego systemu *R(S)* na podstawie zależności podanej wzorem (8) i (12). Odpowiednio analizując z dużą częstotliwością próbkowania czasu można przyjąć, że wartością poziomu gwarancji bezpieczeństwa systemu lub sieci *G(S)* będzie różnica pomiędzy wartością bezpieczeństwa *B(S)* i ryzykiem systemu *R(S)*, co wyrażono zależnością opisaną wzorem (13) i przedstawioną na rysunku 13.

Problemem jednak jest to, że wyznaczenie wartości *B(S)* i *R(S)* możliwe jest na podstawie zdarzeń i faktów, które już zaszły. Użytkownikowi systemu informatycznego zależy natomiast nie tylko na zagwarantowaniu mu w czasie jego obecnego działania wystarczającego

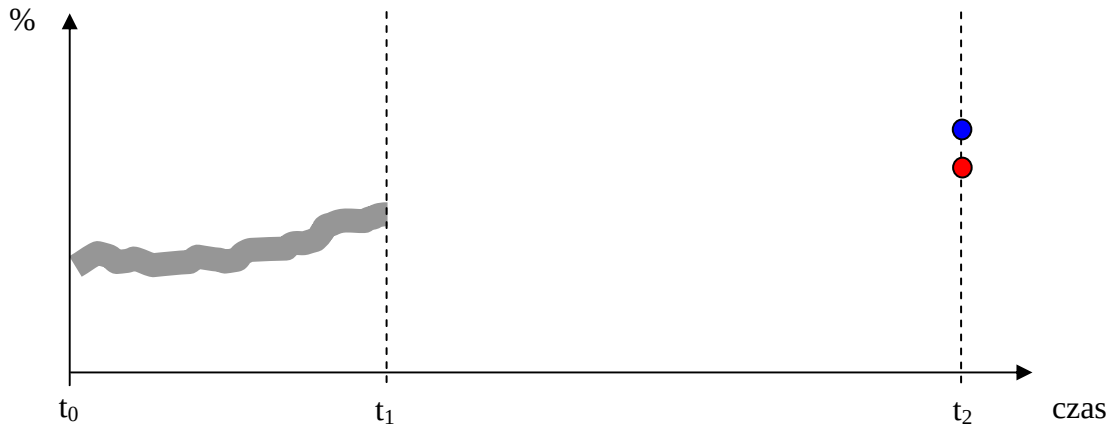
[20] Patrz rozdział 5.

poziomu bezpieczeństwa, ale także bezpieczeństwa podczas jego dalszej współpracy z systemem informatycznym i to z odpowiednim poziomem. Faktem jest, że im dłuższa na przykład transmisja danych przy wykorzystaniu metod bezpieczeństwa takich jak kryptografia, tym większe prawdopodobieństwo ustalenia kluczy szyfrowania transmisji na podstawie jej podsłuchiwania i analizowania. Dlatego też opisywany wcześniej fakt chęci użytkownika systemu na zapewnienie mu odpowiedniego poziomu bezpieczeństwa ma swoje bezpośrednie uzasadnienie w opisywanej metodzie [2][3][4][5].

Stan ten zmusza do wprowadzenia drugiego typu gwarancji, która określi poziom bezpieczeństwa $G_{ps}(S)$ w czasie przyszłym ograniczonym czasem pracy użytkownika w sieci korporacyjnej. Wartości takie można wyznaczać z wykorzystaniem metod statystycznych analizując już posiadane dane z poprzednich okresów działania systemu informatycznego.

System informatyczny można przeanalizować pod kątem bezpieczeństwa i ryzyka na podstawie posiadanych danych z wcześniejszego działania systemu. Założyć należy również, że system działał już wcześniej i prowadzone były odpowiednie obliczenia wartości *B(S)* i *R(S)*. Na podstawie tych wartości można określić rzeczywiste *G(S,t)*.

Zadaniem statystyki będzie ustalenie przybliżonych wartości *B(S)* i *R(S)* w określonym czasie przyszłym odległym od bieżącego punktu pomiarowego o okres ustalony stosownymi przepisami bezpieczeństwa danej korporacji z uwzględnieniem właściwego poziomu istotności dla tego zbioru informacji. Każdy użytkownik może określić niezbędny czas do przeprowadzenia koniecznych czynności z systemem informatycznym, dla którego będziemy prowadzili kolejne analizy. W dalszym rozważaniu zakłada się, że użytkownik określił taki czas i na rysunku 25 reprezentowany jest on przez punkt t_2.

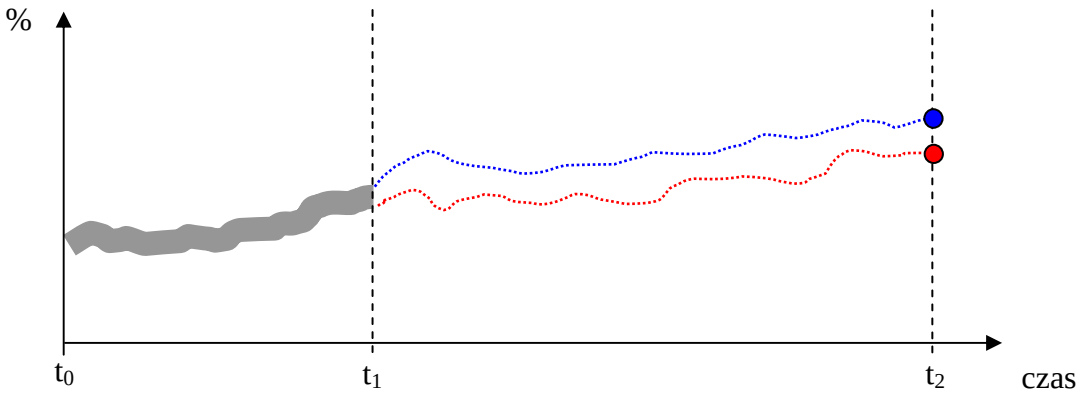


Rysunek 25 Przewidywanie wartości *B(S)* i *R(S)* na chwilę t_2.

Wartości bezpieczeństwa i ryzyka w chwili t_2 wyznaczone metodami statystycznymi posłużą do wykreślenia krzywej pomiędzy punktami w chwili t_1 i t_2. Krzywą wyznaczać będziemy przy pomocy na przykład interpolacji metodą wielomianów. Metodę tę prowadzić będziemy osobno dla składowych poziomu gwarancji bezpieczeństwa *B(S)* oraz ryzyka *R(S)*. Wynikiem przeprowadzonych obliczeń będzie funkcja wielomianowa o stopniu wielomianu *n* równym *N-1* punktów obserwacji każdej ze składowych. Metoda interpolacji wielomianowej jest wyłącznie jedną z rozpatrywanych przez autora i nie jest wyznacznikiem ostatecznym, lecz jedynie propozycją rozwiązania opisywanego problemu.

Określona funkcja pozwoli na wyznaczenie wszystkich pośrednich stanów składowych *B(S)* i *R(S)* w przedziale $<t_1, t_2>$, które to składowe dalej będą obrazować przyszłe, statystyczne zachowanie się systemu informatycznego, czyli zobrazujemy wartość przewidywanego poziomu gwarancji bezpieczeństwa w opisywanym przedziale czasu. Przybliżony obraz opisywanych zachowań przedstawia rysunek 26.

Zastosowanie interpolacji Lagrange'a pozwala na określenie takiej funkcji wielomianowej $W_n(t)$ w przedziale $<t_0, t_n>$ o stopniu $n=N-1$ wartości $B(S)_N$ i $R(S)_N$ takich, że zachodzi zależność (14).



Rysunek 26 Zobrazowanie wyników interpolacji w przedziale $<t_1, t_2>$.

$$W_n(t) = y_0 \frac{(t-t_1)(t-t_2)\ldots(t-t_n)}{(t_0-t_1)\ldots(t_0-t_n)} + y_1 \frac{(t-t_0)(t-t_2)\ldots(t-t_n)}{(t_1-t_0)(t_1-t_2)\ldots(t_1-t_n)} +$$
$$+\ldots+ y_n \frac{(t-t_0)(t-t_1)\ldots(t-t_{n-1})}{(t_n-t_0)(t_n-t_1)\ldots(t_n-t_{n-1})} =$$
$$\sum_{j=0}^{n} y_j \frac{(t-t_0)(t-t_1)\ldots(t-t_{j-1})(t-t_{j+1})\ldots(t-t_n)}{(t_j-t_0)(t_j-t_1)\ldots(t_j-t_{j-1})(t_j-t_{j+1})\ldots(t_j-t_n)} \quad \textbf{(14)}$$

gdzie:

$y_0, y_1 \ldots y_n$ odpowiadają odpowiednio wartościom $B(S)_0$, $B(S)_1$ ... $B(S)_n$ lub wartościom ryzyka $R(S)_0$, $R(S)_1$ ... $R(S)_n$ zależnie od prowadzonej analizy na odpowiednich danych bezpieczeństwa lub ryzyka w systemie informatycznym.

Przyjmując oznaczenie

$$\omega_n(t) = (t-t_0)(t-t_1)\ldots(t-t_n) \quad \textbf{(15)}$$

możemy wzór (14) zapisać w postaci:

$$W_n(t) = \sum_{j=0}^{n} y_j \frac{\omega_n(t)}{(t - t_j)\omega_n'(t_j)} \qquad \textbf{(16)}$$

gdzie:

$y_j = y(t_j)$, a $\omega_n'(t_j)$ jest wartością pochodnej wielomianu $\omega_n(t)$ w punkcie t_j (będącym zerem tego wielomianu).

Stosując wzory prezentowane powyżej oraz opisywaną jedną z dostępnych metod możliwe jest wyznaczenie funkcji wielomianowych, które opiszą przebieg zmian wartości *B(S)* i *R(S)* dając jednocześnie możliwość wyznaczenia poziomu gwarancji bezpieczeństwa *G(S)* w dowolnym skończonym przyszłym czasie działania systemu informatycznego. Czas przyszły, do którego końca zależy nam na wyznaczeniu gwarancji, nie jest wartością nieskończoną, ponieważ możliwe jest wyłącznie wyznaczenie wartości prawidłowych w czasie proporcjonalnym do czasu wcześniejszych rzeczywistych pomiarów.

4.4. Podsumowanie

Analiza zdarzeń i wyznaczanie wartości poszczególnych parametrów w czasie rzeczywistym zmienia całkowicie podejście do opisu systemu sieciowego i jest rozwiązaniem nowym. Możliwe jest to wyłącznie przez zastosowanie opisywanych w rozdziale trzecim metody przetwarzania informacji o bezpieczeństwie i ryzyku systemów informatycznych. Zaproponowana metoda (szczegółowo analizowana we wnioskach niniejszej rozprawy) pozwala na analizowanie opisywanych w pracy parametrów w czasie rzeczywistego działania systemu. Zaprezentowane wyniki obliczeń zostały uzyskane na bazie analizy i symulacji w rozwiązaniu laboratoryjnym sieci z porównaniem jej do rzeczywistej sieci komputerowej.

Trudno jednak porównać uzyskane wyniki do propozycji obecnie stosowanych prób określenia bezpieczeństwa lub ryzyka sieci. Trudność ta wynika głównie z powodu określania wartości bezpieczeństwa i ryzyka na podstawie statycznych norm lub subiektywnej oceny osób zajmujących się tą tematyką. Dane prezentowane w niniejszej rozprawie są uzyskiwane na bieżąco z rzeczywiście działających systemów i nie mają nic wspólnego z subiektywną oceną przez człowieka tych wartości. Stwierdzić można obecnie, że te nowatorskie podejście do problemu sieci i jej bezpieczeństwa nie idzie w parze z dostępnością informacji niezbędnej do wyznaczenia składowych czy to bezpieczeństwa, czy ryzyka. Autor stwierdził duże braki w informacji dostarczanej do dzienników zdarzeń w systemie, które nie pozwalają na w pełni automatyczne (na obecnym stanie rozwoju systemów sieciowych) wyznaczenie opisywanych wartości. Należy wprowadzić nowe zasady definiowania informacji gromadzonych w dziennikach systemowych oraz zapisać na przykład w ramach standardu podstawowe parametry elementów bezpieczeństwa, które powinny być dostępne systemowi monitorowania. Po wprowadzeniu takich modyfikacji wykonanie procesu automatycznego wyznaczania opisywanych w tym rozdziale wartości i parametrów będzie całkowicie możliwe.

5. Realizacja oddzielnych fragmentów systemu dynamicznego zabezpieczenia określonego poziomu bezpieczeństwa

5.1. Aspekty bezpieczeństwa obecnych systemów informatycznych

W pierwszym etapie formalizowania założeń do postrzegania sieci jako heterogenicznej należy określić podstawowe cechy takiej sieci. Jedną z podstawowych cech jest zróżnicowanie systemów operacyjnych, platform sprzętowych oraz mediów transmisyjnych [11][12][13]. Najczęściej wraz z pierwszą cechą rozpatrywana jest łącznie druga tj. ujednolicony system kont dostępowych. Pozostałe dwie cechy takie jak media transmisyjne są zazwyczaj pomijanym zagadnieniem wraz z różnicowaniem platformy sprzętowej. W dalszej części zaprezentowane zostaną przykładowe problemy, które mogą wynikać z nieuwzględnienia ostatnich dwóch cech w opisie aspektów bezpieczeństwa sieci oraz jej słabych punktów.

Postarajmy się opisać założenia do podstawowej cechy takich sieci. Różnice systemów operacyjnych najczęściej uwypuklają się w powiązaniu z cechą trzecią tj. systemem kont systemowych oraz metodami uwierzytelnienia i autoryzacji. W większości przypadków mamy możliwości wykorzystania usług katalogowych (np. eDirectory firmy Novell, ActiveDirectory firmy Microsoft, oraz otwarte systemy tj. Open LDAP itp.). Usługi katalogowe pomagają w przejrzystym opisaniu obiektów sieciowych oraz ich prawidłowym usytuowaniu w hierarchii firmy. Jednak możliwości wykorzystania wielu rozwiązań pociągają za sobą niedostatek formalnego zapisywania własności o każdym obiekcie w inny sposób przez

każde rozwiązanie. Przykładem może być różnorodność wykorzystania metod kryptograficznych w różnych rozwiązaniach.

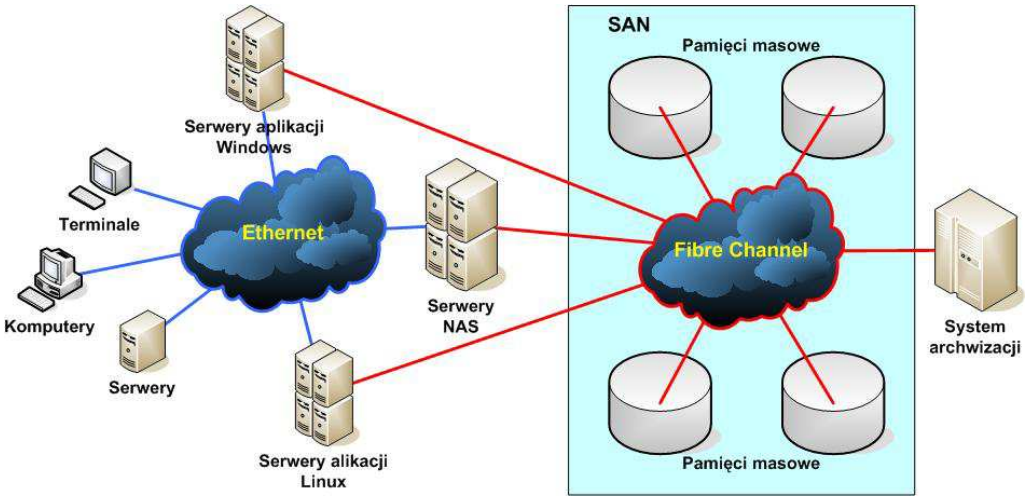
W związku z powyższym musimy rozważyć dwie możliwości: dopuszczamy możliwość wykorzystania jednego preferowanego rozwiązania (np. ActiveDirectory) i dopasowanie innych systemów do tej metody, lub dopuszczenie wykorzystania natywnych (naturalnych dla danego systemu operacyjnego) metod przez wszystkie systemy wchodzące w skład sieci heterogenicznej i dopasowanie metody sprzęgającej do tych rozwiązań [18][26][29][[35].

W drugim przypadku powstaje problem wyboru metody do łączenia różnych środowisk usług katalogowych w jedną spójną całość. Istnieją dwa rozwiązania. Najstarsza metoda to synchronizacja online lub offline na podstawie kopiowania istotnych elementów procesu uwierzytelnienia (np. dla systemu Linux plików passwd i shadow), lub procesów opierających się na zdalnym wykorzystaniu tej informacji z baz danych, jakimi są na przykład usługi katalogowe. Problem takiej synchronizacji i procesu uwierzytelnienia polega głównie na tym, że w przypadku braku dostępu do centralnej składnicy informacji uwierzytelniającej cała struktura nie jest w stanie przeprowadzić procesu uwierzytelnienia, a w przypadku procesów offline mamy do czynienia z nieaktualnymi na czas danymi.

Lepszym rozwiązaniem byłoby wykorzystanie natywnych metod uwierzytelnienia w taki sposób, aby awaria centralnej składnicy nie powodowała zachwiania procesów wewnątrz struktury heterogenicznej. Powstało rozwiązanie pozwalające w trybie online przenosić i uaktualniać wszystkie niezbędne własności obiektów sieciowych w taki sposób, by każdy z systemów operacyjnych posiadał je autonomicznie, jednak z możliwością stałej synchronizacji w przypadku zmian. Opisywane

rozwiązanie przygotowała firma Novell w produkcie Identity Manager [11][12].

Wybraliśmy już sposób połączenia logicznego – pozostał nam aspekt fizyczny. W tym przypadku mówimy o połączeniach



Rysunek 27 Logiczny schemat połączeń sieci rozpatrywanej

międzysystemowych, a dokładniej między poszczególnymi jednostkami wchodzącymi w skład danej sieci. W systemach rozproszonych z jednym kontem istnieje konieczność dostępności danych kluczowych użytkownika (jego dokumenty, klucze i prywatne dane oraz profile) z każdej z wybranych platform i środowisk systemów operacyjnych. Często ten problem (bo właśnie taki istnieje) jest pomijany i nie rozważany. Problem ten polega głównie na bezpiecznym przyłączeniu danych kluczowych użytkownika w taki sposób, by były one dostępne wyłącznie dla niego, niezależnie od wybranej w danym momencie platformy systemu operacyjnego. Do tego celu mamy możliwość wykorzystania trzech najczęściej spotykanych protokołów wymiany danych: NFS[21] (ang. *Network File System*), CIFS[22] (ang.

[21] Specjalny system plików, umożliwiający w środowiskach uniksowych dostęp do zdalnych katalogów i plików.

Common Internet File System), NCP[23] (ang. *NetWare Core Protocol*). Każdy z nich w zależności od miejsca położenia danych (wybór głównej platformy NAS (ang. *Network Attached Storage*), lub SAN (ang. *Storage Attached Network*)) będzie narzucał pewne możliwe do zastosowania rozwiązania. Systemy przechowywania i udostępniania danych muszą w tych przypadkach korzystać również z centralizowanej bazy informacji o użytkowniku lub wykorzystywać zaufane połączenia do innych systemów. Głównie w przypadku wykorzystania zaufania niezbędne jest korzystanie na przykład z wydzielonych separowanych sieci LAN.

Jednym z najbardziej oczywistych typów zagrożeń są wszelkiego rodzaju udogodnienia dla użytkowników sieci. Podstawowym udogodnieniem centralizacji zasobów (w tym użytkowników) jest jedno konto (identyfikator) oraz przypisane mu na przykład hasło. Fakt ten stanowi podstawowe zagrożenie dla bezpieczeństwa sieci heterogenicznej, jak również wszystkich sieci opartych na tym trybie zabezpieczenia. Złamanie jednego hasła powoduje możliwość skorzystania w sposób nieuprawniony z całej gamy systemów operacyjnych i usług, gdzie użytkownik miał dostęp na podstawie swojego identyfikatora i zabezpieczenia dodatkowego, jakim było hasło. W większości przypadków fakt możliwości przełamania hasła można poprawić lub uniemożliwić poprzez wprowadzenie polityki bezpieczeństwa, jaką może być konieczność okresowej zmiany hasła. Niemniej jednak i tu możemy natrafić na efekt wręcz odwrotny. W potocznym żargonie informatycznym nosi to nazwę „efektu nadgorliwego administratora" lub „wirusa administratora".

---

[22] System rozproszonych plików i urządzeń drukujących używany przez sieci firmowe (Microsoft), w szczególności przez sieci pracujące pod systemem operacyjnym Windows NT i korzystające z protokołu TCP/IP.

[23] Protokół jądra NetWare oraz dostępu do danych udostępnianych za pośrednictwem woluminów

Polega to na tym, że pozorne wprowadzanie coraz to nowych i bardziej wyrafinowanych metod zabezpieczeń powoduje teoretyczny efekt zwiększenia bezpieczeństwa systemu, jednak faktycznie obniża jego walory. Dobrym przykładem może być możliwość wprowadzenia wymogu (np. ministerialnego) zmiany hasła co 30 dni oraz dodatkowo niemożliwości powtórzenia poprzednio używanych (historia haseł na przykład 15 wstecz) i konieczności nie słownikowego i zabezpieczonego znakami specjalnymi hasła. Wprowadza to oczywiste polepszenie walorów jakości hasła używanego przez użytkownika, jednak faktycznie zmusza użytkownika do zapisywania tego hasła (w najmniej oczekiwanym miejscu – karteczka na monitorze), ponieważ nie jest w stanie go zapamiętać.

Kolejnym problemem tak zorientowanych sieci jest przestrzeń NAS lub innego rozwiązania składowania danych użytkownika. Dane kont gromadzone są na jednej przestrzeni fizycznej w sposób, który powinien zapewnić prywatny dostęp do określonych zasobów. Systemy operacyjne jednak z domysłu niekoniecznie zapewniają taką strukturę. Przykładem może być możliwość włączenia dla użytkowników sieci publikowania stron własnych z katalogów domowych, co wymaga prawa czytania dla wszystkich katalogu z zawartością strony. Jeżeli połączymy to na przykład z usługą przetwarzania języków skryptowych PHP[24] (ang. *Personal Home Page* lub *Personal Hypertext Preprocessor*), otrzymamy tykające zagrożenie przecieku danych do sieci WEB oraz innych użytkowników tego systemu. Możliwe jest to dzięki temu, że w sposób standardowy system klasy Unix posiada ustawioną maskę tworzenia plików z opcją praw czytania dla wszystkich. Prosty skrypt PHP może w taki sposób odczytać dane z innych kont ponieważ z domysłu uprawnienia systemu na to pozwalają.

[24] Alternatywna wobec ASP i CGI. Technika dynamicznego tworzenia stron WWW.

Oczywistym jest, że każdą niedostateczność systemu można zabezpieczyć w sposób wystarczający w danej chwili, jednak tworzą się ciągle nowe możliwości i to, co teraz jest wystarczające, jutro nie musi być tak samo dobre. Innym przykładem jest popularnie wykorzystany system NFS wersji 3, którego jedynym poświadczeniem możliwości wykorzystania udostępnionego zasobu jest adres IP. Możemy powiązać ten fakt z dostępnymi metodami ataku dość powszechnie używanymi, jakimi są ataki typu DOS lub dDOS i wynikającej z tego faktu możliwości przejęcia w dość prosty sposób wszystkich udostępnianych tak informacji.

Takie przykłady zapewne można mnożyć w nieskończoność, jednak nie w tym rzecz. Należy zdawać sobie sprawę z istniejących zagrożeń, które w każdym rozwiązaniu mogą być inne i specyficzne dla danego wdrożenia. Najczęstszym błędem administratorów jest zaufanie do użytkowników i bagatelizowanie zgoła mało istotnych zagrożeń. Należy przy każdym wdrożeniu sprecyzować oczekiwania stawiane takiej sieci i rozwiązaniom w niej działającym, co powinno doprowadzić do skrupulatnego przeanalizowania możliwych zagrożeń oraz próby ich wyeliminowania.

W obecnych czasach w dziedzinie zabezpieczeń systemów informatycznych ugruntowały się dwa pojęcia: bezpieczeństwo systemu oraz ryzyko systemu. Ich znaczenie przeplata się ze sobą i w niektórych przypadkach ciężko o ich rozgraniczenie. Aspekt bezpieczeństwa jest bardziej „namacalny", niemniej jednak do tej pory nie udało się sprecyzować dokładnych jego parametrów. Obecnie opiera się on na dwóch trendach.

Pierwszy z nich polega na wprowadzeniu wartości liczbowej każdej z opisywanych standardem norm, pozwalających na odpowiednie zabezpieczenie systemu i następnie wyliczeniu wartości uśrednionej określającej bezpieczeństwo danego systemu chronionego. Wartość ta jest

niestety uzyskana w sposób subiektywny, ponieważ wartości liczbowe otrzymywane są na podstawie ankiety przeprowadzanej w środowisku sieciowców. Drugi problem stanowi statyczność tej metody, ponieważ jest ona wyznaczana jednokrotnie i prawie wyłącznie przy tworzeniu systemu i na podstawie założeń. W większości przypadków może się okazać, że proponowane zabezpieczenia nie są adekwatne do chronionych informacji.

Druga metoda polega na określeniu istotnych parametrów bezpieczeństwa i pomiar tych parametrów bądź każdorazowo podczas testowania zabezpieczeń, bądź na bieżąco w czasie rzeczywistym. Głównym niedostatkiem tej metody jest niewystarczalność informacji zapisywanej i dostarczanej przez systemy, które chronimy. Często brakuje podstawowych danych, pozwalających określić wymagane parametry. Jest jednak również zaleta takiej metody. Przy pomocy badania parametrów bezpieczeństwa w czasie działania systemu i gromadzenia w nim danych można przewidywać trendy wahań wartości bezpieczeństwa w czasie przyszłym lub zdefiniowanym przez użytkownika czasie działania z systemem zabezpieczonym. Daje to możliwość odpowiedniego poziomu gwarancji bezpieczeństwa w określonym czasie działania z systemem [11][12[13].

Wszyscy jednak są zgodni, że wartość bezpieczeństwa jest składową wszystkich czynników bezpieczeństwa, czyli sumą najmniejszych komórek bezpieczeństwa systemu. W procesie obliczania nie można jednak pozwolić sobie na pominięcie nawet najbardziej prozaicznego zabezpieczenia.

5.2. Analiza zaproponowanych metod za pomocą dostępnych narzędzi

W systemach informatycznych możemy napotkać szereg różnego rodzaju mechanizmów pozwalających na monitorowanie zachowania się

systemu. Do takiego rodzaju rozwiązań należy zaliczyć daemony składowania informacji w wyniku działania poszczególnych usług sieciowych. W rozpatrywanych badaniach wykorzystany został system składowania logów o nazwie „syslog-ng". Jego podstawowa konfiguracja została zlokalizowana w folderze `/etc/syslog-ng/`[25]. Zawartość konfiguracji pozwala na ustawienie tego daemona na pracę w trybie lokalnym z wieloma odrębnymi składnicami informacji oraz można go dowolnie dostosować do potrzeb. Pozwala on również na dołączanie dodatkowego oprogramowania pośredniczącego, które może dopasowywać informacje napływające od usług i dokonywać wstępnej obróbki danych [103].

Gromadzone dane podlegały przekształceniu przy pomocy własnego i już istniejącego oprogramowania, które pozwalało wydobyć interesujące informacje. W badaniach wykorzystano również tablicę informacji poszukiwanych, czyli ciągu słów lub pojedynczych wyrazów pozwalających opracować niezbędne wyniki. Frazy zaprezentowano w tabeli 6. Zestawione tam frazy zostały również oznaczone pod kątem ich ważności. Wartości poniżej zera zostały potraktowane jako informacje pomocnicze w głównej mierze nie wpływające na działanie i wartości bezpieczeństwa lub ryzyka systemu rozpatrywanego.

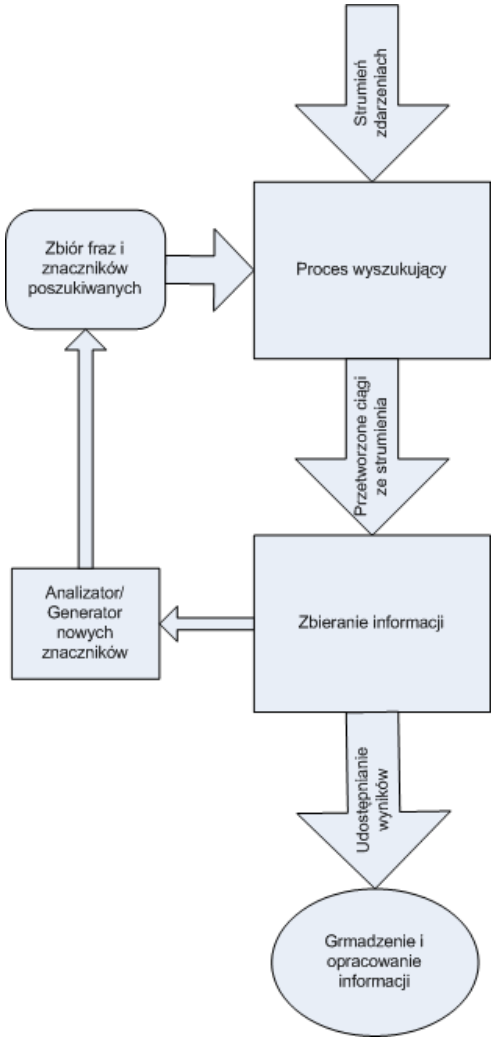
Tabela 6 Zestawienie badanych fraz w ciągu logów systemowych

Fraza	**Typ**	**Opis**
date	*1*	Dates and times
host	*2*	Host names and IP numbers
process	*0*	Sender process
pid	*-1*	PIDs (Process IDs)
pid-sqbr	*0*	Brackets around PIDs

[25] Przykład konfiguracji w dodatku A – patrz Konfiguracja syslog-ng.

default	*1*	Default (not colorised)
email	*0*	E-mail addresses
subject	*3*	Subject lines (procmail)
dir	*1*	Directory names
size	*-1*	Sizes
user	*4*	Usernames
httpcodes	*-1*	HTTP status codes (200, 404, itp.)
getsize	*3*	Transfer sizes
get	*0*	HTTP GET
post	*0*	HTTP POST
head	*0*	HTTP HEAD
put	*0*	HTTP PUT
connect	*0*	HTTP CONNECT
trace	*0*	HTTP TRACE
unknown	*1*	Unknown message
gettime	*3*	Transfer times
uri	*0*	URIs (http://, ftp://, etc)
ident	*-1*	Remote user (proxy/http)
ctype	*-1*	Content type (http/proxy)
error	*5*	Error messages
miss	*5*	Proxy MISS
hit	*4*	Proxy HIT
deny	*5*	Proxy DENIED
refresh	*-1*	Proxy REFRESH
swapfail	*-1*	Proxy SWAPFAIL
debug	*-1*	Debug messages
warning	*5*	Warnings
direct	*-1*	Proxy DIRECT
parent	*4*	Proxy PARENT
swapnum	*2*	Proxy swap number
create	*-1*	Proxy CREATE
swapin	*-1*	Proxy SWAPIN
swapout	*-1*	Proxy SWAPOUT
release	*-1*	Proxy RELEASE
mac	*-1*	MAC addresses
version	*-1*	Version numbers
address	*-1*	Memory addresses
numbers	*-1*	Numbers
signal	*4*	Signal names
service	*3*	Services

prot	*3*	Protocols
bad	*4*	"Bad words"
good	*1*	"Good words"
system	*1*	"System words"
incoming	*-1*	Incoming mail (exim)
outgoing	*-1*	Outgoing mail (exim)
uniqn	*-1*	Unique ID (exim)
repeat	*-1*	'last message repeated N times'
field	*1*	RFC822 Field
chain	*1*	Chain names (ulogd)
percentage	*4*	Percentages
ftpcodes	*1*	FTP codes
keyword	*4*	Various keywords (like PHP in php.log, itp.)



Rysunek 28 Schemat działania wyszukiwacza

Wszystkie wartości większe od zera oznaczają istotność z punktu widzenia autora i są wartościami subiektywnymi nadanymi w drodze eksperymentów oraz istotności ich występowania [126][130].Do wyznaczenia większości współczynników przydatny okazał się wyszukiwacz własnego pomysłu gromadzący dane w układach godzinnych oraz dniowych.

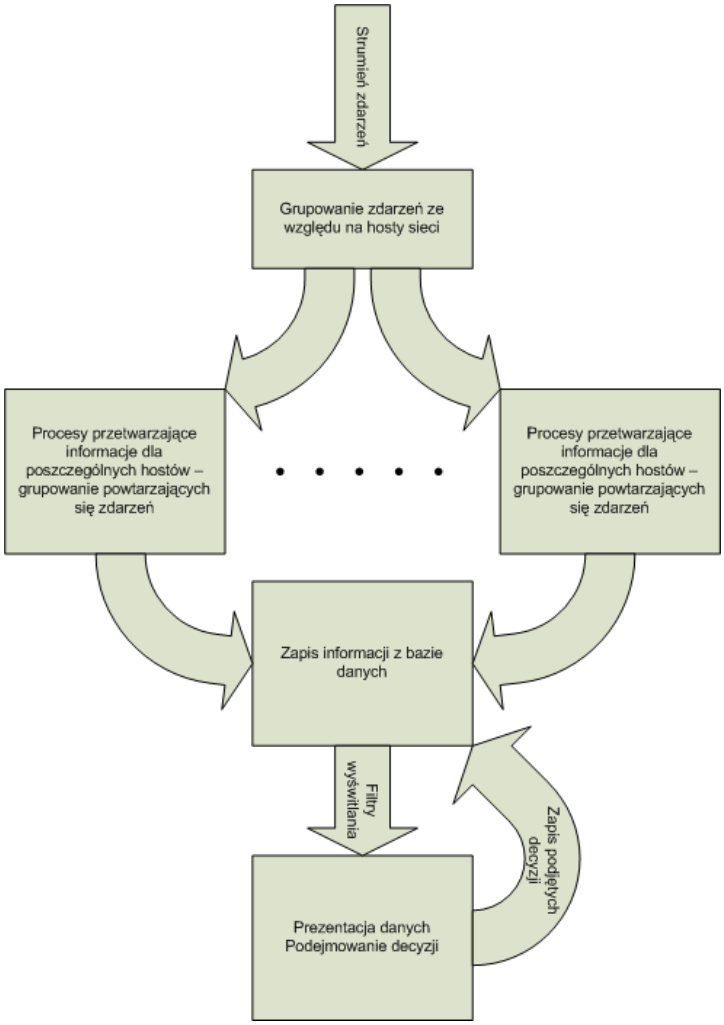
Głównym jego zadaniem jest wyszukiwanie występowania określonych fraz poszukiwanych w systemowych logach (dziennikach zdarzeń) oraz reprezentacja wyników do dalszej obróbki. Algorytm działania

przeszukiwacza został zaprezentowany na rysunku 26, a jego kod w dodatku A w części „Kody programów i wyniki przetwarzania danych". Wyniki dostarczane przez oprogramowanie reprezentowane są wyłącznie w formie ilościowej, ponieważ taka forma była wymagana do wyznaczenia parametrów bezpieczeństwa, jak również ryzyka systemu rozpatrywanego w badaniach laboratoryjnych.

Kolejnym etapem badań było skonstruowanie pakietu oprogramowania pozwalającego na dopasowanie możliwości percepcyjnych człowieka do ilości napływających danych. Jego głównym elementem jest uporządkowanie informacji poprzez grupowanie podobnych zdarzeń (na przykład pochodzących z jednego hosta w sieci) oraz prosta ich prezentacja. Pakiet posiada następujące założenia:

- uporządkowanie (grupowanie) informacji i gromadzenie jej w tablice zgodne z nazwami hostów sieci korporacyjnej,
- zatwierdzanie rejestrów zdarzeń przez wszystkich użytkowników systemu z możliwością określenia priorytetów,
- ignorowanie zdarzeń podobnych o niskim priorytecie na podstawie jednorazowego zatwierdzenia przez operatora systemu i zastosowanie odpowiednich filtrów.

Te podstawowe funkcje eliminują natłok informacji płynących z dzienników systemowych. Natłok informacji może powodować chaotyczność w działaniu operatora lub powodować „nieczułość" na ważne i często powtarzające się przypadki zdarzeń niepożądanych [2][4][5][6].



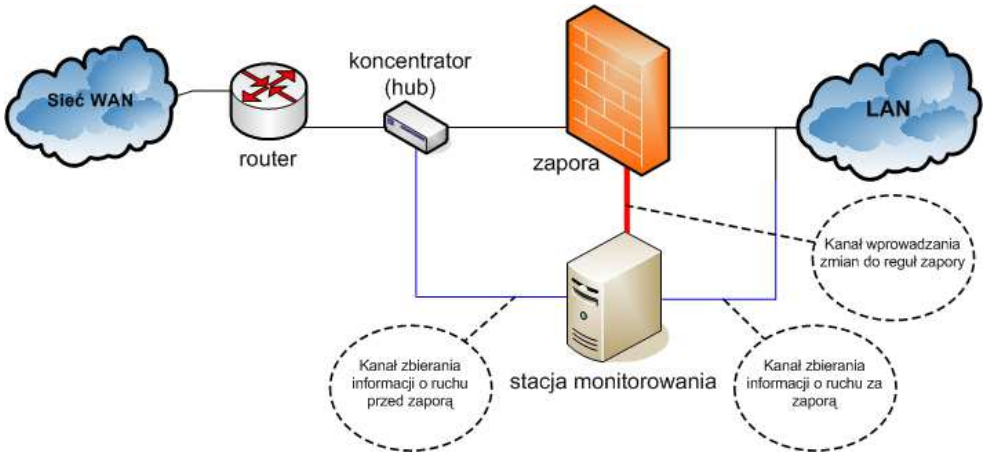
Rysunek 29 Schemat przepływu informacji w procesie decyzyjnym

System został opracowany do współdziałania z różnymi systemami operacyjnymi i z założenia miał być od tych systemów niezależny. Został napisany w językach skryptowych i jest prosty do przystosowywania. Prezentacja graficzna możliwa jest w oparciu o przeglądarki webowe. Przedstawiony na rysunku 29 schemat procesu działania filtrów oraz procesu decyzyjnego opiera się na zestawach filtrów porównawczych, które wyszukują w ciągu informacji o zdarzeniu podobne informacje. Ciąg ten zawiera informacje czasowe, dane o publikującym je hoście, informacje o procesie, który generuje informacje oraz samą treść informacji.

W obecnym czasie problemem jest uproszczenie i grupowanie samej informacji. Rozmaite procesy prezentują różnego typu informacje o swoim działaniu. Nie istnieją w tym zakresie żadne standardy pozwalające na opisanie zasad publikowania informacji o zdarzeniu. Fakt ten nastręcza automatom analizującym, jak i samemu operatorowi dość dużo problemów. Zdarza się niejednokrotnie, że ta sama pod względem ważności i treści informacja jest różnie prezentowana, zależnie od systemu operacyjnego. Innym problemem są identyczne informacje opisujące zdarzenia dla różnych stanów bezpieczeństwa tego samego procesu. Przykładem może być informacja podawana przez proces SSH o treści „possibile break attack", która odnosi się zarówno do próby włamania polegającej na podszyciu się, jak również nieprawidłowym zakończeniu się procedury połączenia z wykorzystaniem protokołu SSH2.
Przygotowany pakiet pozwala jednak na uproszczenie procesu decyzji na przykład administratora sieci, a także ułatwia identyfikację samych zdarzeń.

Innym przykładem praktycznego wykorzystania zaproponowanej wiedzy jest pakiet programowy pozwalający na dokonywanie zmian w ścianie ogniowej (ang. *firewall*) podczas wykrycia nieuprawnionej komunikacji lub nawet próby jej wykonania [110][112].
Główna zasada działania opiera się o wprowadzenie pętli zwrotnej w procesie kreowania reguł sterowania ścianą ogniową (zaporą), tak by efektywnie wpływać na bezpieczeństwo w czasie działania.

Jest to właśnie element odróżniający go od definiowanych statycznie informacji dla zapory. System działa w pełni automatycznie i może być dodatkowo „motywowany" przez operatora sieci.



Rysunek 30 Schemat działania sprzężenia zwrotnego dla zapory

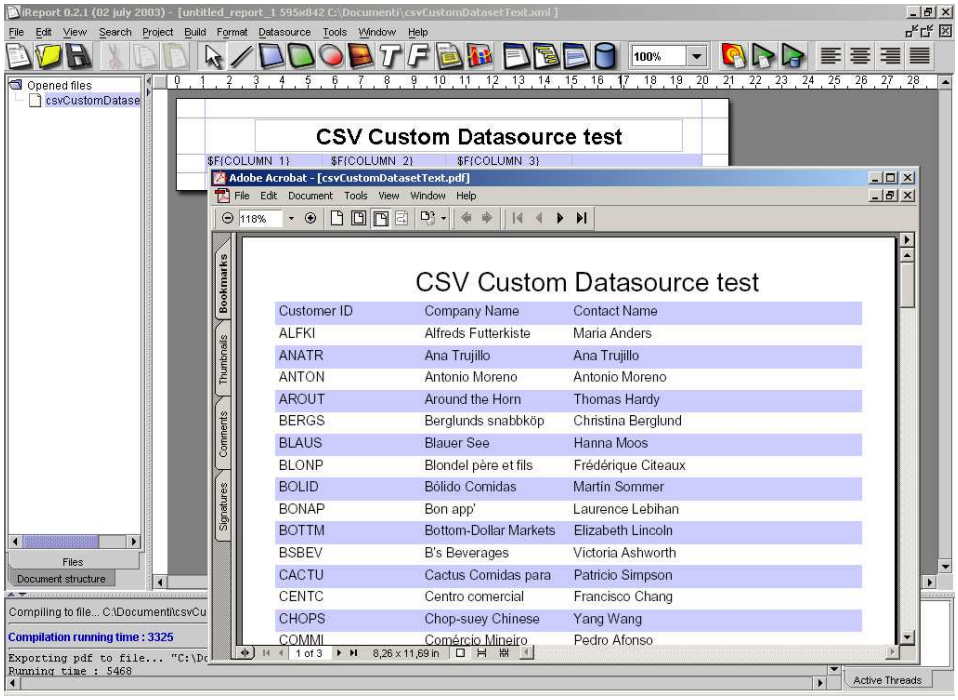
Na rysunku 30 przedstawiono schemat funkcjonalny działania pakietu programowego. Punktem decyzyjnym jest „stacja monitorowania", która została wyposażona w oprogramowanie współpracujące ze znanym pakietem wykrywania anomalii, jakim jest SNORT. Pakiet SNORT wykorzystany jest jako analizator ruchu, a zawarte w nim prototypy pozwalają na wykrywanie anomalii, których występowanie jest realizowane przez dodatkowe oprogramowanie. Głównym zadaniem pakietu dodatkowego oprogramowania jest gromadzenie informacji powstałych z analizy ruchu, kolekcjonowanie ich w bazie danych oraz w przypadku zarejestrowania odpowiedniej ilości danego typu zdarzeń pakiet jest w stanie wygenerować odpowiednie reguły dla zapory. Reguły i ilości zależą od cech sieci i określane są przez administratora sieci bądź jej operatora. Możliwy jest do wykorzystania zestaw zmiennych tworzonych na podstawie informacji od pakietu SNORT, pomagający w generowaniu reguł ściany ogniowej.

Reguły wprowadzane są niezależnym kanałem komunikacyjnym do zapory, co zabezpiecza przez blokowaniem kanałów wejściowych i wyjściowych na zaporze standardowo wykorzystanych do komunikacji sieciowej. Cechą charakterystyczną tego rozwiązania jest fakt zmiany

zachowania się zapory w zależności od generowanego ruchu do i z sieci danej korporacji. System przewiduje również możliwość usuwania dodawanych automatycznie reguł w przypadku zaniku ich występowania. Ta czynność również określana jest przez operatora sieci korporacyjnej.

Ważnym aspektem bezpieczeństwa każdego systemu sieciowego jest rejestrowanie wykonywanych procesów przez użytkowników sieci. Aby to było możliwe, w systemie Windows potrzebna jest dystrybucja systemu operacyjnego minimum enterprise, ponieważ dopiero w tej dystrybucji jądro systemu dostarcza niezbędnych informacji o procesach uruchamianych. Dla systemów klasy Linux niezbędna jest obsługa w jądrze systemu minimum BSD Accounting oraz zestaw narzędzi „acct". W ten sposób skonfigurowany system pozwala na zbieranie niezbędnych informacji. Aby można było analizować zbierane informacje, trzeba je składować centralnie – najlepiej w bazie danych i przetwarzać z wykorzystaniem odpowiednich narzędzi raportujących. W tym konkretnym przypadku wykorzystano zestaw raportujący „iReport" oraz darmową bazę danych PostgreSQL.

Zgromadzone w ten sposób dane można przetwarzać pod wieloma aspektami. Nie wszystkie muszą wiązać się z bezpieczeństwem systemów sieciowych. W pracy skoncentrowano się głównie na aspektach częstotliwości pojawiania się określonych procesów oraz rejestrowaniu wykonywania kluczowych procesów przez osoby inne niż ich twórca lub przeglądaniu kluczowych danych systemu przez osoby postronne (zwykli użytkownicy).



Rysunek 31 Przykładowy ekran raportu systemu iReport.

Wynikiem takich analiz było zgromadzenie odpowiednich raportów oraz informacji o strategicznych punktach systemów sieciowych. Określenie tych punktów istotne jest ze względu na wprowadzanie odpowiednich procedur bezpieczeństwa. Procedury te mają głównie na celu właściwe postępowanie z systemem sieciowym przez operatorów lub administratorów. Może to być właściwa procedura składowania i zabezpieczania kluczowych informacji systemu, jak również wprowadzenie dodatkowych punktów kontrolnych do procedur sprawdzających integralności systemu sieciowego korporacji.

5.3. Podsumowanie

Niniejszy rozdział prezentuje możliwości obecnych systemów oraz metody ich poprawy. Autor zaproponował kilka rozwiązań systemowych w postaci pojedynczych narzędzi lub całych podsystemów, które w znaczny sposób poprawiają niedostatki obecnie dostępnych rozwiązań lub

poprawiają ich możliwości. Autorskie wdrożenia systemu kontroli procesów w dowolnym systemie sieciowych, skojarzenie istniejących rozwiązań IDS z nowym zastosowaniem ich do prowadzenia aktywnej polityki bezpieczeństwa (wykonywanie i zmiana założeń polityki bezpieczeństwa w czasie rzeczywistym) oraz analizatory dzienników systemowych pracujące w czasie rzeczywistym pozwalają na znaczne poprawienie bezpieczeństwa systemu. Analiza zaś tych czynników w sposób ciągły daje możliwość dynamicznego poprawiania wartości ochrony systemu.

Opisywane w niniejszym rozdziale nowe narzędzia posłużyły zarówno do wyznaczenia opisywanych w rozdziale trzecim parametrów i współczynników oraz do przeprowadzenia badań opisywanych w rozdziale czwartym.

6. Wnioski

Badania przeprowadzone podczas realizacji rozprawy rozpoczęły się od wykonania podstawowych analiz danych pochodzących z dzienników zdarzeń rozpatrywanych systemów. Wykonanie tych wstępnych analiz przyczyniło się do rozwoju całego zagadnienia i powstania kwestii opisywanych w prezentowanej pracy. Na podstawie tych właśnie danych opracowano wstępne założenia do pracy, które przekształcono w opisywane w pracy współczynniki i parametry bezpieczeństwa oraz ryzyka sieci. Okazuje się, że w wielu przypadkach niepotrzebne są wyrafinowane narzędzia, aby móc określić niezbędne dane. Uzyskane w ten sposób informacje mogą w sposób wystarczający przyczynić się do wyznaczenia współczynników bezpieczeństwa bądź ryzyka rozpatrywanego systemu. Istnieje również dość poważna trudność. Niektóre ze współczynników, jak na przykład „otwartość", wymagają znajomości często niedostępnych informacji o systemie zabezpieczającym. Przykładem może być konieczność znajomości maksymalnej jednoczesnej zdolności obsługi wielu klientów danej usługi lub podsystemu bezpieczeństwa. Dane takie niejednokrotnie nie są udostępniane właśnie ze względu na możliwości specyficznych ataków na takie podsystemy, w przypadku gdy znamy jego ograniczenia. W dużej mierze byłoby ułatwieniem prezentowanie przez podsystemy nie ich ograniczeń, ale na przykład ilości aktualnie obsługiwanych klientów, jeżeli dalej analizować podany wcześniej przykład. Takich parametrów niestety może być o wiele więcej im bardziej szczegółowo zaczynamy rozpatrywać każdy przypadek.

Innym opisywanym już w pracy problemem jest niejednoznaczność dostarczanych informacji w dziennikach zdarzeń. Bywa czasem, że ten sam producent inaczej prezentuje informacje dla różnych systemów sieciowych.

Nie istnieją w tym zakresie żadne standardy prezentowania informacji. Fakt ten przyczynia się niejednokrotnie do błędnych, a nawet absurdalnych decyzji analizatorów tych danych.

Nie można jednak zostać w przeświadczeniu o niemożliwościach obecnych składowych systemu bezpieczeństwa. Dowodem mogą być prezentowane w tej rozprawie wykonane badania i możliwość uzyskania wyników prowadzących do wyznaczenia wszystkich zakładanych parametrów w rozdziale czwartym.

Analiza dotychczasowych rozwiązań pozwoliła postawić tezę dotyczącą możliwości na podstawie analizy ryzyka i wartości bezpieczeństwa określenia poziomu gwarancji bezpieczeństwa sieci komputerowej. W części teoretycznej pracy:

- Zdefiniowano oryginalne współczynniki: odporność (ξ), otwartość (η), przeciążalność (μ) (rozdział 3.1) pozwalające na wyznaczenie wartości bezpieczeństwa sieci *B(S)* poprzez automatyczne procesy analizy dzienników systemowych,
- Zaproponowano zmodyfikowaną metodę analizy drzew zdarzeń i błędów z nowatorskim rozwiązaniem jakim jest możliwość prowadzenia analizy w czasie rzeczywistym,
- Zdefiniowano nowe pojęcie poziomu gwarancji bezpieczeństwa jako miary różnicy opisanej wzorem 13. Jest to różnica pomiędzy wartością bezpieczeństwa sieci i wartością ryzyka sieci komputerowej,
- Zaproponowano autorskie narzędzie do wykonywania analizy dzienników zdarzeń i odpowiednich obliczeń pozwalających wyznaczyć niezbędne wartości bezpieczeństwa

- Zdefiniowano nowe zależności dla systemów sieciowych zależne od wartości poziomu gwarancji bezpieczeństwa oraz postulaty do dalszego rozwoju systemów bezpieczeństwa i standardów zapisu informacji w dziennikach systemowych.

W wyniku autorskich opracowań opisanych w tej rozprawie było wyznaczenie wartości poszczególnych składowych, a następnie wartości bezpieczeństwa *B(S)* oraz ryzyka *R(S)*. Składowe te dają możliwość operowania wartością poziomu gwarancji bezpieczeństwa opisaną w rozdziale czwartym i piątym, łącznie ze zdolnością przewidywania zachowania się systemu ochrony w czasie dalszej pracy z nim wykraczającym poza aktualne czas działania. Te przewidywane zachowania się podsystemów bezpieczeństwa daje szczególną okazję do zaprezentowania potencjalnym klientom stabilności rozwiązania. Za pomocą tych danych można również planować dalszy rozwój systemów bezpieczeństwa.

Wykonane badania pozwoliły potwierdzić słuszność i poprawność zakładanej tezy – opisanej we wstępie pracy. Jednocześnie udowodnienie słuszności tezy pozwoliło osiągnąć cel pracy, którym było wyznaczenie poziomu gwarancji bezpieczeństwa sieci w czasie rzeczywistym.

Cel pracy zakładał opracowanie metody. W związku z powyższym należy poddać analizie zaproponowane i opisane w pracy rozwiązania, aby jednoznacznie stwierdzić, czy spełniają one definicję metody przytaczaną we wstępie niniejszej pracy. Autor zaproponował możliwość wyznaczania bezpieczeństwa sieci poprzez określenie trzech współczynników. Współczynniki te wyznaczane są na podstawie powtarzalnej procedury możliwej do zrealizowania w dowolnej sieci informatycznej dającej jednoznaczne wyniki – spełnia to kryterium metody. Kolejną prezentowaną

wartością jest ryzyko systemu wyznaczane na podstawie drzew zdarzeń i błędów. Pozyskiwane w tym celu dane pochodzą z podobnych źródeł jak informacje dla wyznaczenia bezpieczeństwa sieci. Można je wyznaczać w sposób ciągły i uzyskiwane wyniki są powtarzalne. Sama procedura wykonania obliczeń jest wykonywalna na dowolnych systemach informatycznych. Na tej podstawie można wnioskować o zgodności z definicją metody. W toku prowadzonych badań zweryfikowano również uzyskane wyniki i potwierdzono możliwość ich wyznaczenia. Jeżeli zatem możliwe jest wyznaczenie poszczególnych współczynników i wartości prowadzących do wyznaczenia głównej wartości jaką jest poziom gwarancji bezpieczeństwa spełnione zostało również kryterium falsyfikowalności.

Na podstawie przeprowadzonej powyżej analizy można stwierdzić osiągnięcie przez autora celu pracy w pełnym jego brzmieniu.

Uzyskane w toku prac parametry bezpieczeństwa, wyznaczenie samego bezpieczeństwa i ryzyka sieci korporacyjnej i jej poszczególnych składowych w bezpośredni sposób przekłada się na poprawę jakości systemów zabezpieczających. W chwili obecnej mogą być udoskonalane, poprawiane, modyfikowane na bieżąco nawet w trakcie pracy systemu. Można również zastosować proponowane rozwiązania celem zmniejszenia kosztów wdrożenia systemów na podstawie analizy bezpieczeństwa w czasie rzeczywistym i bezpośredniego przełożenia na nakłady pieniężne na ten właśnie cel.

Autorskie opracowania nie rozstrzygają jednoznacznie wszystkich możliwości z zakresu tej dziedziny wiedzy. Zaproponowane rozwiązania mogą i wręcz powinny być dalej udoskonalane. Autor na obecnym stanie rozwoju systemów bezpieczeństwa w niektórych przypadkach (np. miara wartości otwartości) nie mógł wykroczyć poza ramy badań laboratoryjnych z powodu niedostatku informacji dotyczących podsystemów

zabezpieczających stosowanych w obecnych systemach sieciowych (opisane dokładnie w rozdziale 5.2), a właściwie z braku standaryzacji zapisu takich danych. Autor zakłada dalsze badania w tej dziedzinie i przygotowanie w toku dalsze pracy odpowiednich zaleceń do stworzenia bardziej wydajnych metod gromadzenia kluczowych informacji o systemie jakimi są zapisy w dziennikach systemowych z działania procesów w systemie sieciowym.

BIBLIOGRAFIA

[1] A. Białas, Podstawy bezpieczeństwa systemów teleinformatycznych, Wydawnictwo Pracowni Komputerowej, Gliwice 2002

[2] Śliwiński Grzegorz, Nowe technologie sieci komputerowych Tom1, Bezpieczeństwo systemu w czasie rzeczywistym. Wydawnictwo Komunikacji i Łączności 2006

[3] Korostil Jerzy, Śliwiński Grzegorz, Wysokowydajne sieci komputerowe – zastosowanie i bezpieczeństwo, Metoda przewidywania gwarancji bezpieczeństwa w sieciach korporacyjnych. Wydawnictwo Komunikacji i Łączności 2005

[4] Korostil Jerzy, Śliwiński Grzegorz, Rocznik informatyki stosowanej WI Nr 8 (Metody informatyki stosowanej w technice i technologii), Wyznaczanie wartości bezpieczeństwa w czasie rzeczywistym. PS II INFORMA, 2005

[5] Korostil Jerzy, Śliwiński Grzegorz. Rocznik informatyki stosowanej WI Nr 8 (Metody informatyki stosowanej w technice i technologii), Gwarancja bezpieczeństwa sieci. PS II INFORMA, 2004

[6] Korostil Jerzy, Śliwiński Grzegorz. Współczesne problemy sieci komputerowych – Nowe technologie, Metoda budowania modelu ryzyka w systemach korporacyjnych. Wydawnictwo Komunikacji i Łączności 2004.

[7] Korostil Jerzy, Śliwiński Grzegorz. Metoda wyznaczania zmiennego ryzyka w systemach komputerowych. Materiały 8 Sesji Naukowej Wydziału Informatyki Szczecin. PS II INFORMA, 2003

[8] Korostil Jerzy, Śliwiński Grzegorz. Zeszyty naukowe Politechniki Śląskiej Studia Informatica Vol24 Nr 2B(54), Analiza parametrów oceny

płynnego poziomu bezpieczeństwa sieci korporacyjnej. Wydawnictwo Komunikacji i Łączności 2003

[9] Korostil Jerzy, Śliwiński Grzegorz. Osobliwości automatyzacji procedur analizy dzienników systemowych administratora. Modelowanie informatyczno - matematyczne układów złożonych - MIMUZ '2002: Materiały Ukraińsko 2002

[10] Śliwiński Grzegorz, Korostil Jerzy. Zeszyty naukowe Politechniki Śląskiej Studia Informatica Vol23 Nr 2B(49), Określenie bieżącej wartości poziomu bezpieczeństwa sieci korporacyjnej. Politechnika Śląska, 2002 S. 271-277 2002

[11] Śliwiński Grzegorz, Wojtaś Robert. Integracja usług w systemie wirtualnym. Pierwsza Konferencja Entuzjastów Informatyki. KEI' 2002 Chełm. Państwowa Wyższa Szkoła Zawodowa 2002

[12] Śliwiński Grzegorz, Wojtaś Robert. Sieci komputerowe w dydaktyce. Materiały 4 Sesji Naukowej Informatyki Szczecin. PS WI INFORMA, 1999 S. 271-275 1999

[13] Śliwiński Grzegorz. Metody dodatkowej ochrony zasobów sieci LAN o publicznie dostępnych stanowiskach. Materiały 3 Sesji Naukowej Instytutu Informatyki Szczecin. PS II INFORMA, 1998 S. 147-154 1998

[14] J. Stokłosa, T. Bilski, T. Pankowski, Bezpieczeństwo danych w systemach informatycznych, PWN 2001

[15] D. Droziński, Hakerzy – Technoanarchiści cyberprzestrzeni, Helion 2001

[16] Jason Ballard, Bud Ratliff. Microsoft Internet Security and Acceleration (ISA) Server 2004. Vademecum Administratora. A.P.N. Promise, 2006

[17] Brian Komar, Ben Smith. Microsoft Windows Security Resource Kit. Wydanie 2, uzupełnione i rozszerzone. A.P.N. Promise, 2006

[18] Eric Cole, James Conley, Ronald L. Krutz. Bezpieczeństwo sieci. Biblia. Wydawnictwo Helion, 2005

[19] Nitesh Dhanjani. Bezpieczeństwo sieci. Narzędzia. Wydawnictwo Helion, 2005

[20] Roberta Bragg. Bezpieczeństwo w Windows Server 2003. Kompendium. Wydawnictwo Helion, 2006

[21] Michael Howard, Steve Lipner. Cykl projektowania zabezpieczeń. Security Development Lifecycle: Proces tworzenia znacząco bezpieczniejszego oprogramowania. A.P.N. Promise, 2006

[22] Anton Chuvakin, Cyrus Pekari. Strażnik bezpieczeństwa danych. Wydawnictwo Helion, 2004

[23] Andrew Lockhart. 100 sposobów na bezpieczeństwo Sieci. Wydawnictwo Helion, 2004

[24] Rick Lehtinen, Deborah Russell. Podstawy ochrony komputerów. Wydawnictwo Helion, 2007

[25] Poland's security policy 1989-2000. SCHOLAR, 2001

[26] J. Carlson, K. Green, E. Schetyna. Bezpieczeństwo w sieci. Wydawnictwo Helion, 2002

[27] Bob Fleck, Bruce Potter. 802.11. Bezpieczeństwo. Wydawnictwo Helion, 2004

[28] Andrew Balinsky, Darrin Miller, Krishna Ankar. Cisco Wireless LAN Security - Bezpieczeństwo sieci bezprzewodowych Cisco. Mikom, 2005

[29] Managing Cisco Network Security Parent. Syngress, 2002

[30] Law Robert. Social Security. East Palgrave Macmillan, 1999

[31] Cisco PIX. Firewalle Wydawnictwo Helion, 2006

[32] Richard Bramante, James Edwards. Nortel Guide to VPN Routing for Security and VIP. Al Martin John Wiley & Sons, 2006

[33] Antoon Rufi. Network Security 1 and 2 Companion Guide. Cisco Press, 2006

[34] M. Kaeo. Designing Network Security.Cisco Press, 2003

[35] Cisco Networking Academy Program Fundamentals of Network Sec Cisco Press, 2004

[36] Cherie Amon, Anne Carasik-Henmi, Debra Littlejohn Skinder. Wielka księga firewalli. Wydawnictwo Helion, 2004

[37] Andrew Vladimirom. Hacking Exposed Cisco Networks. McGraw-Hill, 2006

[38] Paul Taylor. Windows NT 4 Server Czarna Księga Administratora. Wydawnictwo Helion, 1997

[39] Rob Cameron, Chris Cantrell, Anne Hemni. Configuring Juniper Networks NetScreen & SSG Firewalls. Syngress, 2007

[40] D. Teare. CCDA Self-Study Designing for Cisco Internetwork Solutions. Cisco Press, 2007

[41] Stealing Network. How to Own Box. Syngress, 2005

[42] Stealing the Network. Syngress, 2005

[43] Securing & Controlling Cisco Routers CRC Press, Inc., 2002

[44] Alex Lukatsky.Wykrywanie włamań i aktywna ochrona danych. Wydawnictwo Helion, 2004

[45] Bidgoli. Handbook of Information Security. John Wiley & Sons, 2006

[46] Drew Heywood. Windows NT 4 Server - Vademecum profesjonalisty. Wydawnictwo Helion, 1999

[47] Anton Chuvakin, Cyrus Pekari. Strażnik bezpieczeństwa danych. Wydawnictwo Helion, 2004

[48] Angela Orebaugh, Michael Rash. IPS zapobieganie i aktywne przeciwdziałanie intruzom. Mikom, 2005

[49] Hack Proofing Your Network. Edycja polska, Wydawnictwo Helion, 2002

[50] Edgar Danielyan, Patrick T. Lane, James Tanger. Hack Proofing Linux. Edycja polska. Wydawnictwo Helion, 2003

[51] Marek Gusta, Maciej Szmit. 101 zabezpieczeń przed atakami w sieci komputerowej. Wydawnictwo Helion, 2005

[52] Agresja i Ochrona II wydanie. Robomatic, 2003

[53] Henry Benjamin. CCIE 4695 CCIE Security Oficjalny podręcznik przygotowujący do egzaminu. Mikom, 2004

[54] Andrzej Dudek. Nie tylko wirusy. Wydawnictwo Helion, 1998

[55] John Chirillo. Hack Wars. Tom 2. Administrator kontratakuje. Wydawnictwo Helion, 2002

[56] CCSP Snpa Official Exam Certification Guide. Cisco Press, 2006

[57] Z. Naseh. Designing Content Switching Solutions. Cisco Press, 2006

[58] Brian Komar. Administracja sieci TCP/IP dla każdego. Wydawnictwo Helion, 2000

[59] John Chirillo. Hack Wars - tom 1. Na tropie hakerów. Wydawnictwo Helion, 2002

[60] Haking numer 3 / 2006 (17) Software, 2006

[61] Radosław Sokół. Vademecum hakera. Zabezpieczenia w Windows. Wydawnictwo Helion, 2004

[62] Michał Zalewski. Cisza w sieci. Wydawnictwo Helion, 2005

[63] Jonathan Reuvid. E-biznes bez ryzyka. Zarządzanie bezpieczeństwem w sieci. Wydawnictwo Helion, 2007

[64] 100 sposobów na Linux. Wydawnictwo Helion, 2005

[65] Apache. Zabezpieczenia aplikacji i serwerów WWW, Wydawnictwo Helion, 2007

[66] Kathy Ivens. Sposoby na sieci domowe. Wydawnictwo Helion, 2005

[67] Joli Ballew, Jeff Duntemann. Optymalizacja systemu Windows. Wydawnictwo Helion, 2004

[68] Scott Mueller, Terry W. Ogletree. Rozbudowa i naprawa sieci. Kompendium. Wydawnictwo Helion, 2004

[69] Wallace Wang. Tajemnice internetu, hackingu i bezpieczeństwa. Wydawnictwo Helion, 2004

[70] Rob Cameron, Chris Cantrell, Anne Hemni. Configuring Juniper Networks NetScreen & SSG Firewalls. Syngress, 2007

[71] Greg Bastien, Christian Abera Degu. Ściany ogniowe Cisco PIX. Mikom, 2004

[72] Andy McNab. Firewall. Mikom, 2005

[73] Jacek Artymiak. OpenBSD. Tworzenie firewalli za pomocą PF. Wydawnictwo Helion, 2004

[74] Erik Pace Birkholz. Bezpieczeństwo komputerów i sieci. Operacje specjalne. Translator S.C., 2003

[75] Brian Hatch, George Kurtz, James Lee. Hakerzy w Linuksie. Sekrety zabezpieczeń sieci komputerowych. Wydanie drugie. Translator S.C., 2003

[76] Scott Mueller. Rozbudowa i naprawa laptopów. Wydawnictwo Helion, 2004

[77] Mahbub Hassan, Raj Jain Wysoko wydajne sieci TCP/IP. Wydawnictwo Helion, 2004

[78] Kevin Lam, David LeBlanc, Ben Smith. Ocena bezpieczeństwa sieciowego. A.P.N. Promise, 2006

[79] Michael Horton, Clinton Mugge. Notes antyhakera. Bezpieczenstwo sieci. Translator S.C., 2004

[80] Stuart McClure, Joel Scambray. Hakerzy w Windows 2000. Translator S.C., 2002

[81] Jonathan Reuvid. E-biznes bez ryzyka. Zarządzanie bezpieczeństwem w sieci. Wydawnictwo Helion, 2007

[82] James F. Kurose, Keith W. Ross. Sieci komputerowe. Od ogółu do szczegółu z internetem w tle. Wydanie 3. Wydawnictwo Helion, 2006

[83] Jamie Butler, Greg Hoglund. Rootkity. Sabotowanie jądra systemu Windows. Wydawnictwo Helion, 2006

[84] Alan Luber. Bezpieczny komputer. Czy stać Cię na utratę danych? Translator S.C., 2003

[85] Douglas E. Comer. Sieci komputerowe i intersieci. aplikacje internetowe. Wydawnictwa Naukowo Techniczne, 2003

[86] JBoss 4.0. Podręcznik administratora, Wydawnictwo Helion, 2005

[87] Rob Flickenger, Roger Weeks. 100 sposobów na sieci bezprzewodowe. Wydanie II. Wydawnictwo Helion, 2006

[88] Henry Benjamin. CCIP Security Exam Companion Guide, Tłumaczenie: Małgorzata Mikulska, Krzysztof Turczyński., 2004

[89] Greg Bastien, Christian Abera Degu. Ściany ogniowe Cisco PIX, 2004

[90] Anthony Bruno, Jaqueline Kim. CCDA. Certyfikat projektanta sieci Cisco. Mikom, 2004

[91] Jake Babbin, Graham Clark, Angela Orebaugh, Becky Pinkard, Michael. IPS. Zapobieganie i aktywne przeciwdziałanie intruzom. Mikom, 2005

[92] Matthew Strebe. Bezpieczeństwo sieci. Mikom, 2005

[93] Andrew Balinsky, Darrin Miller, Krishna Ankar. Bezpieczeństwo sieci bezprzewodowych. Sri Sundaralingam. Mikom, 2005

[94] Pejman Roshan, Jonathan Leary. Bezprzewodowe sieci LAN 802.11. PWN, 2006

[95] Henry Benjamin. CCIE Security, 2004

[96] Krzysztof Liderman. Podręcznik administratora bezpieczeństwa teleinformatycznego. Mikom, 2003

[97] Greg Bastien, Christian Abera Degu. Ściany ogniowe Cisco. Mikom, 2004

[98] Libor Dostálek. Bezpieczeństwo protokołu TCP/IP. PWN, 2006

[99] Stuart McClure, Joel Scambray, George Kurtz. Hacking zdemaskowany. PWN, 2006

[100] Jake Babbin, Graham Clark, Michael Rash. IPS. Zapobieganie intruzom. Mikom, 2005

[101] Huang, Scott; MacCallum, David. Network Security, Springer-Verlag New York Inc., 2007

[102] Cobb, Chey. Network Security for Dummies, 2002

[103] Malik, Sadat. Network Security Architectures. Pearson Education, 2004

[104] Rufi, Antoon. Network Security 1 and 2 Companion Guide. Cisco Press, 2003

[105] Maiwald, Eric. Network Security A Beginner's Guide 2nd Revised editio, 2003

[106] McNab, Chris. Network Security Assessment, 2004

[107] Cunningham, Bryan; Fuller, Ed; Miles, Greg. Network Security Evaluation Using the NSA IEM, 2005

[108] McNab, Andy. Firewall. Mikom, 2002

[109] Mankell, Henning. Firewall. Helion, 2002

[110] Cheswick, William R.; Bellovin, Steven M.; Rubin, Aviel. Firewalls and Internet Security. Respelling the Wily Hacker, 2003

[111] Komar, Brian; Beekelaar, Ronald; Wettern, Joern. Firewalls for Dummies, 2003

[112] Hemni, Anne. Firewall Administration Professional, 2007

[113] Pohlmann, Norbert; Crothers, Tim. Firewall Architecture for the Enterprise, 2002

[114] Arrtmiak, Jacek; Vandeputte, Wim; Hartmeier, Daniel. Firewall Warrior, 2007

[115] Oakes, Edward T.; Tibbs, Richard. Firewalls and VPN's Principles and Practices, 2005

[116] Amon, Cherie; Maxwell, Douglas; Noble, James S. Checkpoint NG VPN 1/Firewall 1 Advanced Configuration and Troubleshooting New title , 2003

[117] Gollmann, D. Computer Security 2e 2, 2005

[118] Lehtinen, R. Computer Security Basics 2nd Revised editio, 2006

[119] Bace Rebecca, Mall Peter: Intrusion Detection Systems, NIST, listopad 2001 [http://csrc.nist.gov/publications/nistpubs/800-31/sp800-31.pdf]

[120] Boran Sean. The IT security cookbook, Boran Consulting, 2002 [http://www.boran.com/security/]

[121] [http://www.pentics.net/denial-of-service/white-papers/smurf.txt], 2000

[122] Kaeo Merike. Tworzenie bezpiecznych sieci, Mikom, Warszawa 2000

[123] Kruk Dominik. Namierzanie podsłuchu, PCkurier 9/2002

[124] Kruk Dominik. Przynęta dla intruza, PCkurier 2/2002

[125] Maiwald Eric. Bezpieczeństwo w sieci. Kurs podstawowy, Edition 2000, Kraków 2001

[126] National Institute of Standards and Technology. An Introduction to Computer Security, październik 1995

[127] [http://csrc.nist.gov/publications/nistpubs/800-12/], 1995

[128] Wheeler David A. Why Open Source Software / Free Software (OSS/FS)? Look at the Numbers!, 2003

[129] [http://www.sans.org/resources/], 2006

[130] A.Adamski. Prawo do bezpiecznej sieci. Raport - "Bezpieczeństwo sieciowe", 2000

[131] J.Barta, R.Markiewicz. Sposób na pirata, Rzeczpospolita, nr 33/99.

[132] B.Fischer. Intruzi i samuraje, Prawo i życie, nr 1/99.

[133] B.Fischer. Profilaktyka kryminalistyczna, Prawo i życie, nr 46/98.

[134] B.Fischer. Przestępczość komputerowa (3), Prawo i życie, nr 24/97.

[135] K.Indecki. Przestępstwo paserstwa w kodeksie karnym 1969 r. Analiza dogmatyczna, 1991

[136] Internetowe ataki, Pckurier, nr 08/98.

[137] S.Iwanicki. Przemówienia powitalne, (w:) A.Adamski (red.), Prawne aspekty nadużyć popełnionych z wykorzystaniem nowoczesnych technologii przetwarzania informacji. Materiały z konferencji naukowej, TNOiK, Toruń 1994.

[138] K.J.Jakubski. Przestępczość komputerowa-zarys problematyki, Prokuratura i prawo, nr 12/96.

[139] K.J.Jakubski. Wybrane zagadnienia ścigania przestępczości komputerowej,(w:) Materiały seminarium NASK i TP S.A Miedzeszyn'96, część 1, NASK, Warszawa 1996.

[140] PN-I-02000, Technika informatyczna. Zabezpieczenia w systemach informatycznych, 1998

[141] Krzysztof Liderman. Podręcznik administratora bezpieczeństwa teleinformatycznego. Wydawnictwo Mikom, Warszawa 2003.

[142] Ryszard Budziński. Komputerowy system przetwarzania danych ekonomiczno-finansowych w przedsiębiorstwie. Wydawnictwo Naukowe Uniwersytetu Szczecińskiego PAN, Warszawa 2000

[143] Ryszard Budziński. Metodologiczne aspekty systemowego przetwarzania danych ekonomiczno-finansowych w przedsiębiorstwie. Wydawnictwo Naukowe Uniwersytetu Szczecińskiego, Szczecin 2002

Spis rysunków

Spis tabel

Dodatek A

Kody programów i wyniki przetwarzania danych

Analizator fraz (“parser”)

```
#!/usr/bin/php4
<?
function getOptions($opt)
// Czytanie parametrow linii polecen
{
    $paramNum = $GLOBALS["argc"];
    $paramStr = $GLOBALS["argv"];

    for($i=1; $i<$paramNum; $i++) {

     if (($paramStr[$i][0] == "-") && ($subOpt = strstr($opt,
$paramStr[$i][1]))) {
      if (($subOpt[1] == ":") && ($paramStr[$i+1][0] != "-")) {
       $param[$paramStr[$i][1]] = $paramStr[$i+1];
       $i++;
     } else {
    $param[$paramStr[$i][1]] = true;
    }
  }
 }

return $param;
}

// Ustawienie odczytu parametrow:
//  d - tylko czy wlaczona jest ta opcja
//  f: - sprawdzenie opcji -f i jej zawartosci nazwa
$opcje = getOptions("dvf:s:hl");

// Wyswietlenie zawartosci tablicy
//print_r($opcje);

// Wyswietlenie pomocy
if( $opcje[h] ) {
    echo "\nAnalizator informacji plików *.log systemu
LINUX\n\n";
    echo "Wykonanie:\tPARSE [opcje]\n\n";
    echo "OPCJE:\n\n";
    echo "\t-f nazwa  - plik wejściowy dla analizatora,\n\n";
    echo "\t-s fraza  - analizowana FRAZA w pliku
wejściowym,\n\n";
    echo "\t-v        - wyświetlanie poszczegolnych
analizowanych linii,\n\n";
```

```
    echo "\t-d        - prezentacja wyników z dokładnością± do
1-go dnia lub przy\n";
    echo "\t            braku parametru dokładność do 1
godziny,\n\n";
    echo "\t-l        - pokaż licznik poszukiwań,\n\n";
    echo "\t-h        - wyświetlenie tej pomocy.\n\n\n\n";
    exit(0);
}

// Wyświetlenie poszczególnych opcji
if( $opcje[f] ) {
echo "Plik: $opcje[f]\n";
}else{
 echo "Brakuje parametru:\n";
 echo "   -f nazwa_pliku\n\n";
 exit(1);
}

if( $opcje[s] ) {
  echo "Fraza: >$opcje[s]<\n";
}else{
  echo "Brakuje parametru:\n";
  echo "   -s poszukiwana_fraza\n\n";
  exit(1);
}

if( $opcje[v] ) {
 echo "Podglad: ON\n";
}else{
 echo "Podglad: OFF\n";
}

if( $opcje[l] ) {
 echo "Licznik: ON\n";
}else{
 echo "Licznik: OFF\n";
}

if( $opcje[d] ) {
 echo "Dokładność: 24 [h]\n";
}else{
 echo "Dokładność: 1 [h]\n";
}

echo "**************************************************\n";

// Otwarcie pliku z opcji -f nazwa w trybie CZYTANIE
$fd = fopen($opcje[f],"r");

// Ustawienie licznika i na zero
$i=0;
$licznik=0;

while (!feof ($fd)) {
```

```
   $buffer = fgets($fd, 4096);
   // Przeszukiwanie "fraza" linii w pliku zgodnie z parametrem
-s fraza
   if ( strpos($buffer, $opcje[s]) ) {
        // Wyświetlanie poszczególnych linii -v lub wiatraczka
      if( $opcje[v] ) {
          echo $buffer;
      }
      // Dzielenie linii na tablice z pominieciem pustych
znakow {seperator= blank, comma, : }
      $tmpA = preg_split("/[\s,:]+/", $buffer, -1,
PREG_SPLIT_NO_EMPTY);
//    print_r($tmpA);
      // Podzial wynikow na dniowe lub godzinne w ramach dni --
opcja -d
      if( $opcje[d] ) {
      $TAB["$tmpA[0] $tmpA[1]"] = $TAB["$tmpA[0] $tmpA[1]"]+1;
      }else{
      $TAB["$tmpA[0] $tmpA[1] $tmpA[2]"] = $TAB["$tmpA[0]
$tmpA[1] $tmpA[2]"]+1;
      }
//    print_r($TAB);
        $i++;
     }
   if( $opcje[l] ) {
   $licznik++;
   echo "  $licznik:$i\r";
   }
}
fclose ($fd);

// Dodanie do tablicy wartości sumarycznej TOTAL
$TAB["Total"] = $i;
echo " Zakończono\n";
echo "************************************************\n";
// Wyświetlenie wyników
print_r($TAB);
?>
```

Przykład wykonania parsera i wyniku dla ataków DNS

```
Plik: /var/log/daemon.log
Fraza: >lame server<
Podglad: ON
Licznik: ON
Dokładność: 1 [h]
************************************************
May 13 06:43:32 archi named[785]: lame server resolving
'149.248.141.58.relays.visi.com' (in 'relays.visi.com'?): 127.0.0.1#53
May 13 07:02:32 archi named[785]: lame server resolving
'194.88.130.134.relays.visi.com' (in 'relays.visi.com'?): 127.0.0.1#53
May 13 07:02:32 archi named[785]: lame server resolving
'163.241.34.201.relays.visi.com' (in 'relays.visi.com'?): 127.0.0.1#53
May 13 07:05:35 archi named[785]: lame server resolving '40.12.146.211.in-
addr.arpa' (in '146.211.in-addr.arpa'?): 203.119.26.3#53
```

```
May 13 07:10:16 archi named[785]: lame server resolving
'79.89.175.220.relays.visi.com' (in 'relays.visi.com'?): 127.0.0.1#53
May 13 07:15:58 archi named[785]: lame server resolving
'233.111.140.58.relays.visi.com' (in 'relays.visi.com'?): 127.0.0.1#53
May 13 07:16:14 archi named[785]: lame server resolving
'233.111.140.58.relays.visi.com' (in 'relays.visi.com'?): 127.0.0.1#53
May 13 07:17:02 archi named[785]: lame server resolving '166.197.69.85.in-
addr.arpa' (in '197.69.85.in-addr.arpa'?): 82.216.111.76#53
May 13 07:17:02 archi named[785]: lame server resolving '166.197.69.85.in-
addr.arpa' (in '197.69.85.in-addr.arpa'?): 82.216.111.75#53
May 13 07:17:03 archi named[785]: lame server resolving '166.197.69.85.in-
addr.arpa' (in '197.69.85.in-addr.arpa'?): 82.216.111.76#53
May 13 07:17:03 archi named[785]: lame server resolving '166.197.69.85.in-
addr.arpa' (in '197.69.85.in-addr.arpa'?): 82.216.111.75#53
May 13 07:17:05 archi named[785]: lame server resolving '166.197.69.85.in-
addr.arpa' (in '197.69.85.in-addr.arpa'?): 82.216.111.76#53
May 13 07:17:05 archi named[785]: lame server resolving '166.197.69.85.in-
addr.arpa' (in '197.69.85.in-addr.arpa'?): 82.216.111.75#53
May 13 07:19:11 archi named[785]: lame server resolving '160.122.205.195.in-
addr.arpa' (in '122.205.195.in-addr.arpa'?): 81.15.224.211#53
May 13 07:20:08 archi named[785]: lame server resolving
'100.33.156.61.relays.visi.com' (in 'relays.visi.com'?): 127.0.0.1#53
May 13 07:21:08 archi named[785]: lame server resolving '2.105.96.217.in-
addr.arpa' (in '105.96.217.in-addr.arpa'?): 195.66.73.11#53
May 13 07:21:08 archi named[785]: lame server resolving '2.105.96.217.in-
addr.arpa' (in '105.96.217.in-addr.arpa'?): 195.66.73.2#53
May 13 07:21:09 archi named[785]: lame server resolving '2.105.96.217.in-
addr.arpa' (in '105.96.217.in-addr.arpa'?): 195.66.73.11#53
May 13 07:21:09 archi named[785]: lame server resolving '2.105.96.217.in-
addr.arpa' (in '105.96.217.in-addr.arpa'?): 195.66.73.2#53
May 13 07:21:11 archi named[785]: lame server resolving '2.105.96.217.in-
addr.arpa' (in '105.96.217.in-addr.arpa'?): 195.66.73.11#53
May 13 07:21:11 archi named[785]: lame server resolving '2.105.96.217.in-
addr.arpa' (in '105.96.217.in-addr.arpa'?): 195.66.73.2#53
May 13 07:21:52 archi named[785]: lame server resolving '26.136.160.82.in-
addr.arpa' (in '136.160.82.in-addr.arpa'?): 202.157.182.142#53
May 13 07:21:52 archi named[785]: lame server resolving '26.136.160.82.in-
addr.arpa' (in '136.160.82.in-addr.arpa'?): 82.160.136.1#53
May 13 07:21:52 archi named[785]: lame server resolving '26.136.160.82.in-
addr.arpa' (in '136.160.82.in-addr.arpa'?): 82.160.136.1#53
May 13 07:21:53 archi named[785]: lame server resolving '26.136.160.82.in-
addr.arpa' (in '136.160.82.in-addr.arpa'?): 202.157.182.142#53
May 13 07:21:53 archi named[785]: lame server resolving '26.136.160.82.in-
addr.arpa' (in '136.160.82.in-addr.arpa'?): 82.160.136.1#53
May 13 07:21:54 archi named[785]: lame server resolving '26.136.160.82.in-
addr.arpa' (in '136.160.82.in-addr.arpa'?): 202.157.182.142#53
May 13 07:24:04 archi named[785]: lame server resolving
'8.117.110.217.relays.visi.com' (in 'relays.visi.com'?): 127.0.0.1#53
May 13 07:24:25 archi named[785]: lame server resolving
'8.117.110.217.relays.visi.com' (in 'relays.visi.com'?): 127.0.0.1#53
May 13 07:25:57 archi named[785]: lame server resolving '71.219.255.84.in-
addr.arpa' (in '71.219.255.84.in-addr.arpa'?): 84.255.216.123#53
May 13 07:25:57 archi named[785]: lame server resolving '71.219.255.84.in-
addr.arpa' (in '71.219.255.84.in-addr.arpa'?): 84.255.216.53#53
May 13 07:25:58 archi named[785]: lame server resolving '71.219.255.84.in-
addr.arpa' (in '71.219.255.84.in-addr.arpa'?): 84.255.216.123#53
May 13 07:25:58 archi named[785]: lame server resolving '71.219.255.84.in-
addr.arpa' (in '71.219.255.84.in-addr.arpa'?): 84.255.216.53#53
May 13 07:26:00 archi named[785]: lame server resolving '71.219.255.84.in-
addr.arpa' (in '71.219.255.84.in-addr.arpa'?): 84.255.216.123#53
May 13 07:26:00 archi named[785]: lame server resolving '71.219.255.84.in-
addr.arpa' (in '71.219.255.84.in-addr.arpa'?): 84.255.216.53#53
May 13 07:28:41 archi named[785]: lame server resolving 'rev1.kornet.net' (in
'kornet.net'?): 211.216.50.160#53
May 13 07:28:41 archi named[785]: lame server resolving 'rev2.kornet.net' (in
'kornet.net'?): 211.216.50.160#53
```

```
May 13 07:37:53 archi named[785]: lame server resolving
'12.113.219.193.relays.visi.com' (in 'relays.visi.com'?): 127.0.0.1#53
May 13 07:47:32 archi named[785]: lame server resolving '17.42.143.83.in-
addr.arpa' (in '42.143.83.in-addr.arpa'?): 83.143.40.158#53
May 13 07:48:13 archi named[785]: lame server resolving '209.144.219.81.in-
addr.arpa' (in '144.219.81.in-addr.arpa'?): 62.233.128.18#53
May 13 07:48:13 archi named[785]: lame server resolving '209.144.219.81.in-
addr.arpa' (in '144.219.81.in-addr.arpa'?): 80.85.225.162#53
May 13 07:51:49 archi named[785]: lame server resolving '46.87.98.217.in-
addr.arpa' (in '87.98.217.in-addr.arpa'?): 77.232.72.50#53
May 13 07:52:35 archi named[785]: lame server resolving '32.170.136.195.in-
addr.arpa' (in '170.136.195.in-addr.arpa'?): 195.136.250.201#53
May 13 07:58:27 archi named[785]: lame server resolving '148.104.122.212.in-
addr.arpa' (in '104.122.212.in-addr.arpa'?): 213.0.184.69#53
May 13 07:59:13 archi named[785]: lame server resolving '200.151.57.61.in-
addr.arpa' (in '151.57.61.in-addr.arpa'?): 61.57.159.130#53
May 13 07:59:13 archi named[785]: lame server resolving '200.151.57.61.in-
addr.arpa' (in '151.57.61.in-addr.arpa'?): 61.57.159.135#53
May 13 08:00:12 archi named[785]: lame server resolving '78.0/25.127.219.81.in-
addr.arpa' (in '0/25.127.219.81.in-addr.arpa'?): 217.97.239.165#53
May 13 08:01:20 archi named[785]: lame server resolving '37.55.78.210.in-
addr.arpa' (in '78.210.in-addr.arpa'?): 203.119.26.3#53
May 13 08:06:27 archi named[785]: lame server resolving '241.156.199.200.in-
addr.arpa' (in '156.199.200.in-addr.arpa'?): 200.223.0.135#53
May 13 08:10:03 archi named[785]: lame server resolving '26.136.160.82.in-
addr.arpa' (in '136.160.82.in-addr.arpa'?): 82.160.136.1#53
May 13 08:10:04 archi named[785]: lame server resolving '26.136.160.82.in-
addr.arpa' (in '136.160.82.in-addr.arpa'?): 202.157.182.142#53
May 13 08:11:37 archi named[785]: lame server resolving '166.197.69.85.in-
addr.arpa' (in '197.69.85.in-addr.arpa'?): 82.216.111.75#53
May 13 08:11:37 archi named[785]: lame server resolving '166.197.69.85.in-
addr.arpa' (in '197.69.85.in-addr.arpa'?): 82.216.111.76#53
May 13 08:11:56 archi named[785]: lame server resolving '71.219.255.84.in-
addr.arpa' (in '71.219.255.84.in-addr.arpa'?): 84.255.216.123#53
May 13 08:11:56 archi named[785]: lame server resolving '71.219.255.84.in-
addr.arpa' (in '71.219.255.84.in-addr.arpa'?): 84.255.216.53#53
May 13 08:14:36 archi named[785]: lame server resolving '148.104.122.212.in-
addr.arpa' (in '104.122.212.in-addr.arpa'?): 213.0.184.69#53
May 13 08:15:23 archi named[785]: lame server resolving '200.151.57.61.in-
addr.arpa' (in '151.57.61.in-addr.arpa'?): 61.57.159.135#53
May 13 08:15:24 archi named[785]: lame server resolving '200.151.57.61.in-
addr.arpa' (in '151.57.61.in-addr.arpa'?): 61.57.159.130#53
May 13 08:15:52 archi named[785]: lame server resolving '78.0/25.127.219.81.in-
addr.arpa' (in '0/25.127.219.81.in-addr.arpa'?): 217.97.239.165#53
May 13 08:16:38 archi named[785]: lame server resolving '121.184.69.85.in-
addr.arpa' (in '184.69.85.in-addr.arpa'?): 82.216.111.75#53
May 13 08:16:38 archi named[785]: lame server resolving '121.184.69.85.in-
addr.arpa' (in '184.69.85.in-addr.arpa'?): 82.216.111.76#53
May 13 08:20:11 archi named[785]: lame server resolving '86.217.122.192.in-
addr.arpa' (in '217.122.192.in-addr.arpa'?): 192.111.39.1#53
May 13 08:26:10 archi named[785]: lame server resolving
'34.172.177.85.relays.visi.com' (in 'relays.visi.com'?): 127.0.0.1#53
May 13 08:31:13 archi named[785]: lame server resolving '166.197.69.85.in-
addr.arpa' (in '197.69.85.in-addr.arpa'?): 82.216.111.75#53
May 13 08:31:13 archi named[785]: lame server resolving '166.197.69.85.in-
addr.arpa' (in '197.69.85.in-addr.arpa'?): 82.216.111.76#53
May 13 08:32:50 archi named[785]: lame server resolving '71.219.255.84.in-
addr.arpa' (in '71.219.255.84.in-addr.arpa'?): 84.255.216.123#53
May 13 08:32:50 archi named[785]: lame server resolving '71.219.255.84.in-
addr.arpa' (in '71.219.255.84.in-addr.arpa'?): 84.255.216.53#53
May 13 08:35:13 archi named[785]: lame server resolving '95.205.230.80.in-
addr.arpa' (in '230.80.in-addr.arpa'?): 192.116.192.9#53
May 13 08:44:49 archi named[785]: lame server resolving '42.79.77.193.in-
addr.arpa' (in '79.77.193.in-addr.arpa'?): 193.189.160.12#53
May 13 08:46:48 archi named[785]: lame server resolving 'rev1.kornet.net' (in
'kornet.net'?): 211.216.50.160#53
```

```
May 13 08:46:48 archi named[785]: lame server resolving 'rev2.kornet.net' (in
'kornet.net'?): 211.216.50.160#53
May 13 08:50:13 archi named[785]: lame server resolving '26.136.160.82.in-
addr.arpa' (in '136.160.82.in-addr.arpa'?): 202.157.182.142#53
May 13 08:50:13 archi named[785]: lame server resolving '26.136.160.82.in-
addr.arpa' (in '136.160.82.in-addr.arpa'?): 82.160.136.1#53
May 13 08:54:47 archi named[785]: lame server resolving '8.1.230.80.in-
addr.arpa' (in '230.80.in-addr.arpa'?): 192.116.192.9#53
May 13 09:01:49 archi named[785]: lame server resolving '200.151.57.61.in-
addr.arpa' (in '151.57.61.in-addr.arpa'?): 61.57.159.130#53
May 13 09:01:49 archi named[785]: lame server resolving '200.151.57.61.in-
addr.arpa' (in '151.57.61.in-addr.arpa'?): 61.57.159.135#53
May 13 09:13:02 archi named[785]: lame server resolving '166.197.69.85.in-
addr.arpa' (in '197.69.85.in-addr.arpa'?): 82.216.111.75#53
May 13 09:13:03 archi named[785]: lame server resolving '166.197.69.85.in-
addr.arpa' (in '197.69.85.in-addr.arpa'?): 82.216.111.76#53
May 13 09:15:11 archi named[785]: lame server resolving '71.219.255.84.in-
addr.arpa' (in '71.219.255.84.in-addr.arpa'?): 84.255.216.123#53
May 13 09:15:11 archi named[785]: lame server resolving '71.219.255.84.in-
addr.arpa' (in '71.219.255.84.in-addr.arpa'?): 84.255.216.53#53
May 13 09:15:20 archi named[785]: lame server resolving '71.219.255.84.in-
addr.arpa' (in '71.219.255.84.in-addr.arpa'?): 84.255.216.123#53
May 13 09:15:20 archi named[785]: lame server resolving '71.219.255.84.in-
addr.arpa' (in '71.219.255.84.in-addr.arpa'?): 84.255.216.53#53
May 13 09:22:42 archi named[785]: lame server resolving
'50.73.150.213.relays.visi.com' (in 'relays.visi.com'?): 127.0.0.1#53
May 13 09:22:52 archi named[785]: lame server resolving
'50.73.150.213.relays.visi.com' (in 'relays.visi.com'?): 127.0.0.1#53
May 13 09:23:04 archi named[785]: lame server resolving
'50.73.150.213.relays.visi.com' (in 'relays.visi.com'?): 127.0.0.1#53
May 13 09:23:16 archi named[785]: lame server resolving
'50.73.150.213.relays.visi.com' (in 'relays.visi.com'?): 127.0.0.1#53
May 13 09:32:13 archi named[785]: lame server resolving '200.151.57.61.in-
addr.arpa' (in '151.57.61.in-addr.arpa'?): 61.57.159.135#53
May 13 09:32:13 archi named[785]: lame server resolving '200.151.57.61.in-
addr.arpa' (in '151.57.61.in-addr.arpa'?): 61.57.159.130#53
May 13 09:34:13 archi named[785]: lame server resolving '26.136.160.82.in-
addr.arpa' (in '136.160.82.in-addr.arpa'?): 82.160.136.1#53
May 13 09:34:13 archi named[785]: lame server resolving '26.136.160.82.in-
addr.arpa' (in '136.160.82.in-addr.arpa'?): 202.157.182.142#53
May 13 09:38:34 archi named[785]: lame server resolving '122.68.13.151.in-
addr.arpa' (in '68.13.151.in-addr.arpa'?): 193.70.192.100#53
May 13 09:38:34 archi named[785]: lame server resolving '122.68.13.151.in-
addr.arpa' (in '68.13.151.in-addr.arpa'?): 195.210.91.100#53
May 13 09:38:55 archi named[785]: lame server resolving '122.68.13.151.in-
addr.arpa' (in '68.13.151.in-addr.arpa'?): 193.70.192.100#53
May 13 09:38:55 archi named[785]: lame server resolving '122.68.13.151.in-
addr.arpa' (in '68.13.151.in-addr.arpa'?): 195.210.91.100#53
May 13 09:55:03 archi named[785]: lame server resolving '108.5.31.121.in-
addr.arpa' (in '31.121.in-addr.arpa'?): 221.7.128.68#53
May 13 09:55:30 archi named[785]: lame server resolving '166.197.69.85.in-
addr.arpa' (in '197.69.85.in-addr.arpa'?): 82.216.111.75#53
May 13 09:55:31 archi named[785]: lame server resolving '166.197.69.85.in-
addr.arpa' (in '197.69.85.in-addr.arpa'?): 82.216.111.76#53
May 13 09:57:17 archi named[785]: lame server resolving '71.219.255.84.in-
addr.arpa' (in '71.219.255.84.in-addr.arpa'?): 84.255.216.123#53
May 13 09:57:17 archi named[785]: lame server resolving '71.219.255.84.in-
addr.arpa' (in '71.219.255.84.in-addr.arpa'?): 84.255.216.53#53
May 13 10:06:40 archi named[785]: lame server resolving '122.68.13.151.in-
addr.arpa' (in '68.13.151.in-addr.arpa'?): 193.70.192.100#53
May 13 10:06:40 archi named[785]: lame server resolving '122.68.13.151.in-
addr.arpa' (in '68.13.151.in-addr.arpa'?): 195.210.91.100#53
May 13 10:10:21 archi named[785]: lame server resolving '148.104.122.212.in-
addr.arpa' (in '104.122.212.in-addr.arpa'?): 213.0.184.69#53
May 13 10:10:28 archi named[785]: lame server resolving '148.104.122.212.in-
addr.arpa' (in '104.122.212.in-addr.arpa'?): 213.0.184.69#53
```

```
May 13 10:15:08 archi named[785]: lame server resolving '200.151.57.61.in-
addr.arpa' (in '151.57.61.in-addr.arpa'?): 61.57.159.130#53
May 13 10:15:08 archi named[785]: lame server resolving '200.151.57.61.in-
addr.arpa' (in '151.57.61.in-addr.arpa'?): 61.57.159.135#53
May 13 10:15:22 archi named[785]: lame server resolving '200.151.57.61.in-
addr.arpa' (in '151.57.61.in-addr.arpa'?): 61.57.159.135#53
May 13 10:15:22 archi named[785]: lame server resolving '200.151.57.61.in-
addr.arpa' (in '151.57.61.in-addr.arpa'?): 61.57.159.130#53
May 13 10:15:36 archi named[785]: lame server resolving '244.43.123.62.in-
addr.arpa' (in '43.123.62.in-addr.arpa'?): 193.0.0.193#53
May 13 10:16:22 archi named[785]: lame server resolving '26.136.160.82.in-
addr.arpa' (in '136.160.82.in-addr.arpa'?): 202.157.182.142#53
May 13 10:16:22 archi named[785]: lame server resolving '26.136.160.82.in-
addr.arpa' (in '136.160.82.in-addr.arpa'?): 82.160.136.1#53
May 13 10:16:41 archi named[785]: lame server resolving '26.136.160.82.in-
addr.arpa' (in '136.160.82.in-addr.arpa'?): 82.160.136.1#53
May 13 10:16:41 archi named[785]: lame server resolving '26.136.160.82.in-
addr.arpa' (in '136.160.82.in-addr.arpa'?): 202.157.182.142#53
May 13 10:37:48 archi named[785]: lame server resolving '166.197.69.85.in-
addr.arpa' (in '197.69.85.in-addr.arpa'?): 82.216.111.75#53
May 13 10:37:48 archi named[785]: lame server resolving '166.197.69.85.in-
addr.arpa' (in '197.69.85.in-addr.arpa'?): 82.216.111.76#53
May 13 10:39:26 archi named[785]: lame server resolving '71.219.255.84.in-
addr.arpa' (in '71.219.255.84.in-addr.arpa'?): 84.255.216.123#53
May 13 10:39:26 archi named[785]: lame server resolving '71.219.255.84.in-
addr.arpa' (in '71.219.255.84.in-addr.arpa'?): 84.255.216.53#53
May 13 11:31:06 archi named[785]: lame server resolving
'18.240.55.80.relays.visi.com' (in 'relays.visi.com'?): 127.0.0.1#53
May 13 11:31:25 archi named[785]: lame server resolving
'18.240.55.80.relays.visi.com' (in 'relays.visi.com'?): 127.0.0.1#53
May 13 11:45:38 archi named[785]: lame server resolving
'6.59.63.84.relays.visi.com' (in 'relays.visi.com'?): 127.0.0.1#53
May 13 11:46:07 archi named[785]: lame server resolving
'6.59.63.84.relays.visi.com' (in 'relays.visi.com'?): 127.0.0.1#53
May 13 11:52:09 archi named[785]: lame server resolving '1.72.145.82.in-
addr.arpa' (in '72.145.82.in-addr.arpa'?): 213.155.179.2#53
May 13 11:52:23 archi named[785]: lame server resolving
'logserv.wi.tuniv.szczecin.pl' (in 'wi.tuniv.szczecin.pl'?): 213.155.179.2#53
May 13 11:52:23 archi named[785]: lame server resolving
'gamma.wi.tuniv.szczecin.pl' (in 'wi.tuniv.szczecin.pl'?): 213.155.179.2#53
May 13 11:52:41 archi named[785]: lame server resolving
'wifi.wi.tuniv.szczecin.pl' (in 'wi.tuniv.szczecin.pl'?): 212.14.28.44#53
May 13 11:52:41 archi named[785]: lame server resolving
'wifi.wi.tuniv.szczecin.pl' (in 'wi.tuniv.szczecin.pl'?): 213.155.179.2#53
May 13 11:52:42 archi named[785]: lame server resolving
'eta.wi.tuniv.szczecin.pl' (in 'wi.tuniv.szczecin.pl'?): 213.155.179.2#53
May 13 11:52:42 archi named[785]: lame server resolving '9.72.145.82.in-
addr.arpa' (in '72.145.82.in-addr.arpa'?): 212.14.28.44#53
May 13 11:52:42 archi named[785]: lame server resolving
'jota1.wi.tuniv.szczecin.pl' (in 'wi.tuniv.szczecin.pl'?): 213.155.179.2#53
May 13 11:52:42 archi named[785]: lame server resolving
'jota2.wi.tuniv.szczecin.pl' (in 'wi.tuniv.szczecin.pl'?): 213.155.179.2#53
May 13 11:52:43 archi named[785]: lame server resolving '11.72.145.82.in-
addr.arpa' (in '72.145.82.in-addr.arpa'?): 212.14.28.44#53
May 13 11:52:43 archi named[785]: lame server resolving '12.72.145.82.in-
addr.arpa' (in '72.145.82.in-addr.arpa'?): 212.14.28.44#53
May 13 11:52:43 archi named[785]: lame server resolving
'rdp.wi.tuniv.szczecin.pl' (in 'wi.tuniv.szczecin.pl'?): 213.155.179.2#53
May 13 11:52:44 archi named[785]: lame server resolving '17.72.145.82.in-
addr.arpa' (in '72.145.82.in-addr.arpa'?): 212.14.28.44#53
May 13 11:52:44 archi named[785]: lame server resolving
'jota5.wi.tuniv.szczecin.pl' (in 'wi.tuniv.szczecin.pl'?): 213.155.179.2#53
May 13 11:52:45 archi named[785]: lame server resolving '21.72.145.82.in-
addr.arpa' (in '72.145.82.in-addr.arpa'?): 212.14.28.44#53
May 13 11:52:45 archi named[785]: lame server resolving
'beta2.wi.tuniv.szczecin.pl' (in 'wi.tuniv.szczecin.pl'?): 213.155.179.2#53
```

```
May 13 11:52:45 archi named[785]: lame server resolving '25.72.145.82.in-
addr.arpa' (in '72.145.82.in-addr.arpa'?): 212.14.28.44#53
May 13 11:52:46 archi named[785]: lame server resolving
'jota7.wi.tuniv.szczecin.pl' (in 'wi.tuniv.szczecin.pl'?): 213.155.179.2#53
May 13 11:52:46 archi named[785]: lame server resolving '28.72.145.82.in-
addr.arpa' (in '72.145.82.in-addr.arpa'?): 212.14.28.44#53
May 13 11:52:46 archi named[785]: lame server resolving
'epsilon7.wi.tuniv.szczecin.pl' (in 'wi.tuniv.szczecin.pl'?): 213.155.179.2#53
May 13 11:52:47 archi named[785]: lame server resolving '32.72.145.82.in-
addr.arpa' (in '72.145.82.in-addr.arpa'?): 212.14.28.44#53
May 13 11:52:47 archi named[785]: lame server resolving
'epsilon5.wi.tuniv.szczecin.pl' (in 'wi.tuniv.szczecin.pl'?): 213.155.179.2#53
May 13 11:52:48 archi named[785]: lame server resolving '35.72.145.82.in-
addr.arpa' (in '72.145.82.in-addr.arpa'?): 212.14.28.44#53
May 13 11:52:48 archi named[785]: lame server resolving '36.72.145.82.in-
addr.arpa' (in '72.145.82.in-addr.arpa'?): 212.14.28.44#53
May 13 11:52:54 archi named[785]: lame server resolving '37.72.145.82.in-
addr.arpa' (in '72.145.82.in-addr.arpa'?): 212.14.28.44#53
May 13 11:53:03 archi named[785]: lame server resolving
'omicron.wi.tuniv.szczecin.pl' (in 'wi.tuniv.szczecin.pl'?): 213.155.179.2#53
May 13 11:53:04 archi named[785]: lame server resolving '43.72.145.82.in-
addr.arpa' (in '72.145.82.in-addr.arpa'?): 212.14.28.44#53
May 13 11:53:04 archi named[785]: lame server resolving
'epsilon1.wi.tuniv.szczecin.pl' (in 'wi.tuniv.szczecin.pl'?): 213.155.179.2#53
May 13 11:53:05 archi named[785]: lame server resolving
'apc.wi.tuniv.szczecin.pl' (in 'wi.tuniv.szczecin.pl'?): 213.155.179.2#53
May 13 11:53:05 archi named[785]: lame server resolving '47.72.145.82.in-
addr.arpa' (in '72.145.82.in-addr.arpa'?): 212.14.28.44#53
May 13 11:53:05 archi named[785]: lame server resolving
'hp5412.wi.tuniv.szczecin.pl' (in 'wi.tuniv.szczecin.pl'?): 213.155.179.2#53
May 13 11:55:57 archi named[785]: lame server resolving
'214.5.174.81.relays.visi.com' (in 'relays.visi.com'?): 127.0.0.1#53
May 13 12:39:36 archi named[785]: lame server resolving 'img442.imageshack.us'
(in 'imageshack.us'?): 70.86.168.170#53
May 13 13:00:50 archi named[785]: lame server resolving
'www.wi.tuniv.szczecin.pl' (in 'wi.tuniv.szczecin.pl'?): 212.14.28.44#53
 Zakończono
************************************************
Array
(
    [May 13 06] => 1
    [May 13 07] => 45
    [May 13 08] => 28
    [May 13 09] => 25
    [May 13 10] => 17
    [May 13 11] => 36
    [May 13 12] => 1
    [May 13 13] => 1
    [Total] => 154
)
```

Wynik dla wielu dni z dokładnością do godzin:

```
Array
(
    [May 6 06] => 25
    [May 6 07] => 53
    [May 6 08] => 41
    [May 6 09] => 55
    [May 6 10] => 42
    [May 6 11] => 65
    [May 6 12] => 45
    [May 6 13] => 27
    [May 6 14] => 7
    [May 6 15] => 3
    [May 6 16] => 3
    [May 6 17] => 8
```

```
[May 6 18] => 3
[May 6 19] => 10
[May 6 21] => 17
[May 6 22] => 12
[May 6 23] => 48
[May 7 00] => 50
[May 7 01] => 35
[May 7 02] => 34
[May 7 03] => 45
[May 7 04] => 48
[May 7 05] => 50
[May 7 06] => 62
[May 7 07] => 34
[May 7 08] => 18
[May 7 09] => 30
[May 7 10] => 25
[May 7 11] => 8
[May 7 12] => 12
[May 7 13] => 6
[May 7 14] => 4
[May 7 15] => 2
[May 7 16] => 18
[May 7 17] => 10
[May 7 18] => 3
[May 7 19] => 4
[May 7 20] => 8
[May 7 21] => 14
[May 7 22] => 4
[May 7 23] => 16
[May 8 00] => 37
[May 8 01] => 26
[May 8 02] => 24
[May 8 03] => 15
[May 8 04] => 18
[May 8 05] => 29
[May 8 06] => 48
[May 8 07] => 28
[May 8 08] => 39
[May 8 09] => 14
[May 8 10] => 9
[May 8 11] => 36
[May 8 12] => 30
[May 8 13] => 19
[May 8 14] => 23
[May 8 16] => 14
[May 8 17] => 12
[May 8 18] => 6
[May 8 19] => 19
[May 8 20] => 19
[May 8 21] => 11
[May 8 22] => 10
[May 8 23] => 43
[May 9 00] => 34
[May 9 01] => 33
[May 9 02] => 23
[May 9 03] => 19
[May 9 04] => 24
[May 9 05] => 45
[May 9 06] => 33
[May 9 07] => 22
[May 9 08] => 14
[May 9 09] => 22
[May 9 10] => 14
[May 9 11] => 36
[May 9 12] => 19
[May 9 13] => 39
[May 9 14] => 11
```

```
[May 9 15] => 10
[May 9 16] => 47
[May 9 17] => 32
[May 9 18] => 6
[May 9 19] => 23
[May 9 20] => 41
[May 9 21] => 18
[May 9 22] => 4
[May 9 23] => 14
[May 10 00] => 8
[May 10 01] => 2
[May 10 02] => 5
[May 10 03] => 14
[May 10 04] => 4
[May 10 05] => 11
[May 10 06] => 24
[May 10 07] => 9
[May 10 08] => 17
[May 10 09] => 16
[May 10 10] => 12
[May 10 11] => 1
[May 10 12] => 11
[May 10 13] => 20
[May 10 14] => 13
[May 10 15] => 7
[May 10 16] => 32
[May 10 17] => 22
[May 10 18] => 13
[May 10 19] => 8
[May 10 20] => 12
[May 10 21] => 18
[May 10 22] => 9
[May 10 23] => 35
[May 11 00] => 19
[May 11 01] => 28
[May 11 02] => 11
[May 11 03] => 23
[May 11 04] => 14
[May 11 05] => 20
[May 11 06] => 25
[May 11 07] => 22
[May 11 08] => 7
[May 11 09] => 11
[May 11 10] => 15
[May 11 11] => 45
[May 11 12] => 25
[May 11 13] => 15
[May 11 14] => 17
[May 11 15] => 5
[May 11 16] => 3
[May 11 17] => 4
[May 11 18] => 8
[May 11 19] => 6
[May 11 20] => 15
[May 11 21] => 54
[May 11 22] => 13
[May 11 23] => 24
[May 12 00] => 58
[May 12 01] => 36
[May 12 02] => 35
[May 12 03] => 54
[May 12 04] => 9
[May 12 05] => 14
[May 12 06] => 38
[May 12 07] => 41
[May 12 08] => 54
[May 12 09] => 36
```

```
    [May 12 10] => 21
    [May 12 11] => 2
    [May 12 12] => 14
    [May 12 13] => 8
    [May 12 14] => 4
    [May 12 16] => 7
    [May 12 17] => 2
    [May 12 18] => 2
    [May 12 20] => 6
    [May 12 21] => 12
    [May 12 22] => 9
    [May 12 23] => 23
    [May 13 00] => 59
    [May 13 01] => 63
    [May 13 02] => 37
    [May 13 03] => 65
    [May 13 04] => 43
    [May 13 05] => 40
    [May 13 06] => 21
    [Total] => 3681
)
```

Wynik dla wielu dni z dokładnością do jednego dnia:

```
Array
(
    [May 6] => 464
    [May 7] => 540
    [May 8] => 529
    [May 9] => 583
    [May 10] => 323
    [May 11] => 429
    [May 12] => 485
    [May 13] => 328
    [Total] => 3681
)
```

Konfiguracja syslog-ng

```
#
# Configuration file for syslog-ng under Debian
#
# attempts at reproducing default syslog behavior

# the standard syslog levels are (in descending order of priority):
# emerg alert crit err warning notice info debug
# the aliases "error", "panic", and "warn" are deprecated
# the "none" priority found in the original syslogd configuration is
# only used in internal messages created by syslogd


######
# options

options {
        # disable the chained hostname format in logs
        # (default is enabled)
        chain_hostnames(0);
```

```
        # the time to wait before a died connection is re-established
        # (default is 60)
        time_reopen(10);

        # the time to wait before an idle destination file is closed
        # (default is 60)
        time_reap(360);

        # the number of lines buffered before written to file
        # you might want to increase this if your disk isn't catching with
        # all the log messages you get or if you want less disk activity
        # (say on a laptop)
        # (default is 0)
        #sync(0);

        # the number of lines fitting in the output queue
        log_fifo_size(2048);

        # enable or disable directory creation for destination files
        create_dirs(yes);

        # default owner, group, and permissions for log files
        # (defaults are 0, 0, 0600)
        #owner(root);
        group(adm);
        perm(0640);

        # default owner, group, and permissions for created directories
        # (defaults are 0, 0, 0700)
        #dir_owner(root);
        #dir_group(root);
        dir_perm(0755);

        # enable or disable DNS usage
        # syslog-ng blocks on DNS queries, so enabling DNS may lead to
        # a Denial of Service attack
        # (default is yes)
        use_dns(no);

        # maximum length of message in bytes
        # this is only limited by the program listening on the /dev/log Unix
        # socket, glibc can handle arbitrary length log messages, but -- for
        # example -- syslogd accepts only 1024 bytes
        # (default is 2048)
        #log_msg_size(2048);
};


######
# sources

# all known message sources
source s_all {
        # message generated by Syslog-NG
        internal();
        # standard Linux log source (this is the default place for the syslog()
        # function to send logs to)
        unix-stream("/dev/log");
        # messages from the kernel
        file("/proc/kmsg" log_prefix("kernel: "));
        # use the above line if you want to receive remote UDP logging messages
        # (this is equivalent to the "-r" syslogd flag)
        udp();
};


######
```

```
# destinations

# some standard log files
destination df_auth { file("/var/log/auth.log"); };
destination df_syslog { file("/var/log/syslog"); };
destination df_cron { file("/var/log/cron.log"); };
destination df_daemon { file("/var/log/daemon.log"); };
destination df_kern { file("/var/log/kern.log"); };
destination df_lpr { file("/var/log/lpr.log"); };
destination df_mail { file("/var/log/mail.log"); };
destination df_user { file("/var/log/user.log"); };
destination df_uucp { file("/var/log/uucp.log"); };

# these files are meant for the mail system log files
# and provide re-usable destinations for {mail,cron,...}.info,
# {mail,cron,...}.notice, etc.
destination df_facility_dot_info { file("/var/log/$FACILITY.info"); };
destination df_facility_dot_notice { file("/var/log/$FACILITY.notice"); };
destination df_facility_dot_warn { file("/var/log/$FACILITY.warn"); };
destination df_facility_dot_err { file("/var/log/$FACILITY.err"); };
destination df_facility_dot_crit { file("/var/log/$FACILITY.crit"); };

# these files are meant for the news system, and are kept separated
# because they should be owned by "news" instead of "root"
destination df_news_dot_notice { file("/var/log/news/news.notice"
owner("news")); };
destination df_news_dot_err { file("/var/log/news/news.err" owner("news")); };
destination df_news_dot_crit { file("/var/log/news/news.crit" owner("news"));
};

# some more classical and useful files found in standard syslog configurations
destination df_debug { file("/var/log/debug"); };
destination df_messages { file("/var/log/messages"); };

# pipes
# a console to view log messages under X
destination dp_xconsole { pipe("/dev/xconsole"); };

# consoles
# this will send messages to everyone logged in
destination du_all { usertty("*"); };

destination iptables { file("/var/log/iptables.log" owner("root") group("adm")
perm(0640)); };


######
# filters

# all messages from the auth and authpriv facilities
filter f_auth { facility(auth, authpriv); };

# all messages except from the auth and authpriv facilities
filter f_syslog { not facility(auth, authpriv); };

# respectively: messages from the cron, daemon, kern, lpr, mail, news, user,
# and uucp facilities
filter f_cron { facility(cron); };
filter f_daemon { facility(daemon); };
filter f_iptables { facility(kern) and match("IN=[A-Za-z0-9]* OUT=[A-Za-z0-
9]*"); };
filter f_kern { facility(kern); };
filter f_lpr { facility(lpr); };
filter f_mail { facility(mail); };
filter f_news { facility(news); };
filter f_user { facility(user); };
filter f_uucp { facility(uucp); };
```

```
# some filters to select messages of priority greater or equal to info, warn,
# and err
# (equivalents of syslogd's *.info, *.warn, and *.err)
filter f_at_least_info { level(info..emerg); };
filter f_at_least_notice { level(notice..emerg); };
filter f_at_least_warn { level(warn..emerg); };
filter f_at_least_err { level(err..emerg); };
filter f_at_least_crit { level(crit..emerg); };

# all messages of priority debug not coming from the auth, authpriv, news, and
# mail facilities
filter f_debug { level(debug) and not facility(auth, authpriv, news, mail); };

# all messages of info, notice, or warn priority not coming form the auth,
# authpriv, cron, daemon, mail, and news facilities
filter f_messages {
        level(info,notice,warn)
            and not facility(auth,authpriv,cron,daemon,mail,news);
};

# messages with priority emerg
filter f_emerg { level(emerg); };

# complex filter for messages usually sent to the xconsole
filter f_xconsole {
    facility(daemon,mail)
        or level(debug,info,notice,warn)
        or (facility(news)
                and level(crit,err,notice));
};


######
# logs
# order matters if you use "flags(final);" to mark the end of processing in a
# "log" statement

# these rules provide the same behavior as the commented original syslogd rules

# auth,authpriv.*                 /var/log/auth.log
log {
        source(s_all);
        filter(f_auth);
        destination(df_auth);
};

# *.*;auth,authpriv.none          -/var/log/syslog
log {
        source(s_all);
        filter(f_syslog);
        destination(df_syslog);
};

# this is commented out in the default syslog.conf
# cron.*                          /var/log/cron.log
#log {
#        source(s_all);
#        filter(f_cron);
#        destination(df_cron);
#};

# daemon.*                        -/var/log/daemon.log
log {
        source(s_all);
        filter(f_daemon);
        destination(df_daemon);
```

```
};

# kern.*                          -/var/log/kern.log
log {
        source(s_all);
        filter(f_kern);
        destination(df_kern);
};

# lpr.*                           -/var/log/lpr.log
log {
        source(s_all);
        filter(f_lpr);
        destination(df_lpr);
};

# mail.*                          -/var/log/mail.log
log {
        source(s_all);
        filter(f_mail);
        destination(df_mail);
};

# user.*                          -/var/log/user.log
log {
        source(s_all);
        filter(f_user);
        destination(df_user);
};

# uucp.*                          /var/log/uucp.log
log {
        source(s_all);
        filter(f_uucp);
        destination(df_uucp);
};

# mail.info                       -/var/log/mail.info
log {
        source(s_all);
        filter(f_mail);
        filter(f_at_least_info);
        destination(df_facility_dot_info);
};

# mail.warn                       -/var/log/mail.warn
log {
        source(s_all);
        filter(f_mail);
        filter(f_at_least_warn);
        destination(df_facility_dot_warn);
};

# mail.err                        /var/log/mail.err
log {
        source(s_all);
        filter(f_mail);
        filter(f_at_least_err);
        destination(df_facility_dot_err);
};

# news.crit                       /var/log/news/news.crit
log {
        source(s_all);
        filter(f_news);
        filter(f_at_least_crit);
        destination(df_news_dot_crit);
```

```
};

# news.err                        /var/log/news/news.err
log {
        source(s_all);
        filter(f_news);
        filter(f_at_least_err);
        destination(df_news_dot_err);
};

# news.notice                     /var/log/news/news.notice
log {
        source(s_all);
        filter(f_news);
        filter(f_at_least_notice);
        destination(df_news_dot_notice);
};


# *.=debug;\
#         auth,authpriv.none;\
#         news.none;mail.none     -/var/log/debug
log {
        source(s_all);
        filter(f_debug);
        destination(df_debug);
};


# *.=info;*.=notice;*.=warn;\
#         auth,authpriv.none;\
#         cron,daemon.none;\
#         mail,news.none          -/var/log/messages
log {
        source(s_all);
        filter(f_messages);
        destination(df_messages);
};

# *.emerg                         *
log {
        source(s_all);
        filter(f_emerg);
        destination(du_all);
};


# daemon.*;mail.*;\
#         news.crit;news.err;news.notice;\
#         *.=debug;*.=info;\
#         *.=notice;*.=warn       |/dev/xconsole
log {
        source(s_all);
        filter(f_xconsole);
        destination(dp_xconsole);
};

log {
       source(s_all);
       filter(f_iptables);
       destination(iptables);
};
```