STUDIA INFORMATICA 2003
Volume 24 Number 2B (54)

Rafał CICHOCKI
Akademia Morska, Katedra Podstaw Informatyki i Sieci Komputerowych

# BEZPIECZEŃSTWO DANYCH W SYSTEMACH ROZPROSZONYCH – SYSTEMY WYKRYWANIA INTRUZÓW

**Streszczenie.** W obliczu rosnącego zagrożenia oraz nieustannie ewoluujących metod ataków kluczowym zagadnieniem jest wykrywanie każdego przejawu szkodliwej działalności zarówno w zawartości pakietów (analiza sygnatur), jak i w strukturze pakietu (analiza protokołów) czy też anormalnej aktywności użytkowników oraz aplikacji (wykrywanie anomalii). W referacie dokonano próby klasyfikacji Systemów Wykrywania Intruzów (*Intrusion Detection Systems – IDS*), które w ostatnich latach stały się integralnym i bazowym składnikiem systemów bezpieczeństwa sieci komputerowych.

**Słowa kluczowe:** bezpieczeństwo danych, systemy wykrywania intruzów, detekcja anomalii, sieci komputerowe.

# DATA SECURITY IN DISTRIBUTED SYSTEMS – INTRUSION DETECTION SYSTEMS

**Summary.** In the face of growing threat and continuously evolving attacks methods the key issue is to detect and identify symptoms of harmful activity: both packet payload (signature analysis), packet structure (protocol analysis), abnormal user or application activity. This paper is an attempt to classify Intrusion Detection Systems, which are crucial component of any network defense strategy.

**Keywords:** data security, intrusion detection systems, anomaly detection, computer network.

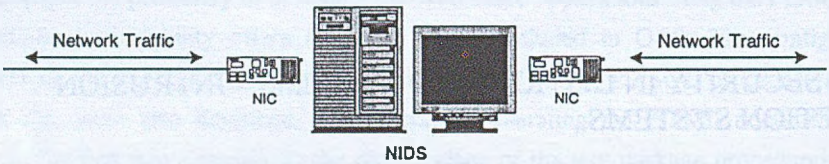
## 1. Wstęp

Bezpieczeństwo danych w sieciach rozległych stanowi jedno z najistotniejszych zagadnień, z którymi spotykają się użytkownicy sieci niezależnie od tego, czy mowa jest o użytkownikach indywidualnych, instytucjonalnych czy komercyjnych. W ostatnich latach

Systemy Detekcji Intruzów (IDS – *Intrusion Detection Systems*) stały się zagadnieniem kluczowym strategii budowania bezpieczeństwa systemów komputerowych. Początki systemów IDS sięgają kilkunastu lat wstecz, a pierwsze z nich umożliwiały analizę logów systemowych w celu wykrycia w nich anomalii związanych z aktywnością użytkowników oraz pracą systemu. Do lat obecnych systemy te przeszły długą drogę, stając się elementarnym składnikiem systemu bezpieczeństwa. W sercu tych systemów leży moduł analizujący każdy otrzymany pakiet sprawdzający, czy zawiera on niebezpieczne informacje oraz wysyła alarm, jeśli jest to konieczne. Droga, jaką przeszły systemy ID, zaowocowała dużą różnorodnością rozwiązań oraz podejść do problemu. Niniejszy artykuł stanowi przegląd technik używanych we współczesnych systemach bezpieczeństwa wykorzystujących systemy wykrywania intruzów.

## 2. Systemy wykrywania intruzów

Zasada funkcjonowania systemów IDS jest bardzo prosta, system podobnie jak klasyczne firewall'e poddaje analizie każdy przychodzący i wychodzący pakiet w celu wykrycia anomalii struktury lub potencjalnie niebezpiecznej zawartości. W tym celu systemy ID wykorzystują dwie odmienne techniki: analizę sygnatur oraz analizę protokołów. Ze względu na umiejscowienie i zadania systemu ID wyróżnia się dwa typy: Host-Based (HIDS) i Network-Based(NIDS).




Rys. 1. Umiejscowienie NIDS w systemie
Fig. 1. NIDS in the network system

Techniki wykorzystywane przez systemy wykrywania intruzów były niedawno przedmiotem publicznej dyskusji na forum wortalu SecurityFocus.com [8] poświęconego w całości zagadnieniom związanym z bezpieczeństwem IT. Wortal ten skupia wokół siebie uznane autorytety IT Security, prowadzi również archiwa słynnej już grupy dyskusyjnej BugTraq.

### 2.1. Analiza Sygnatur

Analiza sygnatur była pierwszą metodą zastosowaną w systemach ID. Metoda ta określana jest jako dopasowanie napisów (*string matching* często: *pattern matching*). Kiedy pakiet wczytywany jest do pamięci, zostaje porównany bajt po bajcie z pojedynczą sygnaturą charakteryzującą złośliwy (niechciany) ruch. Taka sygnatura może zawierać kluczową frazę lub

polecenie, które wskazują na atak. Jeśli takie dopasowanie zostanie znalezione, generowany jest alarm, w przeciwnym wypadku pakiet zostaje porównany z kolejną sygnaturą. Jeśli żadne dopasowanie nie zostanie znalezione, pakiet uznany zostanie za nieszkodliwy i przesłany dalej. Jak zauważa Matthew Tanase [4], metoda ta, określana jako "packet grepping" w odniesieniu do Unixowego polecenia *grep*, jest elementarna i może zostać z powodzeniem zaimplementowana przy wykorzystaniu prostego polecenia systemów Unix.

### 2.2. Analiza Protokołów

Druga z metod wykorzystuje specyfikację techniczną protokołów (RFC) w celu wykrycia anomalii w strukturze pakietu. Ponieważ każdy z pakietów zawiera informacje specyficzne dla poszczególnych warstw sieciowych, a atak może być przeprowadzony (lub wykryty) na poziomie każdej z nich, system wykrywania intruzów musi najpierw zdekodować pakiet odczytując i analizując informacje z każdej z warstw. W każdej z warstw zawarte są informacje (znaczniki), które powinny zawierać informacje zgodne ze standardem RFC. Każdy ze znaczników analizowanego pakietu powinien więc zawierać ściśle określoną informację. Wykrycie zawartości znacznika niezgodnego z RFC lub posiadającego informacje nie przewidziane w stosownym RFC zostaje uznany za pakiet zawierający szkodliwe treści. Systemy ID dokonują analizy każdego pola poszczególnych protokołów: IP, TCP oraz UDP. Jakakolwiek odchyłka od standardu powoduje wygenerowanie alarmu. Warto w tym miejscu zauważyć, że metoda ta umożliwia potencjalne wykrycie nieznanej dotychczas metody ataku (tzw. *zero days exploits*), co nie jest możliwe w przypadku metody opartej na analizie sygnatur.

Jakkolwiek obie metody posiadają swoje mocne i słabe strony, szybko zostało zauważone, iż uzupełniają się one. I rzeczywiście, większość dostępnych systemów opartych na analizie sygnatur potrafi dokonać analizy podstawowych informacji (adresy źródłowy i docelowy, flagi TCP) zawartych w warstwach trzeciej i czwartej modelu odniesienia ISO/OSI, a więc protokołów IP, TCP i UDP. Przykładem może być reguła SNORT`a:

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:
      "WEB-IIS File permission cannonalization"; uricon-
      tent: "/scripts/..%c1%9../"; flags A+; nocase;
      sid:983; rev:1;)
```

Reguła ta analizuje zawartość pakietu kierowanego do serwera WWW w poszukiwaniu sygnatury ataku na serwer MS IIS, umożliwiającego wykonanie dowolnego polecenia na serwerze pracującym pod kontrolą systemu operacyjnego Windows poprzez uruchomienie interpretera poleceń (cmd.exe). Zauważmy, że reguła ta analizuje zawartość pakietu na poziomie warstwy aplikacyjnej (Layer 7), jak również zawartość nagłówka TCP pobierając z niego adres źródłowy, docelowy oraz numer portu.

Mocną stroną systemów opartych na analizie sygnatur jest ich duża wydajność. Proces porównywania stringów w niewielkim stopniu obciąża procesor. Jednocześnie pisanie reguł

jest proste, a same reguły przejrzyste, łatwe do analizowania i dostosowania do własnych potrzeb. Istnieje także bardzo duże wsparcie ze strony wspólnoty specjalistów. Z drugiej strony, wydajność tych systemów bardzo mocno spada w przypadku dużego rozbudowania reguł. Stanowi to istotne ograniczenie, bowiem każda nowa metoda ataku wymaga napisania nowej sygnatury umożliwiającej jej wykrycie. Również niektóre techniki ataków, takie jak np. Unicode czy wielorakie modyfikacje zbliżone do tych znalezionych w ataku poprzez SNMP community string, czy programy umożliwiające tzw. shellcode morphing, czyli przekształcanie poleceń, mogą stanowić nie lada wyzwanie dla systemów opartych na porównywaniu sygnatur.

Systemy oparte na analizie protokołów również posiadają swoje mocne i słabe strony. Większość z nich pozostaje w opozycji do systemu analizy sygnatur. Analiza protokołów jest procesem stosunkowo mocno obciążającym procesor, jednocześnie napisanie reguł jest procesem wymagającym bardzo dobrej znajomości standardów. Niestety, nie wystarczy tu wyłącznie znajomość stosownych RFC ze względu na fakt, iż różni producenci w różny sposób implementują protokoły w swoich produktach, nierzadko dokonując modyfikacji czy też odbiegając w znaczący sposób od standardu. Same reguły są również trudne do analizy. Jej podstawową zaletą jest możliwość wykrywania ataków opartych na nieznanych wcześniej technikach.

Każda z przedstawionych powyżej technik celuje w innych zastosowaniach. Wydaje się, że twórcy systemów dostrzegają ten fakt i obecnie wdrażane systemy ID wykorzystują obydwie techniki. Wiele spośród dostępnych systemów analizujących protokoły wykorzystuje również analizę zawartości pakietów na poziomie warstwy aplikacyjnej. Również systemy oparte na analizie sygnatur potrafią sprawdzić i wykorzystać informacje zawarte w warstwach transportowej i sieciowej, jak również potrafią wykryć podstawowe naruszenia standardów RFC.

## 3. Systemy Wykrywania Anomalii

Systemy wykrywania anomalii stanowią kolejny etap na drodze ewolucji systemów wykrywania intruzów. Koncepcja działania systemów AD została opisana przez Dorothy Denning [2] . Podstawowa różnica pomiędzy klasycznym systemem IDS a ADS polega na tym, iż ADS w przeciwieństwie do IDS nie szuka określonych zdefiniowanych wcześniej zjawisk, które mogą stanowić potencjalne naruszenie polityki bezpieczeństwa. Systemy AD wykorzystują zbudowany wcześniej model aktywności systemu, który określa pewną normę zachowania systemu w określonym interwale czasowym. Każde odstępstwo od zachowania „normalnego" jest uznawane za potencjalne zagrożenie bezpieczeństwa.

Model odniesienia może być zbudowany dla sieci, pojedynczego hosta, protokołu lub nawet aplikacji. Istnieją dwa zasadnicze podejścia do budowania modelu odniesienia: statistics-based oraz specification-based.

### 3.1. Podejście Statystyczne

To podejście wykorzystuje zależności statystyczne do budowy modelu odniesienia. Pierwsza faza to uczenie sensorów. Polega to na obserwacji zachowania systemu w określonym przedziale czasu. Obserwacja ta może dotyczyć ruchu w sieci, wywołań systemowych, wykorzystania aplikacji, logowań użytkowników itp. W kolejnym kroku wykorzystuje się jedną z wielu metod matematycznych do wygenerowania ilościowych miar zaobserwowanych danych. Otrzymane wyniki stanowią punkt odniesienia dla zachowania systemu. Zmienna losowa x reprezentuje ilościową miarę skumulowaną w pewnym okresie czasu. Model definiuje trzy podstawowe miary:

1. *Licznik zdarzeń – x*: reprezentuje jedno zdarzenie odpowiadające zapisowi w logach systemowych.
2. *Czasomierz – x*: odległość czasowa pomiędzy dwoma związanymi zdarzeniami, np. wejścia i wyjścia z systemu, uruchomienia i zakończenia programu.
3. *Miernik zasobów – x*: jest ilością zasobów wykorzystywanych przez obserwowane procesy w okresie czasu.

Model nie robi żadnych założeń co do rozkładu zmiennej losowej *x*, wszystko co o niej wiemy, pochodzi z obserwacji. Podstawowym problemem w tym podejściu jest wybór metody do pomiaru odchyleń. Denning [2] proponuje pięć metod:

4. *Model operacyjny* – zakłada się, iż anormalność zjawiska może być ustalona na podstawie obserwacji nowej wartości ponad ustalony limit. Poprzednie wartości *x* nie są brane pod uwagę, natomiast limit ustalany jest na podstawie obserwacji tej zmiennej w procesie budowania modelu i przyjmowany za niezmienny. Przykładem takiego podejścia może być obserwacja nieudanych logowań (złe hasło) w pewnym okresie czasu, gdzie 10 nieudanych prób oznacza atak brute-force.
5. *Model oparty na pomiarze średniej i odchylenia standardowego* – model zakłada, że wszystko co wiemy na temat $x_1, \ldots x_n$, to średnia oraz odchylenie standardowe określone:

$$sum = x_1 + \ldots + x_n$$

$$sumkwadr = x_1^2 + \ldots + x_n^2$$

$$średnia = sum / n$$

$$odchstd = sqrt\left(sumkwadr \,/\, (n+1) - średnia^2\right)$$

Nowa zaobserwowana wartość $x_{n+1}$ jest uznana za anormalną, jeśli nie zawiera się w tzw. *przedziale ufności* określonym jako

*średnia + d * odchstd*

Prawdopodobieństwo, że wartość wypadnie poza przedziałem ufności wynosi co najwyżej $1/d^2$, dla d = 4 wynosi więc 0.0625.

Model ten może być zastosowany do oceny pomiarów zdarzeń, pomiarów przedziałów czasowych, mierników zasobów skumulowanych w ustalonym przedziale czasowym lub pomiędzy dwoma zdarzeniami. Model ten posiada dwie zalety w stosunku do poprzedniego. Po pierwsze, nie wymaga znajomości „normalnej" aktywności mierzonego zjawiska w celu ustalenia przedziału ufności. Model uczy się, co oznacza normalną aktywność na podstawie bieżących obserwacji i automatycznie przedział ufności odzwierciedla stan aktualnej wiedzy na temat zjawiska. Ponieważ przedział ufności zależy od obserwowanych danych, to co jest normalne dla jednego użytkownika, może zostać uznane za nienormalne dla innego.

1. *Model wielowariancyjny* – model ten jest zbliżony do poprzedniego, przy czym bazuje na pomiarach zależności pomiędzy dwoma lub więcej wartościami. Model jest użyteczny, jeśli zaobserwowane dane sugerują, że ocena zjawisk i korelacji między nimi ogranicza w sposób istotny liczbę rozwiązań. Tak może być np. przy ocenie wykorzystania procesora na jednostkę wykorzystaną przez program, częstotliwość logowania użytkownika w odniesieniu do średniego czasu trwania sesji.
2. *Model oparty na procesach Markowa* – model wykorzystywany wyłącznie w odniesieniu do liczników zdarzeń, przyporządkowuje każdy typ zdarzenia (zapisanych w logach systemowych) jako zmienną stanu, oraz wykorzystuje macierz przejść do scharakteryzowania częstotliwości przejść między stanami. Nowa obserwacja zostaje uznana za anormalną, jeśli jego prawdopodobieństwo określone przez poprzedni stan i macierz przejść jest zbyt niskie. Ten rodzaj modelu może być wykorzystany do obserwacji wydawanych poleceń, w sytuacji kiedy ich kolejność jest ważna.
3. *Model serii czasowych* – model wykorzystuje pomiar przedziałów czasowych w połączeniu z licznikiem zdarzeń lub pomiarem zasobów. Mierzone są zarówno kolejność, przedziały czasowe, jak i wartości. Zjawisko zostaje uznane za niepożądane, jeśli prawdopodobieństwo jego wystąpienia w określonym przedziale czasowym jest zbyt niskie.

Podane modele pokrywają całe spektrum problemu detekcji anomalii, pozwalając na wykrycie każdego niepożądanego zjawiska, nawet w sytuacji dużego rozbudowania systemu, poprzez obserwacje wielu różnorodnych zjawisk i korelacji pomiędzy nimi. Największym problemem w występującym przy budowaniu modelu odniesienia przy wykorzystaniu metod statystycznych jest zmienność środowiska, które model ma odzwierciedlać. Każda jego zmiana wymaga uaktualnienia modelu, co w rozbudowanych systemach może prowadzić do konieczności ciągłego treningu modelu.

### 3.2. Podejście Specyfikacyjne

Calvin, Ruschitzka i Levitt [1] opisali inne podejście do budowy modelu. Podejście to nie bazuje na obserwacji zależności statystycznych, ale na logicznym opisie normalnej aktywności znaczących z punktu widzenia bezpieczeństwa systemu aplikacji. Większość ze spotykanych problemów bezpieczeństwa hosta związanych jest bezpośrednio lub pośrednio z pojedynczymi programami pracującymi z wysokim poziomem uprzywilejowania. W pierwszej fazie budowany jest model odzwierciedlający cztery najistotniejsze aspekty funkcjonowania aplikacji:

1. *Dostęp do obiektów systemowych* – np. program `finger` pracujący w niektórych systemach w trybie suid, w trakcie normalnej pracy program ten wykorzystuje plik: /usr/bin/finger i w niektórych sytuacjach czyta pliki .plan .profile i /var/log/utmp. Próba dostępu do jakiegokolwiek innego pliku oznacza anormalne działanie aplikacji i prawdopodobnie próbę wykorzystania atrybutu suid, np. w ataku typu *buffer-overflow.*
2. *Sekwencyjność* – pewne zdarzenia w systemie zachodzą w ściśle ustalonym porządku, np. rejestracja użytkownika w systemie zawsze oznacza, że program login czyta plik /etc/passwd, uwierzytelnia użytkownika, czyta jego ustawienia środowiskowe, a następnie uruchamia /bin/sh i ostatecznie przekazuje sterowanie użytkownikowi. Jakiekolwiek zakłócenie tego porządku oznaczać może próbę włamania.
3. *Synchronizacja* – trywialnym przykładem problemów z synchronizacją jest wykonanie przez użytkownika polecenia `passwd`, w trakcie kiedy administrator edytuje plik /etc/shadow, problemy tego typu mają duże znaczenie w systemach rozproszonych, tam gdzie prawidłowy przebieg procesu wymaga synchronizacji wielu procesów na wielu maszynach.
4. *Race conditions* – specjalny przypadek problemu synchronizacji, jeśli program posiada jakieś wady, tego typu atakujący może zakłócić zachowanie programu wykonując określone czynności w trakcie jego pracy. Wykrycie tego typu przypadków wymaga nie tylko obserwowania samego programu, ale także pozostałych procesów systemowych.

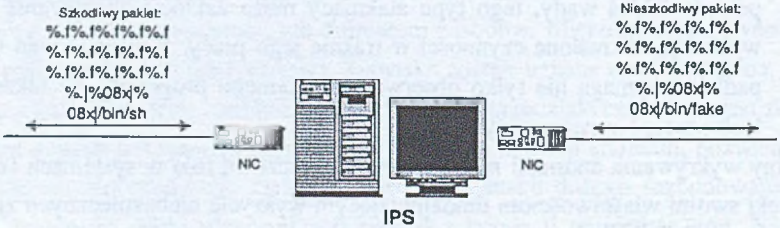
Systemy wykrywania anomalii mogą odgrywać kluczową rolę w systemach bezpieczeństwa. Dzięki swoim właściwościom umożliwiającym wykrycie niebezpiecznych zjawisk począwszy od anomalii ruchu sieciowego, poprzez zabezpieczanie systemu operacyjnego i aplikacji, po śledzenie poczynań użytkowników, systemy te sprawdzają się wszędzie tam, gdzie bezpieczeństwo odgrywa kluczową rolę. Dzieje się tak ze względu na fakt, iż umożliwiają wykrycie wszelkich przejawów szkodliwej działalności, co nie jest bez znaczenia, w sytuacji kiedy te ostatnie stają się coraz bardziej zaawansowane.

## 4. Systemy Zapobiegania Intruzom

Systemy tego rodzaju potrafią nie tylko wykryć, ale również zatrzymać wszelką aktywność związaną z wykrytą próbą ataku zarówno znanego, jak i nieznanego typu. Idąc dalej, systemy te posiadają również możliwość przechwycenia ataku i generowania fałszywych odpowiedzi, dając atakującemu zwodnicze poczucie, iż jego wysiłki odniosły sukces, jednocześnie dając administratorowi możliwość poznania stosowanej metody ataku. Istnieje kilka rodzajów tego typu systemów, każdy z nich łączy w sobie tradycyjne systemy ID/AD z technikami stosowanymi w firewall'ach aplikacyjnych. W chwili obecnej wyróżnia się pięć podstawowych technik budowania systemów IP.

### 4.1. Inline Network Intrusion Detection System

Umiejscowienie tego typu systemów zostało pokazane na ilustracji. Typowy system tego rodzaju posiada trzy interfejsy. Dwa sposórd nich przekazują pakiety przechodzące przez sieć poddając je ówczesnej analizie pod kątem wykrycia potencjalnie szkodliwej zawartości. Jeśli takowa zostanie wykryta, pakiet zostaje odrzucony, w przeciwnym wypadku jest przepuszczany. Możliwe jest również logowanie pakietów zamiast ich blokowania. Rozszerzenia tego systemu pozwalają na dokonywanie modyfikacji wewnątrz pakietu. Zawartość pakietu zostaje przepisana do nowego pakietu (pozbawionego ewentualnych złośliwych zmian w strukturze samego pakietu, co, jak wiadomo, może samo w sobie stanowić technikę ataku) po uprzednim zmodyfikowaniu lub usunięciu szkodliwej zawartości. W efekcie pakiet dociera do punktu docelowego, nie czyniąc żadnych szkód, jednocześnie atakujący nie ma pojęcia, że atak się nie powiódł i został zarejestrowany. Dzięki temu, kiedy będzie go kontynuował, możliwe jest zapoznanie się ze szczegółami wykorzystanej przez intruza techniki. Wadą systemów Inline NIDS jest fakt, iż systemy te w przypadku uszkodzenia lub złej konfiguracji mogą zatrzymywać cały ruch w segmencie sieci, w którym się znajdują.




Rys. 2. Inline NIDS
Fig. 2. Inline NIDS

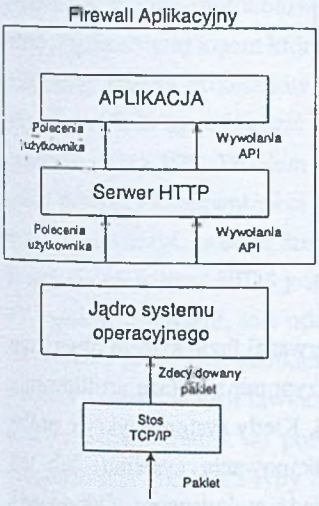
W przypadku systemów, w których nie przerwane działanie jest zagadnieniem kluczowym,może to stanowić poważny problem. Główną zaletą jest to, że system ten może być posadowiony przed elementem, który ma być chroniony, co pozwala zabezpieczyć systemy,

w przypadku których ingerencja w ich funkcjonowanie jest utrudniona (np. AS400 czy systemy mainframe).

### 4.2. Przełączniki warstwy aplikacji

Przełączniki warstwy aplikacji pojawiły się w odpowiedzi na coraz większe wymagania odnośnie do pasma, jak również ze względu na potrzebę równoważenia obciążeń poszczególnych serwerów. Aby tym potrzebom sprostać, należało wyposażyć te urządzenia w możliwości analizowania zawartości pakietów na poziomie warstwy siódmej. Dzięki temu możliwe jest np. przekierowanie połączenia na wybrany serwer www w zależności od tego, jaki URL został wpisany. Stąd już tylko krok do zaimplementowania reguł, które np. zapewniałyby ochronę przed atakami typu DoS czy DDoS. Zasada działania tych urządzeń jest zbliżona do INIDS pracującego w oparciu o analizę sygnatur.

### 4.3. Firewall aplikacyjny/IDS



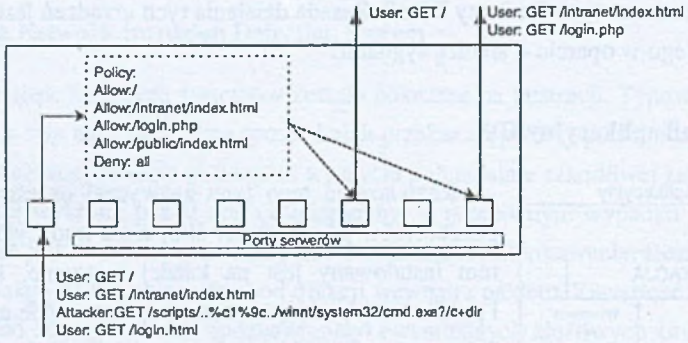

Rys. 3. Firewall Aplikacyjny
Fig. 3. Application Firewall

Rozwiązania tego typu zazwyczaj określane są przez producentów jako systemy IPS, a nie tradycyjne IDS. System instalowany jest na każdej maszynie, którą chroni i profilowany pod kątem każdej aplikacji. Nie analizuje jednak zawartości pakietów, sprawdza natomiast wywołania API, zarządzanie pamięcią (np. w celu wykrycia ataków typu *buffer-overflow*), interakcje aplikacji z systemem i użytkownika z aplikacją. Pozwala to zabezpieczyć się zarówno przed źle napisanymi programami, jak i wykryć ataki nieznanego typu. We wstępnej fazie dokonuje się tzw. profilowania systemu. Na podstawie zaobserwowanej interakcji aplikacja – system operacyjny – użytkownik tworzony jest profil aplikacji, która ma być chroniona.

System działa na zasadzie odrzucającej wszelkie zachowania poza jawnie zezwolonymi. Oznacza to konieczność bardzo dokładnego przebadania aplikacji, w przeciwnym wypadku jej funkcjonalność może zostać ograniczona. Profilowanie trzeba powtarzać po każdej aktualizacji aplikacji bądź też systemu operacyjnego, co stanowi pewną wadę, szczególnie w dynamicznie zmieniających się środowiskach. Mocną stroną systemu jest jego elastyczność i możliwość dostosowania do każdego systemu czy aplikacji. Dzięki temu, iż system zainstalowany jest na każdej chronionej maszynie, nie ma obaw, że źle napisane lub niedostosowane do konkretnego systemu reguły ograniczą skuteczność systemu, lub zakłócą jego funkcjonowanie.

### 4.4. Przełączniki hybrydowe

Wykorzystują typowe dla firewall'i aplikacyjnych metody analizy zawartości pakietów umożliwiając jednocześnie dokonywanie przełączania pakietów. Są więc, jak sama nazwa wskazuje, hybrydą obydwu rozwiązań. Typowe umiejscowienie tych urządzeń w architekturze systemu jest takie samo jak w przypadku przełączników aplikacyjnych, jednak przełączniki hybrydowe mają pełną wiedzę nie tylko na temat np. serwera WWW, ale również aplikacji na nim działających. Wiedza ta pozwala na skuteczne zabezpieczenie się przed atakiem skierowanym do konkretnej aplikacji, zanim jeszcze pakiet trafi na chronioną maszynę. Wykorzystują one politykę typową dla standardowych zapór ogniowych, odrzucając wszystko, co nie zostało wprost (odpowiednimi regułami) dozwolone.




Rys. 4. Przełącznik hybrydowy
Fig. 4. Hybrid switch

### 4.5. Fałszywe Aplikacje

Technika odmienna od dotychczas prezentowanych. W pierwszej fazie system obserwuje sieć w celu rozpoznania jej prawidłowego funkcjonowania. Przypomina to fazę profilowania wykorzystywaną w zaporach ogniowych warstwy aplikacyjnej. Kiedy system wykryje próbę połączenia się z nieistniejącym serwisem (np. w trakcie skanowania systemu) lub też anormalne połączenie z istniejącym serwisem, wyśle odpowiedź atakującemu. Odpowiedź systemu zawierać będzie „marker" w postaci fałszywych danych. Kiedy atakujący pojawi się ponownie próbując wykorzystać zdobytą wiedzę, system rozpozna „marker" i zablokuje cały ruch z adresu intruza niezależnie od tego, czy atakowana jest ta sama czy inna usługa.

## 5. Podsumowanie

Systemy ID przeszyły długą drogę od swoich początków w latach osiemdziesiątych po dzień obecny. Pierwsze systemy tego typu analizowały logi w celu wykrycia w nich zapisów

wskazujących na próbę włamania lub dowodzących, że próba powiodła się. Gwałtowny rozwój systemów ID zaowocował wieloma rozwiązaniami umożliwiającymi nie tylko bierne obserwowanie, ale również przeciwdziałanie wszelkim przejawom działań niezgodnych z polityką bezpieczeństwa. W zależności od umiejscowienia i zadań wykonywanych przez IDS systemy te dzielimy na dwie podstawowe kategorie: Host-Based IDS oraz Network-Based IDS. Rozwój systemów IT stawia nowe wyzwania przed systemami IDS. W chwili obecnej dwa spośród nich wydają się być znaczące — gwałtowny wzrost prędkości transmisji danych i związany z nim rozwój technologii przełączanych oraz szyfrowanie danych. Przełączniki separują ruch, przekazując pakiety wyłącznie do portu, do którego są dedykowane. Stanowi to utrudnienie, ponieważ sercem systemu IDS jest sniffer. Teoretycznie większość przełączników wyposażonych jest w funkcję umożliwiającą przekazywanie wszystkich pakietów do wybranego portu. Niestety, brzmi to dobrze tylko w teorii. W praktyce, przy dużym ruchu powoduje to znaczne spowolnienie pracy przełącznika. Innym problemem z wydajnością jest fakt, iż IDS musi dokonać analizy wszystkich pakietów i jednocześnie nie wpływać na ograniczenie transferu. W przypadku klasycznych systemów IDS pracujących w oparciu o analizę protokołów czy analizę sygnatur może to być problematyczne, bowiem ilość sygnatur, pod kątem których musi być przebadany pakiet, jest bardzo duża i stale rośnie, natomiast analiza protokołów jest sama w sobie procesem relatywnie mocno obciążającym procesor. Może to prowadzić nie tylko do ograniczenia pasma, ale co gorsza do „gubienia" pakietów przez IDS. Problem ten można rozwiązać stosując systemy IDS, tak aby zabezpieczały wybrany fragment sieci lub pojedynczy host, dzięki czemu ilość informacji, którą IDS musi przetworzyć, zostaje zredukowana. Kolejnym aspektem wskazującym na zwiększenie znaczenia rozwiązań HIDS jest szyfrowanie danych. Jedyną możliwością, aby dokonać analizy zawartości pakietu, jest odczytanie go po zdekodowaniu na miejscu przeznaczenia bądź też ochrona przed jego skutkami za pomocą systemu AD. Wraz z olbrzymim wzrostem ilości przetwarzanych danych lawinowo rośnie ilość fałszywych alarmów (*ang. false positives*), co zmusza administratora do poświęcania coraz większej ilości czasu na ich sprawdzanie. Z drugiej strony, niektóre typy ataków, takich jak np. slow port scanning, mogą pozostać w ogóle nie zauważone w takiej ilości danych. Wyznacza to kierunek, w którym powinny rozwijać się systemy wykrywania intruzów — analiza danych, badanie korelacji przy wykorzystaniu zaawansowanych systemów eksperckich, sieci neuronowych, sztucznej inteligencji itp. System będący połączeniem wielu niezależnych podsystemów chroniących newralgiczne punkty sieci, pojedyncze serwery i stacje robocze pracujące zarówno pod kontrolą systemów Unix'owych, jak i Windows oraz jedną stację zarządzającą zbierającą dane od poszczególnych elementów systemu, poddającą je skrupulatnej analizie w poszukiwaniu związków pomiędzy zdarzeniami wydaje się być nieuchronną przyszłością systemów ID.

## LITERATURA

1. Ko C., Ruschitzka M. and Levitt K.: "Execution Monitoring od Security-Critical Programs in Distributed Systems: A Specification–Based Approach", IEEE, 1997.
2. Denning D.: "An Intrusion-Detection Model", IEEE Symposium on Security and Privacy, 1986.
3. Kent Frederic K.: "Network Intrusion Detection Signatures, Part One – Part Three". SecurityFocus.com, 2001.
4. Tanase M.: "The Great IDS Debate: Signature Analysis Versus Protocol Analysis". SecurityFocus.com, 2003.
5. Tanase M.: "The Future of IDS". SecurityFocus.com, 2001.
6. Tanase M.: "One of These Things is not Like the Others: State of Anomaly Detection". SecurityFocus.com, 2002.
7. Desai N.: "Intrusion Prevention Systems: the Next Step in the Evolution of IDS", SecurityFocus.com, 2003.
8. Publiczna debata, "Signature Analysis versus Protocol Analysis", http://online.SecurityFocus.com/archive/96/260781/2002-03-03/2002-03-09/1, 2002-2003
9. Roesch M., Green C., "Snort User Manual, Snort Release: 1.9.1" 2002.

Recenzent: Dr inż. Adam Ziębiński

Wpłynęło do Redakcji 26 marca 2003 r.

**Abstract**

An Intrusion Detection System (IDS for short) monitors machines or networks for anomalies, attempted breaches and general misuse. Intrusion detection has become an essential component of computer network security in recent years. The beginning of IDS goes back to early 80`s and the first of them enable to analyze systems logs in order to detect abnormal user or system activity. This paper is attempted to classify wide spectrum of Intrusion Detection Systems beginning from classical, signature (2.1) or protocol (2.2) analysis, approach through anomaly detection systems (3) based on statistics (3.1) or specification (3.2) to advanced hybrid systems (4.4). Contemporary systems enable both intrusion detection and prevention (4). Depending on location of IDS we distinguish two main categories: host–based and network–based IDS. In conclusion author has concentrated on data encryption and new challenges created by switched environment. First problem might bring significance of Host-

based systems, second is connected with efficiency of IDS and large amount of false positives which force research into analysis systems (AI, fuzzy logic, neural network, data mining and mathematical based).

Adres

Rafał CICHOCKI: Akademia Morska w Gdyni, Katedra Podstaw Informatyki i Sieci Komputerowych, ul. Morska 83, 81-225 Gdynia, Polska, rafi@am.gdynia.pl