informatyka+

Algorytmika i programowanie

Bazy danych

Multimedia, grafika i technologie internetowe

Sieci komputerowe

Tendencje w rozwoju informatyki i jej zastosowań

Człowiek – najlepsza inwestycja

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Warszawska Wyższa Szkoła Informatyki

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

informatyka+

Wszechnica Poranna:

Tendencje w rozwoju informatyki i jej zastosowań

Wpływ systemów wykrywania włamań na bezpieczeństwo informatyczne instytucji

Krzysztof Różanowski

Człowiek – najlepsza inwestycja

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Warszawska Wyższa Szkoła Informatyki

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Wpływ systemów wykrywania włamań na bezpieczeństwo informatyczne instytucji

Rodzaj zajęć: Wszechnica Poranna
Tytuł: Wpływ systemów wykrywania włamań na bezpieczeństwo informatyczne instytucji
Autor: dr inż. Krzysztof Różanowski

Redaktor merytoryczny: prof. dr hab. Maciej M Sysło

Zeszyt dydaktyczny opracowany w ramach projektu edukacyjnego **Informatyka+** — ponadregionalny program rozwijania kompetencji uczniów szkół ponadgimnazjalnych w zakresie technologii informacyjno-komunikacyjnych (ICT).
www.informatykaplus.edu.pl
kontakt@informatykaplus.edu.pl

Wydawca: Warszawska Wyższa Szkoła Informatyki
ul. Lewartowskiego 17, 00-169 Warszawa
www.wwsi.edu.pl
rektorat@wwsi.edu.pl

Projekt graficzny: FRYCZ I WICHA

Warszawa 2009

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Warszawska Wyższa Szkoła Informatyki

UNIA EUROPEJSKA
EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Wpływ systemów wykrywania włamań na bezpieczeństwo informatyczne instytucji

Krzysztof Różanowski
WWSI Warszawa
krozan@wiml.waw.pl

Streszczenie

Wykład

Na wykładzie zostaną przedstawione podstawowe pojęcia i definicje związane z bezpieczeństwem informatycznym. Bezpieczeństwo jest elementem szerszego kontekstu, nazywanego wiarygodnością systemu informatycznego, dlatego na tle podstawowych atrybutów informacji związanych z systemem przedstawione i omówione zostaną dodatkowe atrybuty wiarygodności. Przedstawione zostanie znaczenie bezpieczeństwa w odniesieniu do roli systemów informatycznych, trudności związanych ze skonstruowaniem i eksploatacją systemu spełniającego wysokie wymagania w zakresie bezpieczeństwa oraz elementarnego konfliktu interesów występującego pomiędzy użytecznością systemu, a ryzykiem związanym z jego wykorzystaniem. Przedstawione zostaną również współczesne zagrożenia bezpieczeństwa i przykłady ataków na systemy informatyczne.

Omówione będą też następujące kwestie:

- Ataki i zagrożenia przypadkowe lub powstałe w efekcie celowego działania.
- Ataki i zagrożenia mogące wynikać z nieświadomości lub naiwności użytkownika lub motywowane chęcią zysku czy odwetu.
- Ataki i zagrożenia mogące pochodzić z zewnątrz systemu lub od jego środka.

Większość działań skierowanych w efekcie przeciwko bezpieczeństwu komputerowemu jest w świetle aktualnego prawa traktowana jako przestępstwa. Praktycznie wszystkie przypadki naruszające bezpieczeństwo wyczerpują znamiona przestępstw określonych w obowiązującym prawie. Dlatego w trakcie wykładu poruszone zostaną również aspekty prawne.

Przedstawione zostaną komponenty systemu informatycznego w kontekście bezpieczeństwa (stanowisko komputerowe i infrastruktura sieciowa, system operacyjny i usługi narzędziowe, aplikacje użytkowe) oraz wskazówki do projektowania systemów zabezpieczeń. Omówione zostaną problemy bezpieczeństwa sieci komputerowych w warstwie sieciowej i transportowej. Na tej podbudowie scharakteryzowane zostaną przywołane w temacie systemy wykrywania włamań IDS/IPS, których zadaniem jest identyfikacja i reagowanie na nieautoryzowaną działalność skierowaną przeciwko chronionym zasobom sieciowym. Charakterystyka dotyczyć będzie trzech głównych rodzajów systemów IDS (HIDS, NIDS, NNIDS).

Warsztaty

Laboratorium dedykowane jest praktycznej realizacji i testowaniu skuteczność działań systemów wykrywania włamań na przykładzie programu SNORT, który jest programem typu *open-source*, docelowo stworzonym dla systemu UNIX, dostępnym także dla systemu Windows. Służyć może do analizy pakietów, wyszukiwania i dopasowywania podejrzanych treści, a także wykrywania ataków i anomalii np. ataków na serwery WWW, ukrytego skanowania portów czy prób identyfikacji.

Dla słuchaczy przygotowane zostanie środowisko pracy w postaci maszyny wirtualnej z konfiguracją: system SNORT, aplikacja zarządzania danymi BASE dla danych z systemu SNORT, serwer WWW+ PHP, MySQL, ADOdb, zestaw exploitów oraz skaner sieciowy (Nessus).

Celem ćwiczeń będzie właściwa konfiguracja i utworzenie własnych reguł wykrywania włamań i ich kwalifikacja do określonej klasy ataków w zależności od sygnatur wykorzystywanych exploitów. Ataki te będą typu SQL Injection.

System SNORT zostanie przetestowany w trzech trybach pracy: Sniffer – przechwytywanie wszystkich pakietów i wyświetlanie na ekranie, Packet Logger – zapis wszystkich przechwyconych pakietów do pliku, Network Intrusion Detection Mode – sieciowy system wykrywania włamań.

Spis treści

WPROWADZENIE

Celem warsztatów jest zapoznanie uczestników z działaniem systemów wykrywania włamań do sieci komputerowej. Dedykowane są praktycznej realizacji i testowaniu skuteczność działań takich systemów na przykładzie systemu SNORT. Jest to program typu *open-source*, utworzony dla systemu UNIX, dostępny także dla systemu Windows. Służy m.in. do analizy pakietów, wyszukiwania i dopasowywania podejrzanych treści, a także do wykrywania ataków i anomalii, np. ataków na serwery WWW, ukrytego skanowania portów czy prób identyfikacji. Słuchacze wykorzystają przygotowane środowisko pracy w postaci maszyny wirtualnej w konfiguracji: system SNORT, aplikacja zarządzania danymi BASE dla danych z systemu SNORT, serwer WWW + PHP, MySQL, ADOdb, zestaw exploitów oraz skaner sieciowy (Nessus).

Praktycznym zadaniem do wykonania będzie właściwa konfiguracja i utworzenie własnych reguł wykrywania włamań i ich kwalifikacja do określonej klasy ataków w zależności od sygnatur wykorzystywanych exploitów. Ataki te będą typu SQL Injection. System SNORT zostanie przetestowany w trzech trybach pracy:

- Sniffer (przechwytywanie wszystkich pakietw i wyświetlanie na ekranie),
- Packet Logger (zapis wszystkich przechwyconych pakietów do pliku),
- Network Intrusion Detection Mode (sieciowy system wykrywania włamań).

1 KRÓTKA CHARAKTERYSTYKA SYSTEMÓW WYKRYWANIA WŁAMAŃ

Systemy wykrywania włamań są komputerowymi odpowiednikami alarmu antywłamaniowego. Monitorują punkty dostępu do systemu oraz niepożądane działania w systemie. Według najprostszej definicji, system IDS (ang. *Intrusion Detection System*) jest specjalistycznym narzędziem, służącym do interpretacji zasobów dzienników zdarzeń routerów, firewalli i innych urządzeń sieciowych lub systemowych według ustalonych reguł. System IDS przechowuje bazę znanych sygnatur ataków i porównuje ją z wzorcami zapisanymi w dziennikach. Jeśli pasują do sygnatur ataków, to oznacza, że włamania ma lub miało miejsce. System IDS może uruchomić alarmy lub alerty, a także podjąć akcję automatycznie. Większość systemów IDS ma możliwość definiowania takich czynności. Dość istotnym jest podjęcie czynności, która umożliwi identyfikację włamywaczy i zebranie dowodów wrogiej działalności. Wykrywanie włamań oznacza zatem detekcję nieautoryzowanego wykorzystania systemu informatycznego lub sieci. Systemy typu IDS są zaprojektowane i wykorzystywane do wykrywania włamań, a następnie blokowania ataków. Podobnie jak firewall, mogą być pakietami oprogramowania lub połączeniem odpowiedniego sprzętu i oprogramowania. Bardzo często oprogramowanie IDS działa na tych samych serwerach co firewall i serwery Proxy. System IDS, który nie działa na tym samym urządzeniu co firewall lub inne usługi jest w stanie kontrolować dokładniej. Mimo, iż urządzenia z systemem IDS zazwyczaj działają na peryferiach sieci, są w stanie radzić sobie zarówno z zewnętrznymi jak i wewnętrznymi atakami na sieć. Systemy IDS można klasyfikować według kryterium działania, typu komunikacji, transakcji lub systemów, które mają monitorować. Można je podzielić na systemy dla hostów, sieciowe systemy IDS i tzw. systemy rozproszone. Systemy IDS, które monitorują sieć nazywane są **sieciowymi systemami IDS** (ang. *network-based IDS*), a działające na konkretnych hostach i monitorujące ich systemy plików są nazywane **systemami IDS dla hostów** (ang. *host-based IDS*). Systemy te można dodatkowo podzielić na trzy kategorie:

1. **Tradycyjne** – z programem agenta zainstalowanym na każdej chronionej maszynie. Agent nadzoruje logi systemowe, dziennik zdarzeń, kluczowe pliki systemowe oraz inne zasoby, które mogą podlegać weryfikacji. Podejrzane działania są wykrywane po ich zarejestrowaniu przez system. Ostrzeżenia o nadużyciach i zauważonej nieautoryzowanej działalności są wysyłane poprzez sieć do centralnej konsoli.

2. Programy **badające integralność plików** – sprawdzają status kluczowych plików systemowych oraz rejestru. Zapamiętują stan wybranych plików (najczęściej poprzez stworzenie bazy z sumami kontrolnymi) i w określonym czasie dokonują porównania ze stanem bieżącym. Taki sposób działania umożliwia wykrycie podmiany plików (np. na konia trojańskiego) oraz zmiany w konfiguracji. Przykładem takiego programu jest Tripwire.

3. Systemy **zapobiegania włamaniom** (IPS – *Intrusion Protect System*) – przyjmują aktywną postawę w stosunku do zagrożeń – integrują się z systemem operacyjnym, przechwytując wywołania systemowe jądra lub interfejsów programowych API. W momencie wykrycia podejrzanych działań, programy IPS są w stanie zablokować wywołanie danej funkcji i udaremnić atak.

Systemy działające jako zdalne sensory i składające raporty do centrum zarządzania są nazywane **rozproszonymi systemami IDS** (ang. *Distributed IDS*).

Systemy IDS można również sklasyfikować ze względu na sposób analizowania zdarzeń. Podstawowa metoda, tzw. **analiza sygnatur** (dopasowywanie wzorców: zestawów bajtów, wyrażeń regularnych) przypomina programy antywirusowe, które wykorzystują sygnatury wirusów do blokowania zainfekowanych zasobów. Systemy IDS wykorzystują bazy danych wzorców komunikacji lub aktywności odpowiadające różnym typom znanych ataków (sygnatury ataków).

Nieco bardziej zaawansowaną techniką wykorzystywaną w systemach IDS jest **wykrywanie anomalii**. Metoda wykorzystuje predefiniowane pojęcia opisujące „normalne" i „nienormalne" zachowania systemu. Umożliwia wykrywanie anomalii odbiegających od zwykłego toku działania systemu. Czasami wykorzystywane są również profile użytkownika, które można przygotować za pomocą metod statystycznych, np. sieci neuronowych. Bada się również częstości zdarzeń i przekroczenia pewnych limitów w określonej jednostce czasu.

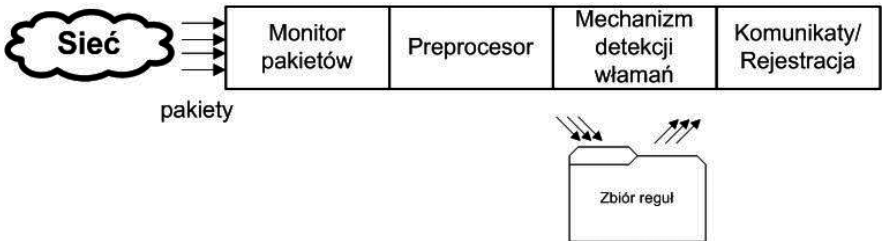
2 SYSTEM WYKRYWANIA WŁAMAŃ SNORT

SNORT jest aplikacją sieciowego systemu wykrywania i zapobiegania atakom, dostępną na licencji wolnego oprogramowania. Jest bardzo dobrym rozwiązaniem systemu typu NIDS (ang. *Network Intrusion Detection System*) o wielu użytecznych opcjach. Poza realizacją podstawowego zadania wykrywania włamań, umożliwia monitorowanie ruchu sieciowego, rejestrowanie pakietów oraz pracę w trybie *in-line*. Przesyła on do użytkownika komunikaty alarmowe, dzięki którym administrator jest na bieżąco informowany o potencjalnych i rzeczywistych zagrożeniach w systemie. Aplikacja ta jest popularna ponieważ nie obciąża systemu, jest niekiedy nazywana **lekkim systemem NIDS**. Dodatkowo zawiera funkcje, które kiedyś dostępne były jedynie w rozwiązaniach komercyjnych.

Działanie systemu SNORT bazuje na mechanizmie weryfikacji sygnatur i przechwytywaniu pakietów sieciowych za pomocą zestawu reguł sprawdzających ich zawartość. Aplikacja składa się z kilku ważnych komponentów. Są to mechanizmy przetwarzania wstępnego, dodatki wyzwalania alarmów oraz system alarmowy. Cały program składa się z czterech głównych modułów:

- monitora,
- preprocesora,
- mechanizmu detekcji włamań,
- modułu wyjściowego.

Ogólny schemat blokowy aplikacji jest przedstawiony na rys. 1.

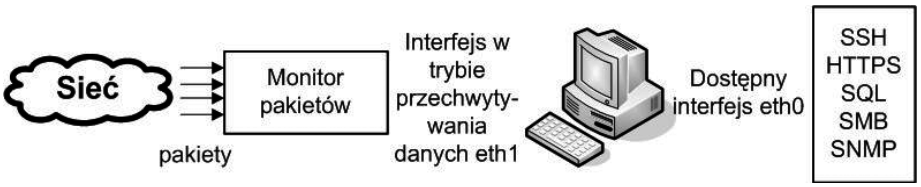


Rysunek 1.
Schemat blokowy systemu SNORT

2.1 MONITOR PAKIETÓW

W podstawowej konfiguracji SNORT jest zwykłym skanerem sieciowym. Przechwytuje pakiety za pomocą biblioteki *pcap*. Rozpoznaje różne protokoły, np.: Ethernet, 802.11, Token Ring, IP, TCP, UDP i ICMP. Pakiety po zdekodowaniu wędrują do preprocesorów (warstwa transportowa), gdzie są testowane i w razie konieczności obrabiane na potrzeby silnika detekcji (warstwa sesji). Tam dokonywana jest analiza pakietów pod kątem zbioru zadanych reguł. Następnie po wykryciu próby ataku bądź anomalii sieciowych, system przekazuje odpowiednie dane do modułu wyjściowego, który decyduje o zapisaniu wyniku wykrycia w logach lub wszczęciu alarmu. Alarm wywoływany jest tylko przez reguły natomiast logowanie odbywa się poprzez preprocesory.

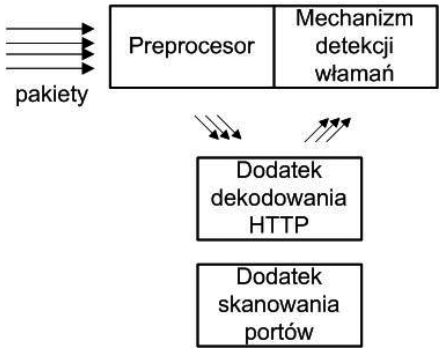
Monitor pakietów jest elementem umożliwiającym podsłuch danych przesyłanych przez sieć. Można go wykorzystywać do m.in.: wykrywania usterek w działaniu sieci, przygotowywania statystyk ruchu oraz przechwytywania niezaszyfrowanych haseł (rys. 2).



Rysunek 2.
Przechwytywanie pakietów w programie SNORT

2.2 BLOK PREPROCESORA

Moduł preprocesora pobiera nieprzetworzone pakiety i sprawdza ich zawartość zgodnie z definicją znajdującą się w dodatkach (rys. 3). Dodatki te weryfikują dane pakietów sprawdzając występowanie informacji o określonym sposobie ich przekazywania. Jeśli test wykaże, że pakiet należy do określonej grupy, to jest on przekazywany do mechanizmu detekcji włamań.



Rysunek 3.
Preprocesor programu SNORT

Preprocesory umożliwiają użytkownikom i programistom w prosty sposób rozbudowę funkcjonalności całego systemu poprzez pisanie dodatkowych wtyczek (ang. *plugins*).

Preprocesory analizują pakiety przed wykorzystaniem ich przez silnik detekcji. W ten sposób zwiększają możliwości całego procesu wykrywania ataków sieciowych, wzbogacając go o zdolność składania (reasemblacji) pakietów, wykonywania specyficznych dla poszczególnych protokołów operacji (np. konwersji na ASCII znaków z URI zakodowanych szesnastkowo, usuwania ciągów binarnych z sesji FTP czy Telnet, normalizacji żądań RPC), jak i wykrywania niezgodności z tymi protokołami. Poniżej zamieszczamy krótką charakterystykę preprocesorów, mogących być wykorzystanych w systemie SNORT.

Frag2 – defragmentuje i normalizuje dane przychodzące w postaci fragmentów, co utrudnia ukrywanie ataków prowadzonych za pomocą nieprawidłowo sfragmentowanych pakietów IP. Umożliwia wychwytywanie znanych ataków wykorzystujących zniekształcenia fragmentów pakietów np. teardrop, fragroute.

Stream4 – rozwija model detekcji oparty na testowaniu pojedynczych pakietów umożliwiając śledzenie sesji (stanu połączenia) TCP i składanie (reasemblacji) strumieni TCP, co nie byłoby możliwe w mechanizmie opartym na wyszukiwaniu wzorców.

Flow i *Flow-Portscan* – zawierają mechanizm śledzenia połączeń, zapisując całość do tablicy stanów w celu dalszego przetwarzania. Na tej podstawie flow-portscan wykrywa próby skanowania w bardziej wyrafinowany sposób niż preprocesory stream4 i portscan. Celem wykrywania, są skanowania wielu hostów i wielu portów przeprowadzane przez danego hosta.

HTTP Inspekt – jest ogólnym dekoderem protokołu HTTP na poziomie warstwy aplikacyjnej. Za pomocą ustalonego bufora wynajduje odpowiednią składnię HTTP i ją normalizuje. Działa w obydwu trybach jednocześnie: *client requests* (zapytania klienta) i *server responses* (odpowiedzi serwera). Głównymi jego zadaniami jest przetwarzanie adresów URL, konwertując na ASCII znaki zakodowane w postaci szesnastkowej.

Portscan Detektor – wykrywa próby skanowania portów, polegające na przekroczeniu pewnej progowej liczby prób połączeń z różnymi portami w określonym przedziale czasu. Ze względu na brak możliwości uniknięcia fałszywych alarmów w typowych przypadkach (np. obciążony serwer DNS), istnieje możliwość wyłączenia alarmów wzbudzanych przez określone adresy IP używając dodatkowego procesora *portscan-ignorehosts*. Umożliwia także zapisanie wyników w oddzielnym pliku dziennika.

Telnet Decode – usuwa z sesji TELNET i FTP binarne ciągi mogące utrudnić wyszukiwanie z udziałem sygnatur ataków.

RPC Decode – normalizuje wywołania przesyłane protokołem RPC, utrudniając ukrywanie podejrzanych pakietów za pomocą mniejszych operacji.

Back Orifice detektor – wyszukuje w pakietach UDP próby połączeń konia trojańskiego Back Orifice i próbuje złamać zabezpieczające je słabe kodowanie.

Arpspoof – wykrywa podejrzane pakiety ARP, mogące sygnalizować próby *ARP spoofingu*.

Performance Monitor – udostępnia wszelkiego rodzaju statystyki liczbowe, odnośnie liczby przeanalizowanych pakietów, zużycia procesora itp. Całość wyświetlana jest na ekranie konsoli lub zapisywana do pliku, według ustalonych wcześniej wartości [14].

2.3 MECHANIZM DETEKCJI WŁAMAŃ

Mechanizm detekcji włamań jest najważniejszym elementem systemu SNORT. Pobiera on dane dostarczane przez preprocesor i poddaje testom określonym przez zbiór reguł. Jeśli dane zawarte w pakiecie spełniają kryteria którejkolwiek z reguł, to są przekazywane do modułu generowania alarmu. Zatem główny mechanizm systemu detekcji zagrożeń polega na ocenie podobieństwa przetworzonych pakietów i ich zrekonstruowanych strumieni z bazą sygnatur. System detekcji porównuje cechy pakietu ze zbiorem reguł. Po dopasowaniu, zostaje podjęta odpowiednia akcja. Do porównywalnych cech należą atrybuty główne – adresy, porty źródłowe i docelowe oraz opcje pomocnicze, takie jak: flagi TCP identyfikujące żądania związane z WWW, różne typy pakietów ICMP, opcje IP, czy wreszcie sama treść pakietu. W głównej części reguł możliwe jest śledzenie protokołów IP, ICMP, TCP i UDP. Reguły identyfikowania ataku umożliwiają podjęcie pięciu rodzajów akcji:

- *pass* – przepuszczenie pakietu,
- *log* – zapisanie informacji do dziennika,
- *alert* – ogłoszenie alarmu,
- *activate* – alarmowanie i podjęcie do działania innej dynamicznej reguły,
- *dynamic* – pozostanie w spoczynku do czasu aktywowania przez regułę activate, po czym działanie jako reguła log.

Dodatkowo w trybie *in-line* są dostępne jest akcje *drop*, nakazująca filtrowi odrzucenie pakietu oraz *sdrop* bez procesu logowania informacji o odrzuceniu pakietu.

Reguły systemu SNORT zazwyczaj składają się z dwóch głównych sekcji – nagłówka i ciała (treści). Nagłówek określa m.in., jaką akcję należy podjąć po dopasowaniu reguły, informacje o wykorzystanym protokole, adresy bądź porty źródłowe i docelowe. W ciele reguły można rozwinąć informacje zawarte w nagłówku, tu także podaje się treść wzbudzanych alarmów i różnego rodzaju informacje dodatkowe.

Najprostsze reguły obejmują wskazanie akcji, protokołu, kierunku, adresów i portów będących przedmiotem obserwacji, jak np. poniższa reguła, stanowiąca reakcję na próbę skorzystania z usługi pop3 (port 110):

```
log tcp any any -> 192.168.1.0/24 110
```

W podanych powyżej regułach wykorzystany był jednokierunkowy operator –>. Język sygnatur umożliwia zadeklarowanie reguły, która dopasuje pakiety poruszające się w obu stronach za pomocą operatora dwukierunkowego <>, np.:

```
alert tcp any any <> 192.168.1.0/24 110
```

Do zasadniczej części reguły można dodać ograniczone okrągłymi nawiasami pole opcjonalne (tzw. ciało), zawierające definicję bardziej złożonych i wyrafinowanych działań związanych z przejęciem danego pakietu. Użytkownik może także sformułować własny komunikat, np.:

```
log tcp any any -> 192.168.1.0/24 110 (msg: „Proba
polaczenia z pop3”;)
```

Podjęte działania nie muszą być ograniczone do pojedynczej czynności. Średnik separuje deklaracje poszczególnych działań, jak w poniższym przykładzie, w którym opcją `content` testowana jest treść przesyłanego strumienia TCP, a w razie odnotowania podejrzanego ciągu znaków generowany jest odpowiedni komunikat:

```
alert tcp any any -> 192.168.1.0/24 80 (content: „/cgi-bin
/phf”; msg: „PHF probe!”;)
```

Opcji `content` można użyć nawet kilka razy w jednej regule, dzięki temu można wyszukiwać wiele różnych ciągów znaków w obrębie przesyłanych treści.

Bardzo silną konstrukcją w regułach systemu SNORT jest możliwość aktywowania kolejnych reguł po pierwszym dopasowaniu. Konstrukcja ta nosi nazwę *activate/dynamic rules* i ma następując a postać:

```
activate tcp any any -> $HOME _ NET 143 (flags: PA; content:
„|E8C0FFFFFF|bin|;activates: 1; msg: „IMAP buffer overflow!”;)
dynamic tcp any any -> $HOME _ NET 143 (activated _ by: 1;
 count: 50;)
```

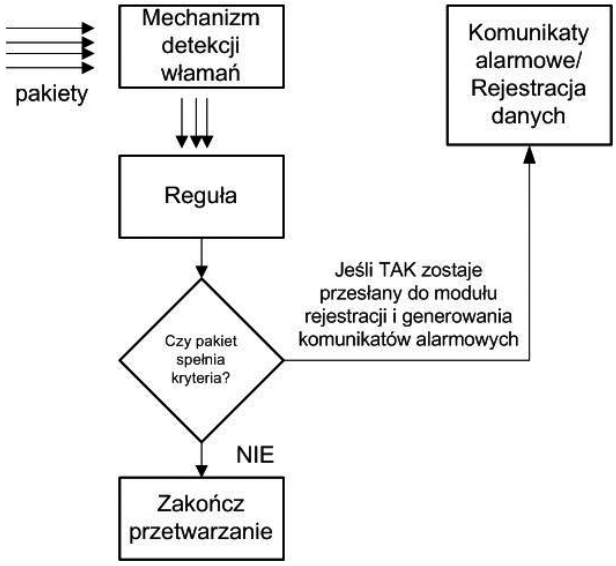
Opcje `activates` i `activated _ by` wiążą reguły `activate` i `dynamic`. W powyższym przykładzie wykrycie ataku typu `buffer overflow` na serwer IMAP powoduje uruchomienie kolejnej, dynamicznej reguły, która zbiera treść następnych 50 pakietów (opcja `count`) do późniejszej analizy. Druga opcja reguły dynamicznej jest obligatoryjna – reguła zawierająca wyłącznie opcję dowiązania do innej, macierzystej konstrukcji jest bezużyteczna.

Wybrane słowa kluczowe reguł systemu SNORT:

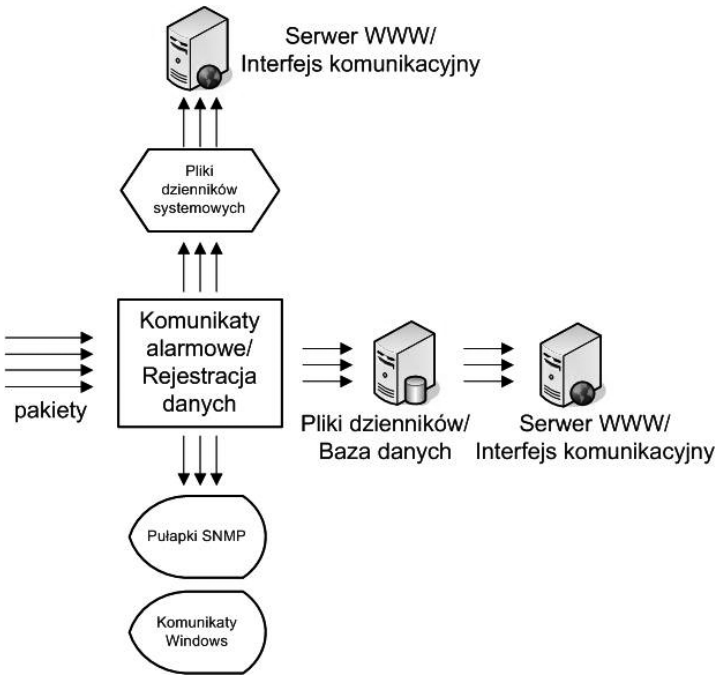
- `msg` – zapisanie wiadomości w logu,
- `reference` – odnośnik do opisu ataku,
- `sid` – numer reguły,
- `content` – sprawdzenie czy pakiet zawiera dany tekst,
- `rawbyte` – sprawdzenie czy pakiet zawiera ciąg liczb,
- `depth` – określenie liczby przeszukiwanych pakietów,
- `offset` – określenie pierwszego przeszukiwanego pakietu,
- `ttl` – filtrowanie względem parametru TTL,
- `tos` – filtrowanie względem parametru *tos*,
- `flags` – filtrowanie względem flag (ustawione, ignorowane), np.: (`flags`: FP,U12 oznacza, że flagi PSH, FIN mają być ustawione, SYN,RST nie ustawione, zaś wartości bitów dodatkowych 1 i 2 oraz flagi U nie mają znaczenia)
- `eplace` – dokonuje podmiany odnalezionego tekstu (np. poleceniem `content`). Zamienia jedynie teksty o tej samej długości. Wymagana jest praca trybie *in-line*.

Na rys. 4 przedstawiono w ogólny sposób działania mechanizmu detekcji włamań, który wykorzystuje omówione wcześniej reguły. Po przetworzeniu danych przez ten mechanizm muszą one zostać przekazane do kolejnego bloku programu. Jeśli zawartość pakietu spełnia kryteria reguł, wyzwalany jest alarm. Komunikaty alarmowe mogą być przekazywane do plików dzienników, albo przez sieć do gniazd Unix, systemu obsługi komunikatów Windows (SMB) lub jako pułapki SNMP. Informacje mogą być zapisywane w bazach danych SQL, np.: MySQL i PostreSQL.

Dodatkowo program SNORT jest wspierany przez wiele różnych narzędzi, które umożliwiają wyświetlanie zawartości plików dziennika na stronach serwera WWW. Budowa modułu generowania komunikatów alarmowych jest przedstawiona na rys. 5. Na przykład aplikacja *snortSnarf* służy do analizy komunikatów i wygenerowanie raportu w postaci kodu HTML. Program *snortplot.php* umożliwia przedstawienie danych o atakach

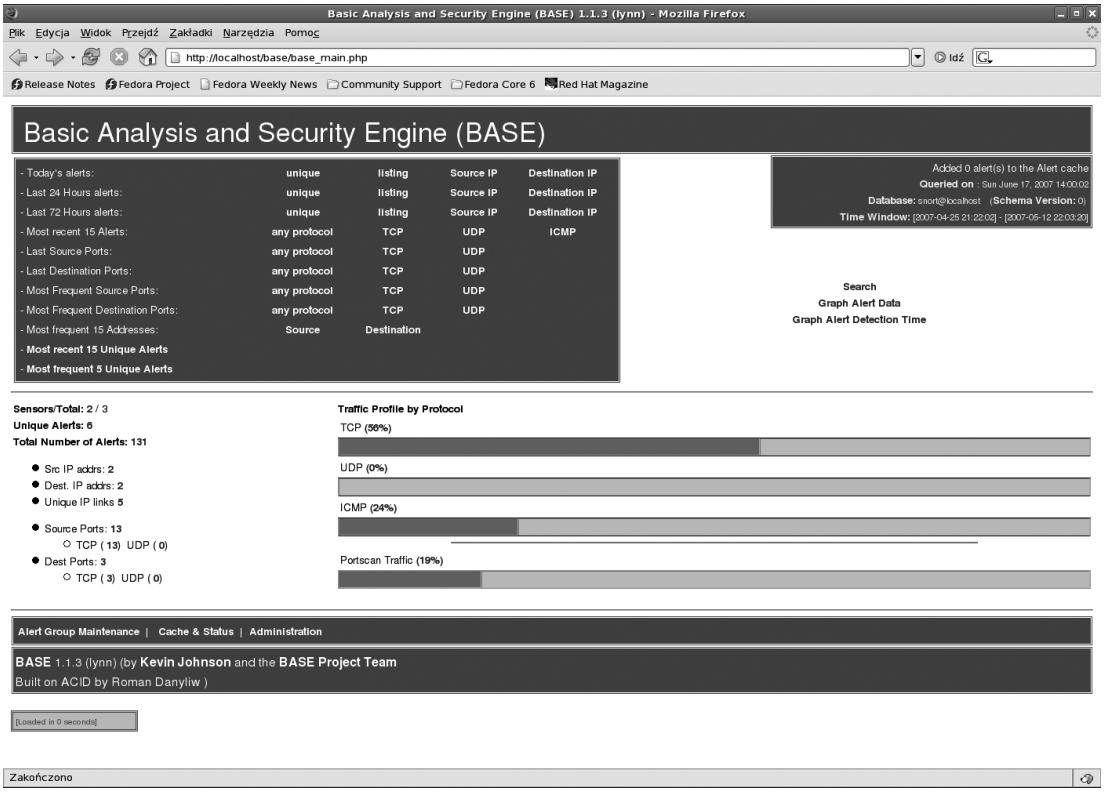


Rysunek 4.
Mechanizm detekcji włamań programu SNORT



Rysunek 5.
Budowa modułu generowania komunikatów alarmowych aplikacji SNORT

w postaci graficznej. *Swatch* – analizuje zawartość dzienników w czasie rzeczywistym i przekazuje komunikaty za pomocą poczty elektronicznej. Program *ACID* zapewnia analizę komunikatów systemu SNORT, lecz wymaga zainstalowania języka PHP, serwera Apacze i dodatku SNORT, odpowiedzialnego za komunikację z bazami danych. Aplikacja *Loghog* przekazuje komunikaty za pomocą postów lub blokuje ruch poprzez integrację z *iptables*. *SNORT Report* jest dodatkowym modułem, który na bieżąco generuje raporty o wykrywaniu włamań. Do aktualizacji sygnatur może posłużyć perlowy skrypt *Oinkmaster*. Ma on duże możliwości, które w prosty sposób umożliwiają zarządzanie regułami systemu SNORT. Wymagane pakiety do uruchomienia to: perl, perl-base, perl-modules i vixie-cron albo hc-cron.



Rysunek 6.
Okno główne aplikacji BASE z przykładowymi alertami wygenerowanymi w środowisku testowym

W pracy w środowisku testowym nad analizą danych, przetworzonych przez system SNORT jest wykorzystana aplikacja BASE (ang. *Basic Analysis and Security Engine*), będąca wynikiem prac i wykorzystaniem doświadczeń z projektu ACID (ang. *Analysis Console for Intrusion Databases*). Główne okno aplikacji jest przedstawione na rys. 6. Aplikacja ta dostarcza webowy front-end, za pomocą którego możliwe jest zadawanie pytań oraz analiza alertów pochodzących z systemu SNORT. Wykorzystuje ona system autentykacji na podstawie zdefiniowanych ról. Administrator zatem może zdefiniować zakres informacji dla poszczególnych użytkowników, do której będą mieli dostęp. Ma przyjemny interfejs, zwłaszcza dla tych, którzy niechętnie zaglądają do logów zapisanych w postaci tekstowej. Aplikacja ta jest tworzona w ramach otwartego projektu. Więcej na temat tego projektu można zaleźć na jego głównej stronie http://base.secureideas.net/.

2.4 KONFIGURACJA I URUCHOMIENIE SYSTEMU SNORT

Do działania jako sieciowy system wykrywania włamań, SNORT potrzebuje sprecyzowania zasad funkcjonowania całości w głównym pliku konfiguracyjnym `snort.conf`. W starszych wersjach systemu, wszystkie opcje, łącznie z regułami ataków, znajdowały się w jednym pliku. Ciągła rozbudowa tego programu, rosnąca liczba sygnatur i ogólna funkcjonalność, wymusiły rozdzielenie niektórych części konfiguracyjnych, w tym reguł ataków. Przejrzystość i spójność pliku `snort.conf` została przywrócona przy użyciu polecenia `include`, którym dołącza się odpowiednie zestawy sygnatur i inne części konfiguracyjne, np.:

```
include: ścieżka _ do _ pliku/nazwa
```

Bazy reguł charakteryzują się nazwą pliku z końcówką `.rules`, pierwszy człon nazwy zawiera rodzaj usługi lub typ ataku, którego dotyczy dany zestaw. Pozostałymi plikami konfiguracyjnymi są:

- `classification.config` – zawierający klasyfikatory rodzajów ataków z nadanym priorytetem zagrożenia, tak jak poniżej:
- `reference.config` – zawierający skróty adresów do stron organizacji z bazą opisów ataków:
- `threshold.conf` – metody łagodzenia licznych, fałszywych alarmów,
- `unicode.map` – zestaw kodowanych znaków unicode, na potrzeby preprocesora `http _ inspect`.

Główny plik konfiguracyjny można podzielić na cztery sekcje.

Pierwsza jest odpowiedzialna za ustalanie zmiennych `var`, wykorzystywanych w składni reguł ataków. Pliki konfiguracyjne znajdują się bezpośrednio w katalogu `/etc/snort`, a reguły – w `/etc/snort/rules`. Dla dwóch monitorowanych podsieci o adresach 192.168.1.0/24 i 192.168.2.0/24 plik konfiguracyjny może mieć postać:

```
# Adres sieci lokalnej
var HOME _ NET [192.168.1.0/24,192.168.2.0/24]
# Adres sieci zewnetrznej
var EXTERNAL _ NET !$HOME _ NET
# Lista adresow serwerow znajdujacych sie w strefie chronionej
var DNS _ SERVERS $HOME _ NET
var SMTP _ SERVERS $HOME _ NET
var HTTP _ SERVERS $HOME _ NET
var SQL _ SERVERS $HOME _ NET
var TELNET _ SERVERS $HOME _ NET
var SNMP _ SERVERS $HOME _ NET
# Lista portow
var HTTP _ PORTS 80
var SHELLCODE _ PORTS !80
var ORACLE _ PORTS 1521
# Lista serwerow czat, komunikatorow
var AIM _ SERVERS [64.12.24.0/24,64.12.25.0/24,64.12.26.14/24,64.12.28.0/24,]
# Scieżka do katalogu z regulami atakow
var RULE _ PATH /etc/snort/rules
```

W tej samej sekcji znajduje się zestaw parametrów uruchomieniowych, zaczynających się od wyrażenia `config` (pełna ich lista znajduję się w dokumentacji), na przykład:

```
# wybor interfejsu do nasłuchu (jeden demon SNORTa moze
# obslugiwac tylko jeden interfejs sieciowy)
config interface: eth0
```

Druga sekcja głównego pliku konfiguracyjnego zawiera ustawienia preprocesorów, które zostały opisane wcześniej.

Trzecia sekcja `snort.conf`, zawiera metody konfiguracji modułów wyjściowych, czyli różnych sposobów logowania wyników i tzw. akcji reguł. Jeżeli informacja będzie logowana do bazy MySQL, to należy dopisać następujące wyrażenie:

```
output database: alert, mysql, user=login password=haslo \
dbname=snort _ log host=127.0.0.1
```

Czwarta i ostatnia sekcja głównego pliku konfiguracyjnego zawiera odniesienia do zestawów reguł i wcześniej już opisanych plików: `classification.config, reference.config, threshold.conf`:

```
include classification.config
include reference.config

include $RULE _ PATH/bad-traffic.rules
include $RULE _ PATH/exploit.rules
include $RULE _ PATH/telnet.rules
include $RULE _ PATH/rpc.rules
include $RULE _ PATH/dos.rules
(...)
```

```
# dodatkowy zestaw regul Bleeding SNORT -
# http://www.bleedingsnort.com
include $RULE _ PATH/bleeding.rules
include threshold.conf
```

Konfiguracja systemu SNORT wymaga konfiguracji preprocesorów, ale przede wszystkim określenie zbiorów reguł, które mają być brane pod uwagę, czyli jakie pliki z regułami mają być dołączane za pomocą polecenia `include`. Dobranie odpowiednich dla danego środowiska reguł jest kluczowe dla działania całego systemu, duża ilość fałszywych alarmów nie tylko zużywa zasoby, ale może również bardzo skutecznie ukryć prawdziwy atak na nasz system. Obecna baza sygnatur liczy przeszło 2500 reguł i jest praktycznie każdego dnia wzbogacana o nowe opisy ataków [15]. Wykrywane rodzaje ataków:

- `attack-responses` – odpowiedzi usług na próbę ataku;
- `backdoor` – działalność tzw. tylnych drzwi, trojanów, rootkitów;
- `bad-traffic` – nieprawidłowy ruch, np. na port 0;
- `chat` – aktywność różnego rodzaju komunikatorów;
- `ddod, dos` – zmasowane ataki Distributed Denial of Service (rozproszony atak typu blokada usługi);
- `deleted` – reguły przestarzałe, wykasowane;
- `dns` – ataki na usługę Domain Name System;
- `experimental` – zestaw eksperymentalnych reguł;
- `exploit` – programy mające na celu wykorzystywanie błędów w oprogramowaniu;
- `info, icmp` – komunikaty ICMP z różnych programów, testujących ruch;
- `imap, pop2, pop3, smtp` – ataki na systemy pocztowe;
- `info` – próby logowania na usługi Telnet, Ftp;
- `local` – zestaw własnych reguł;
- `misc` – różne, dotyczące usług CVS, MS Terminal, BOOT, UPnP itd.;
- `multimedia` – strumienie audio, wideo;
- `mysql, sql, oracle` – ataki na znane serwer baz danych;
- `netbios` – anomalia związane z protokołem Netbios/SMB;
- `nntp` – ataki na serwer grup dyskusyjnych;
- `other-ids` – działalność innych systemów IDS;
- `p2p` – aktywność programów Peer to Peer;
- `policy` – próby ataków na usługi policy ftp itp.
- `porn` – aktywność stron pornograficznych;
- `rpc` – ataki na usługi Remote Procedure Call;
- `finger, rservices, telnet` – ataki na dość słabo zabezpieczone usługi uniksowe: finger, rlogin, rsh, rexec, telnet;
- `scan` – różnego rodzaju techniki skanowania portów;
- `shellcode` – wykorzystanie nieprawidłowego kodu do prób przepełnienia bufora;
- `snmp` – ataki na usługi SNMP;
- `tftp, ftp` – zdarzenia związane z przesyłaniem plików poprzez serwer ftp;
- `virus` – transfer poczty z podejrzanym załącznikiem;
- `web-attacks, web-misc, web-client, web-cgi, web-php, web-coldfusion, web-frontpage, web-iis` – ataki na różnego typu serwery WWW, przeważnie z wykorzystaniem błędów w skryptach cgi, php;
- `x11` – aktywności sesji serwera XFree86.

Jak wspomniano wcześniej, program SNORT w podstawowej konfiguracji jest aplikacją przechwytywania pakietów. Uruchomienie go w tym trybie wymaga użycia następujących opcji:

```
# snort –d –e –v
```

gdzie:
`–v` – przełączenie programu SNORT w tryb monitorowania sieci (analiza jedynie nagłówków TCP),
`–d` – uwzględnienie nagłówków warstwy sieciowej (IP i ICMP),
`–e` – uwzględnienie nagłówków warstwy łącza danych.

Format wyniku jest zbliżony do raportu, jaki generuje TCPDump. Generalnie jest zbliżony do formatu większości monitorów sieci:
{data}-{czas} {źródłowy-adres-MAC} -› { docelowy-adres-MAC} {typ}
{rozmiar} {źródłowy-adres-IP:port} -› {docelowy-adres-IP:port} {protokół} {TTL}
{TOS} {ID} {rozmiar-IP} {rozmiar-datagramu} {rozmiar-pola-danych} {wartości-hex.} {wartości-ASCII}

Aby włączyć opcję rejestrowania pakietów należy wykonać polecenie:

```
# snort –dev {katalog-rejestracji} –h {adres-podsieci-w-notacji-CIDR}
```

Zapisywanie pakietów w formacie binarnym znacznie przyspiesza działanie aplikacji. Wówczas potrzebne są dwie opcje:
`–L` – rejestracja
`–b` – ustanowienie trybu binarnego

```
# snort –b –L {plik dziennika}
```

Bardziej zaawansowane operacje rejestrowania pakietów wymagają zastosowania filtru BPF (ang. *Berkeley Packet Filter*). Moduł ten umożliwia filtrowanie pakietów na poziomie jądra. Rozwiązanie to znacznie usprawnia działanie aplikacji monitorujących i rejestrujących pakiety sieciowe, gdyż odrzucanie niepotrzebnych pakietów odbywa się na poziomie jądra (*kernel*) systemu operacyjnego.

Aby uruchomić system SNORT jako IDS, wystarczy uzupełnić funkcję rejestracji o nazwę pliku konfiguracyjnego:

```
# snort –dev –l {katalog-rejestarcji} –h {adres-podsieci-w-notacji-CIDR}\–c {plik–konfiguracyjny}
```

Alarmy będą wyzwalane zgodnie z regułami zapisanymi w podanym pliku konfiguracyjnym.

2.5 KONCEPCJE ROZWOJOWE SYSTEMU SNORT

Obecnie dużą popularność zyskuje nowy rodzaj systemów IDS, tzw. in-line IDS [http://www.snort.org/docs/tap/]. Rozwiązania te są co do zasady działania podobne do systemów typu Firewall. Ruch wchodzi do urządzenia IDS przez interfejs sieciowy i wewnątrz jest poddawany analizie, a następnie wychodzi z urządzenia drugim interfejsem. Dzięki temu kontrola ruchu sieciowego jest łatwiejsza, pojawia się mniej jak w zwykłych IDS fałszywych alarmów i co najważniejsze całkowicie eliminowane jest zagrożenie, że IDS zgubi pakiet. Istotne jest także, że w in-line IDS ataki mogą być skutecznie blokowane w czasie rzeczywistym. Zwykły system IDS, nasłuchując sieć identyfikuje zdarzenia w czasie gdy intruzi wykonali już ataki na chronione zasoby i w praktyce nie jest w stanie ich zablokować. Systemy IDS działające w trybie *in-line* to m.in. Check Point SmartDefence, ISS RealSecure Guard i OneSecure obecnie NetScreen IDP.

Realna wydajność rozwiązań in-line IDS wynosi od 100 do 400 Mb/s, co oznacza, że na razie można je stosować do ochrony segmentów sieci Fast Ethernet oraz mało obciążonych sieci gigabitowych. Według zapowiedzi producentów w najbliższej przyszłości pojawią się rozwiązania in-line IDS o przepływności ponad 1 Gb/s (np. OneSecure zostanie zaimplementowany jako ASIC[1] i powinien mieć możliwość rekonfiguracji systemu zaporowego. Jednak sens tej koncepcji jest raczej umiarkowany. Szczegółowa analiza takiego rozwiązania pokazuje, iż automatyczne blokowanie na firewallu adresów, z których wywodzi się atak wydaje się nierozsądne z kilku powodów. Po pierwsze, włamywacze posługują się zazwyczaj komputerami/sieciami osób i firm trzecich. W trakcie prowadzenia wielu ataków mogą także wykorzystywać dowolne, przypisane sobie adresy IP. Łatwo wyobrazić sobie sytuację, w której włamywacz wiedząc, lub nawet tylko podejrzewając, że firma stosuje taką strategię obrony wykorzysta do ataku sieci (lub tylko ich adresy) należące do jej kluczowych partnerów lub klientów. W ciągu kilku minut firma może więc sama pozbawić się kontaktu ze światem.

[1] Implementacja algorytmów zabezpieczeń w specjalizowanych układach scalonych ASIC (ang. *Application-Specific Integrated Circuit*)

Po drugie, wziąwszy pod uwagę stosunkowo dużo fałszywych alarmów generowanych przez systemy IDS, taka automatyzacja może oznaczać w praktyce dla administratorów więcej, a nie mniej pracy.

Inna koncepcja integracji systemu typu IDS z Firewall, znajdująca większe zastosowanie praktyczne, polega na utrzymywaniu wspólnej bazy rejestrowanych przez nie zdarzeń. System IDS przesyła w czasie rzeczywistym logi do stacji zarządzającej Firewall, aby tam mogły zostać poddane wspólnej analizie. Takie podejście stwarza lepsze możliwości wykrywania nadużyć bezpieczeństwa i wyjaśniania zaistniałych incydentów. Przykładem może być opracowany przez Check Point protokół ELA (ang. *Event Logging API*), poprzez który zdarzenia z IDS mogą w czasie rzeczywistym być przesyłane do konsoli zarządzania Firewall (SmartCenter) [16, 17].

IPS (ang. *Intrusion Prevention Systems*) – jeszcze do niedawna były to rozwiązania, które masowo były porównywane do trochę bardziej rozbudowanych, intuicyjnych i efektywnych systemów IDS. Do głównych zadań systemów IPS należy sama prewencja. Nieco mniej istotne jest alarmowanie administratora o mającym właśnie miejsce włamaniu. Ważne jest szybkie zakończenie trwającego właśnie ataku. System IDS jest potencjalnie powolny – analizuje pakiety, komunikuje się ze swoimi bazami sygnatur, przekazuje informacje do firewalla, tworzy określone reguły itd.. Daje to wystarczająco dużo czasu potencjalnemu włamywaczowi. W systemach IPS wszystkie te procedury schodzą na dalszy plan; system taki ma za zadanie natychmiast przerwać atak a dopiero w następnej kolejności podjąć odpowiednie procedury formalne (tutaj funkcja informowania o mającym miejsce zajściu). W dobie dzisiejszego szybkiego rozwoju technologii, bardzo często już istniejące systemy IDS przejmują – bądź już przejęły – niektóre funkcje systemów IPS. Dlatego trudno jest rozróżniać poszczególne produkty pod względem ich przynależności. Jednak odnośnie ich funkcjonalności chyba najlepiej mówią same ich nazwy – IDS, Intrusion Detection Systems,- IPS, Intrusion Prevention Systems – są to systemy, których głównym zadaniem jest czynna ochrona sieci, niekoniecznie identyfikacja, czy chociażby ew. raportowanie. Dla przykładu, typowym zachowaniem dla systemu IPS jest m.in. przechwytywanie wywołań systemowych, „uodparnianie" stosu, podkładki programowe, czy zmiana danych w warstwie aplikacji. Dokładność systemu IPS musi być znacząco większa niż ma to miejsce w systemach IDS. Często rozważa się instalowanie programów (lub ich całych pakietów) z rodziny IPS na różnych warstwach jednocześnie, m.in. warstwa łącza danych, warstwa transportu oraz warstwa aplikacji (np. poprzez oddzielny proces serwera internetowego w celu analizowania szyfrowanych strumieni czy chociażby dla ogólnego podniesienia poziomu bezpieczeństwa samego serwera). Ostatnio bardzo często systemy IPS przybierają także formę kompleksowych rozwiązań mających na celu chronić maszynę w sieci. W takim znaczeniu – IPS to z reguły Firewall z funkcją IDS/IPS, Antywirusa, narzędzi zapewniających prywatność i poufność. Widać iż, rozwiązania typu IPS, które niejako powstały i ewoluowały ze statycznych systemów IDS, poprzez dynamiczne i intuicyjne systemy IDS/IPS, coraz częściej przybierają formę całościowych zestawów oprogramowania, których priorytetem jest zapewnienie bezpieczeństwa sieci [18].

System SNORT mimo, iż jest bardzo dobrą aplikacją nie jest pozbawiona wad. Należy do nich przede wszystkim pomijanie części pakietów, generowanie fałszywych alarmów, ignorowanie prawdziwych alarmów. Również uaktualnianie aplikacji jest mało przyjemnym zajęciem. Trzeba pamiętać, iż zestawy reguł oraz interfejs dzienników alarmów ulegają ciągłym zmianom.

Niestety bardzo często sama aplikacja staje się obiektem ataku. Ostatnio Sourcefire poinformowało o odkryciu luki w SNORT, która umożliwia zainstalowanie dowolnego kodu w systemie monitorującym i przejęcie całkowitej kontroli nad zarażoną maszyną. Ta luka jest związana z błędem przepełnienia bufora w protokole DCE/RPC. Jest on co prawda domyślnie aktywny, jednak zwykle generowany przez niego ruch jest blokowany na firewallu. Dlatego też większość użytkowników systemu SNORT nie powinna być narażona na niebezpieczeństwo. Luki wykryto w wersjach 2.6.1, 2.6.1.1 oraz 2.6.1.2 oraz 2.7.0. Problem rozwiązuje zainstalowanie wersji 2.6.1.3. Dla SNORT 2.7.0 poprawki nie zostały jeszcze opublikowane. Specjaliści radzą, by użytkownicy tej wersji wyłączyli DCE/RPC [19].

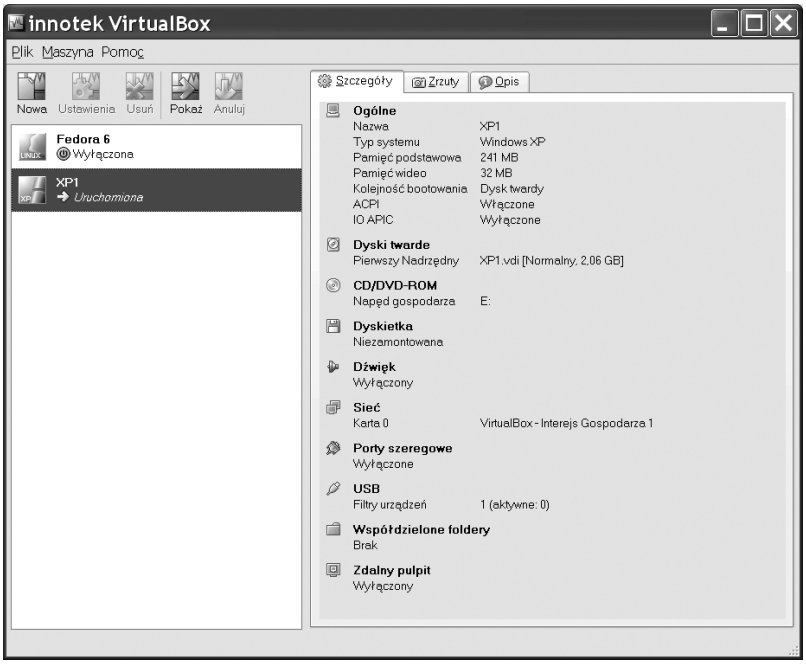
Jeżeli założymy, że nasza aplikacja jest bezpieczna, to problemem mogą być luki w systemie operacyjnym. Konieczne będzie zatem wykonanie podstawowych czynności, które mogą podnieść ogólny poziom zabezpieczeń systemu, tj. wyłączenie niepotrzebnych usług, zagwarantowanie spójności, zastosowanie firewalla, stosowanie szyfrowania oraz aktualizacja pakietów oprogramowania.

Budowany system IDS może składać się z większej liczby aplikacji SNORT. Wprowadzenie redundancji zwiększa niezawodność rozwiązania i poziom zabezpieczenia sieci. Opisana wcześniej struktura może być wykorzystana jako pasywne lub też aktywne monitorowanie sieci.

3. WARSZTATY. PRAKTYCZNA REALIZACJA ĆWICZEŃ

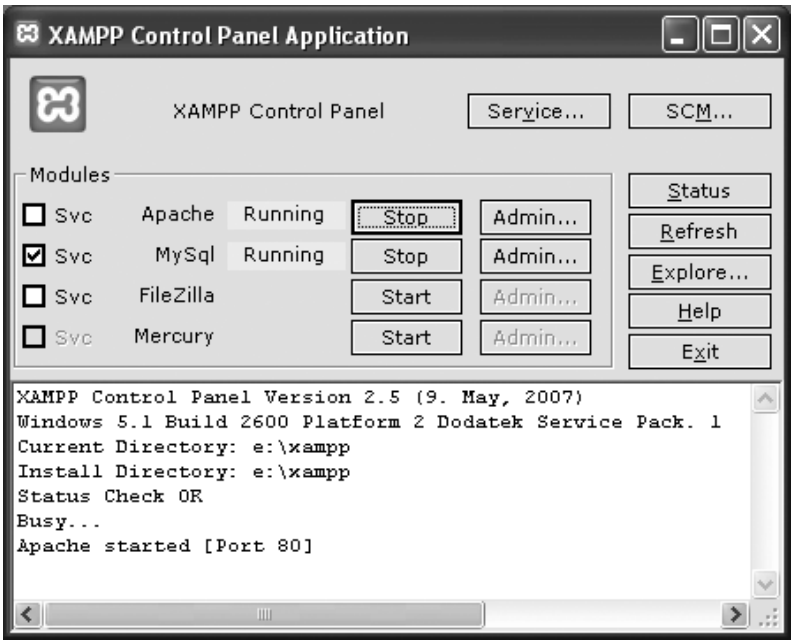
PRZYGOTOWANIE ŚRODOWISKA

W środowisku VirtualBox v. 1.5.4 uruchomić maszynę wirtualną XP1 (rys. 7).



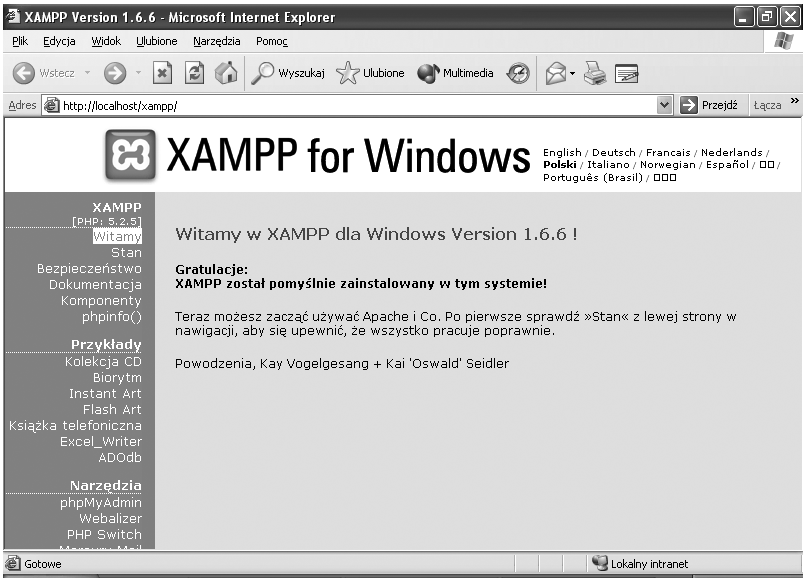
Rysunek 7.
Uruchomienie maszyny wirtualnej XP1

Następnie uruchomić usługę Apache oraz MySql tak, aby otrzymać status Running (rys. 8).



Rysunek 8.
Uruchomienie usług Apache i MySql

Zweryfikować poprawność instalacji XAMPP poprzez wpisanie w oknie przeglądarki: http://localhost/xampp (rys. 9).

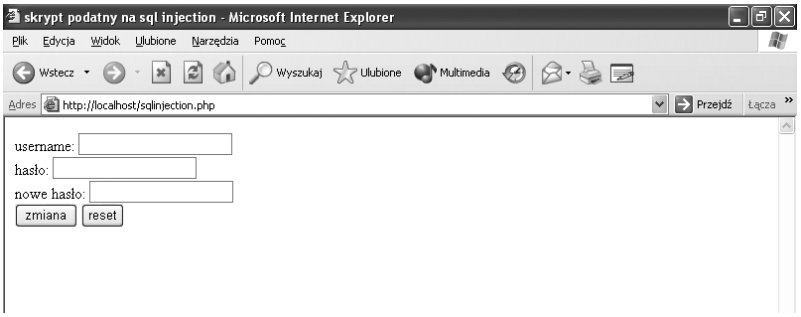


Rysunek 9.
Weryfikacja poprawności instalacji XAMPP

W środowisku testowym w lokalizacji (c:\snort\snort_pliki) znajdują się następujące pliki, które należy wykorzystać:

1. `sqlinjection.php` – skrypt służący do zmiany hasła użytkownika w testowej bazie danych (autoryzacja odbywa się poprzez stare hasło),
2. `exploit.exe` – aplikacja do uruchomienia na komputerze atakującym.

Test działania skryptu `sqlinjection.php` (rys. 10).

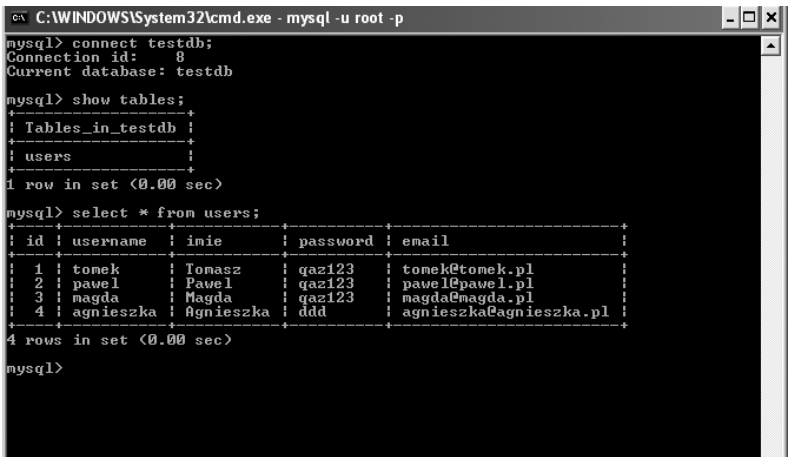


Rysunek 10.
Testowanie skryptu

Za pomocą tego skryptu można zmieniać hasła użytkownikom, którzy znajdują się w bazie danych. Aby sprawdzić zawartość bazy danych należy:

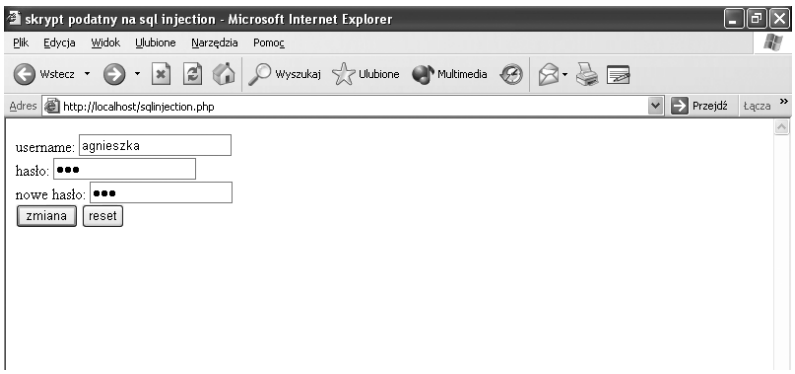
1. wejść do katalogu: `E:\xampp\mysql\bin`
2. uruchomić: `mysql –u root –p (zatwierdzić bez hasła)`
3. połączyć się z bazą: `connect SNORT`
4. wyświetlić dostępne bazy: `show databases` (rys. 11)
5. połączyć się z baza testdb: `connect testdb`
6. wyświetlić dostępne tabele: `show tables`
7. wyświetlić zawartość tabeli `users`: `select * from users`

Rysunek 11.
Dostępne bazy danych



Rysunek 12.
Zawartość tabeli `users`

Hasło dla użytkownika `Agnieszka` to `ddd`, zmieniamy je na inne, np. `eee` (rys. 13).

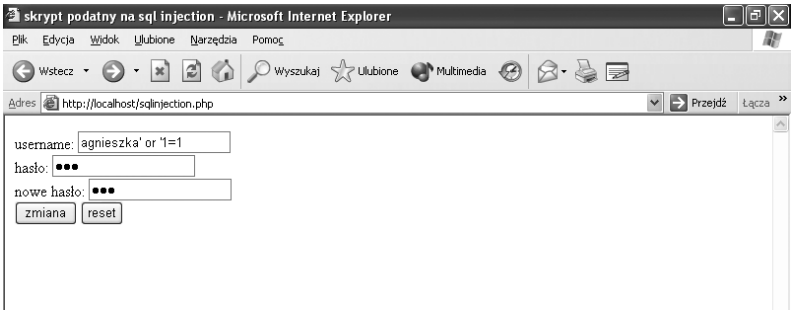


Rysunek 13.
Zmiana hasła użytkownika `Agnieszka`

Podglądamy w bazie, czy hasło zostało rzeczywiście zmienione? Widać, że hasło zostało zmienione na `eee` (rys. 14).

Rysunek 14.
Potwierdzenie zmany hasła

Stosując technikę `sqlinjection` spróbujmy zmienić hasło (nadać nowe) bez znajomości starego. Wykonujemy to wpisując w pole username (dla użytkownika agnieszka): `Agnieszka' or '1=1` (rys. 15) i sprawdzamy w bazie danych: widać, że hasło zostało zmienione na `ccc` (rys. 16).



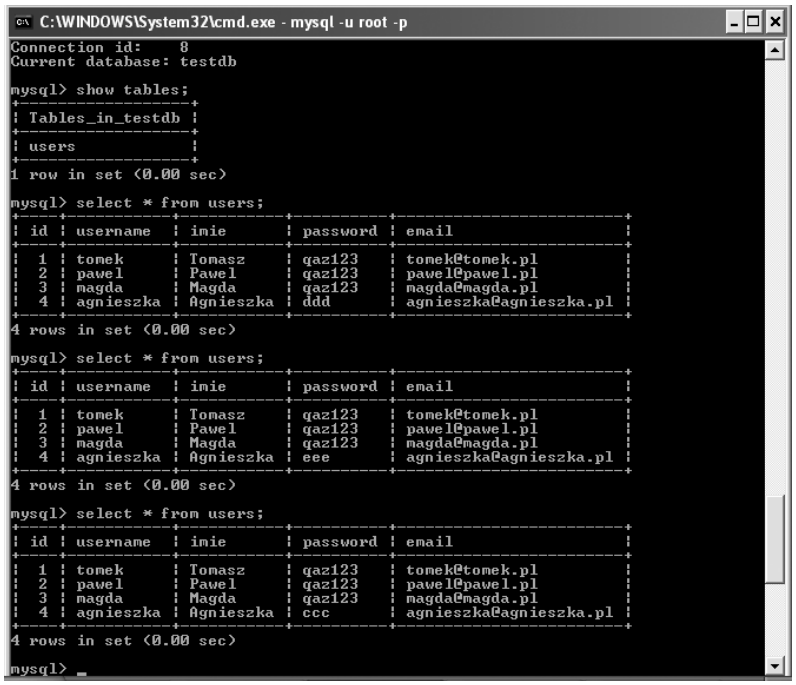
Rysunek 15.
Ponowna próba zmiany hasła

Zadanie do samodzielnego wykonania

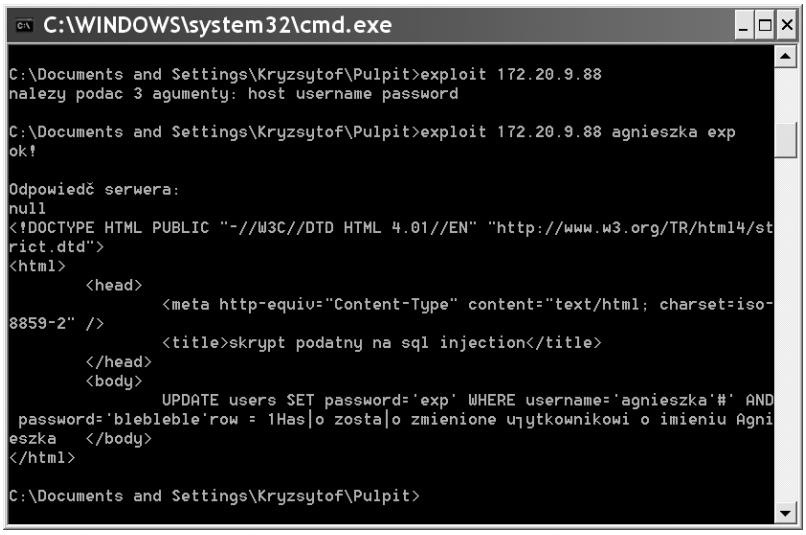
Mając podstawowe umiejętności w zakresie obsługi bazy, należy przeprowadzić atak SQL Injection na dołączony skrypt PHP – napisać regułę dla systemu NIDS SNORT, który w przypadku wykorzystania programu `exploit` będzie generował alert.

Wskazówki, jak należy postępować:

1. Na zdalnej maszynie uruchamiany program `exploit` (znajduje się on `c:\snort\snort _ pliki`) i wpisujemy: `exploit <adres IP atakowanego hosta> <username> <nowe haslo>`
   Np.: `exploit 172.20.9.88 agnieszka exp` (rys. 17).



Rysunek 16.
Potwierdzenie ponownej zmiany hasła

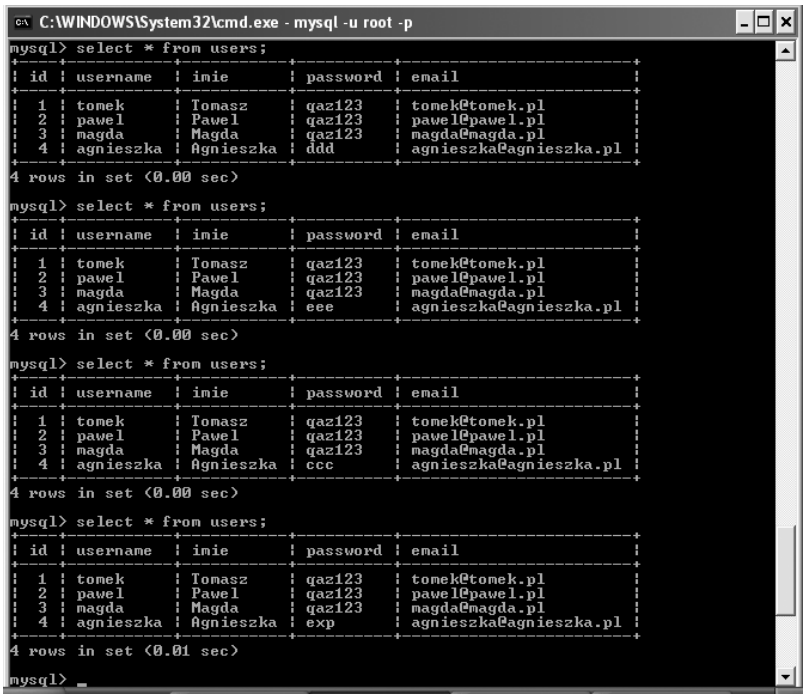


Rysunek 17.
Kolejna próba zmiany hasła

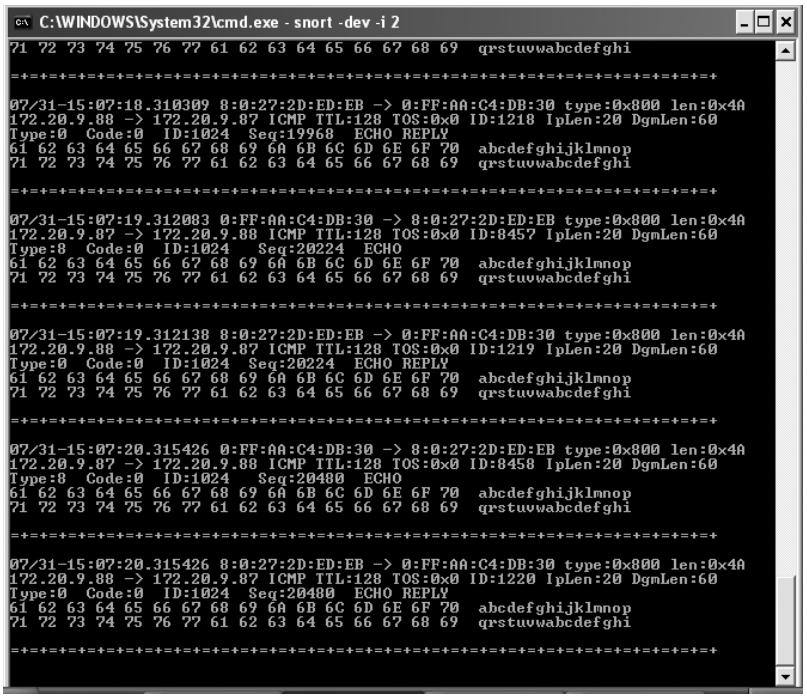
2. Sprawdzamy czy hasło zostało zmienione (powinno być teraz `exp` rys. 18). Hasło zostało zmienione, `exploit` skutecznie zmienił hasło bez znajomości starego.

3. Włączamy ochronę w postaci programu SNORT – wchodzimy do folderu `C:\snort\bin` i uruchamiamy ten program (w trybie monitorowania): `snort –dev –i 2`

   W oknie odpowiedzi na polecenie `ping` (host atakujący) na hoście, na którym uruchomiono program SNORT, powinniśmy uzyskać informację jak na rys. 19.

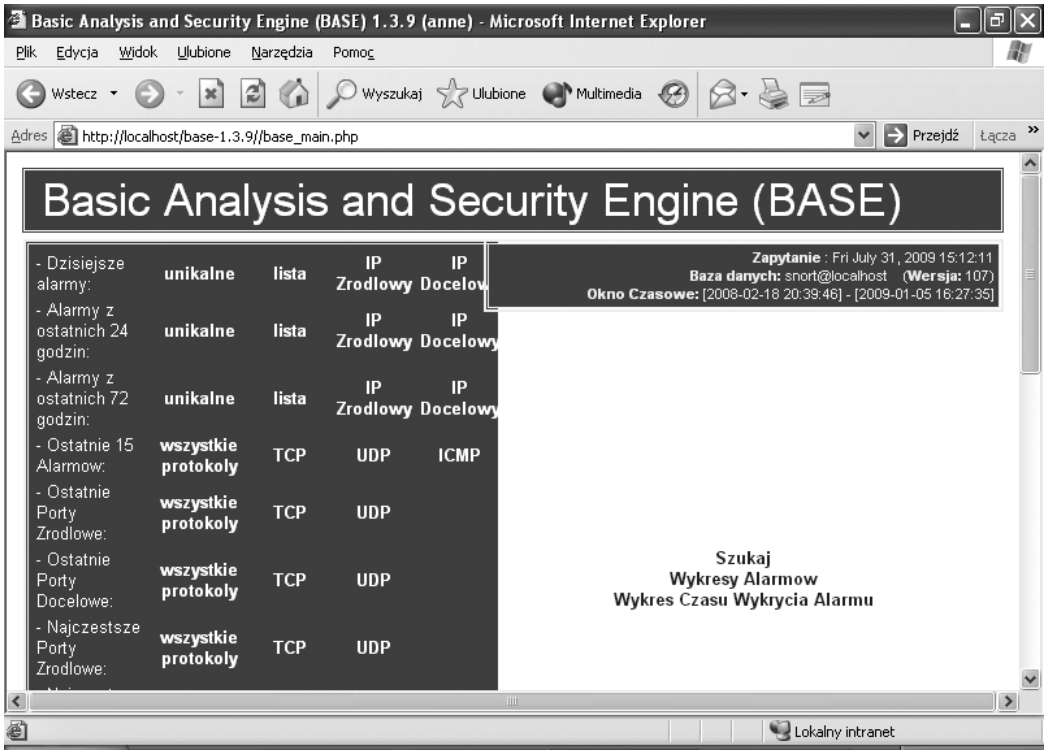


Rysunek 18.
Potwierdzenie zmiany hasła

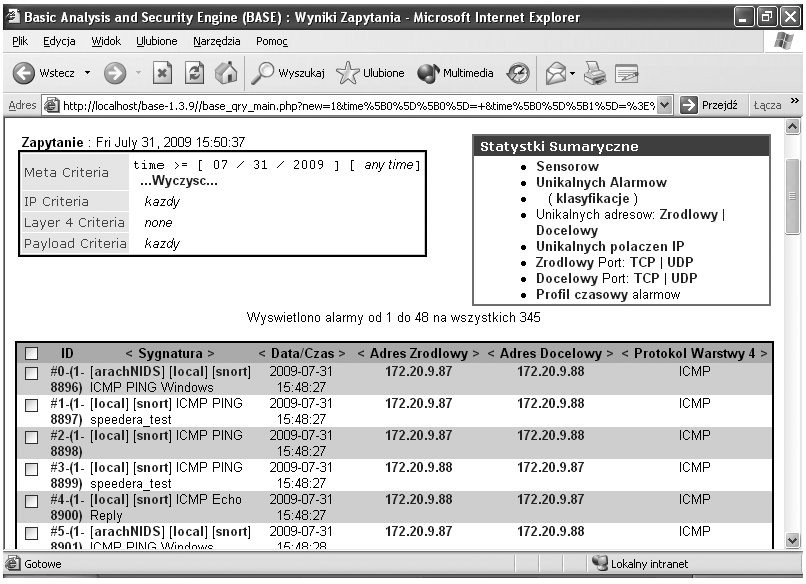


Rysunek 19.
Reakcja na polecenie `ping`

4. Sprawdzamy, czy jakaś informacja została zalogowana w systemie IDS/IPS SNORT. W tym celu otwieramy okno aplikacji BASE (rys. 20) oraz uruchamiamy system SNORT: `snort –dev i 2 –c c:\snort\etc\snort.conf –l c` i sprawdzamy w BASE (rys. 21). Widać, że zarejestrowano proces zainicjowany poleceniem `ping` z hosta 172.20.9.87.

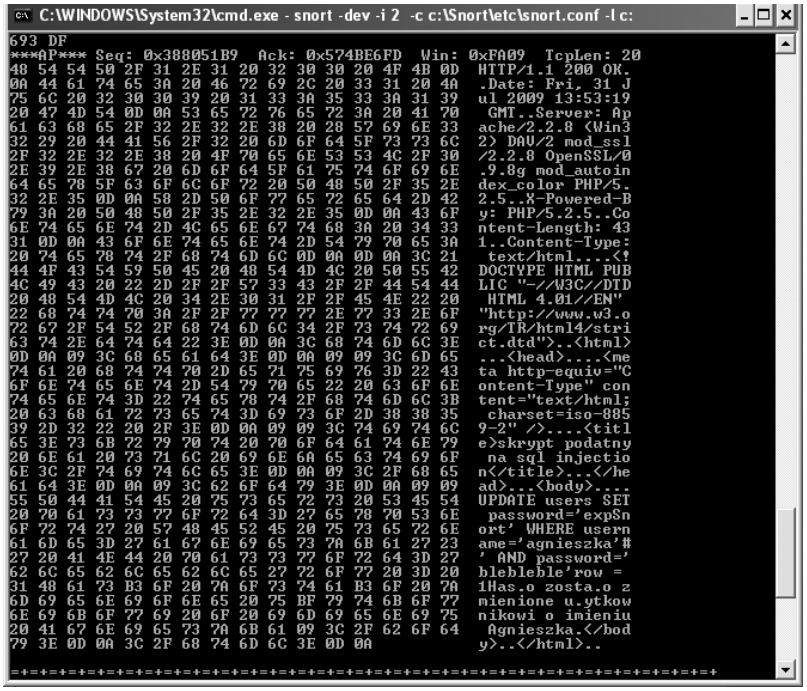


Rysunek 20.
Okno aplikacji BASE

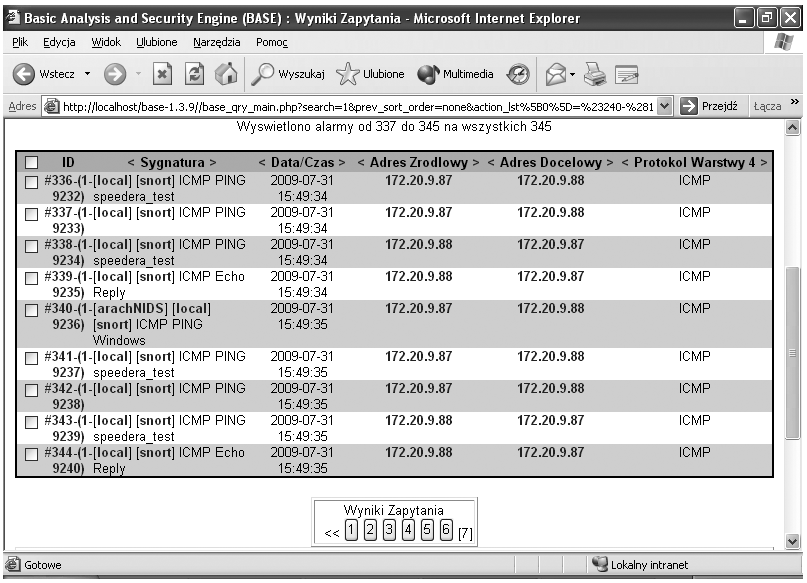


Rysunek 21.
Efekt uruchomienia systemu SNORT w systemie BASE

5. Włączamy ponownie program SNORT w celu wykrycia włamania poprzez program `exploit`, a następnie uruchamiamy ten program na zdalnym hoście. Na atakowanym hoście obserwujemy (rys. 22). W aplikacji BASE widać, iż nie został jednak wygenerowany żaden alert związany z działaniem `exploit` (rys. 23).



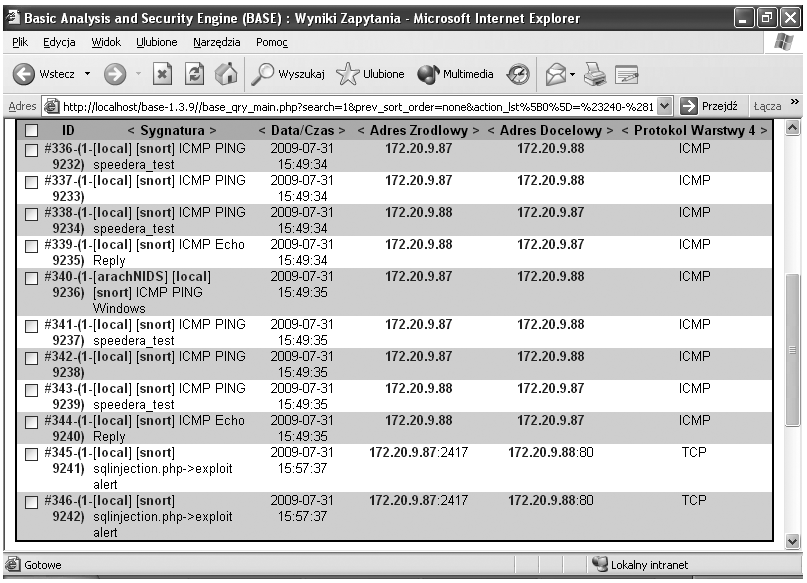
Rysunek 22.
Obserwowanie atakowanego hosta



Rysunek 23.
Podgląd aplikacji BASE

Należy zatem napisać regułę tak, aby w efekcie otrzymać wygenerowany alert (rys. 24). Reguła, którą można dopisać np. do `c:\snort\rules\exploit.rules`, może mieć następująca postać:

```
alert tcp any any -> any any (msg:"sqlinjection.php->exploit alert"; content:"application/x-www-form"; sid:999999;)
```



Rysunek 24.
Alert po dopisaniu odpowiedniej reguły

PODSUMOWANIE

Właściwy wybór systemu IDS/IPS, adekwatny do potrzeb firmy, jest jednym z kluczowych elementów realizacji polityki bezpieczeństwa teleinformatycznego. Oczywiście stanowi jedynie część tzw. warstwowego systemu ochrony dogłębnej. Istnieje wiele rozwiązań typu open-source, jak również komercyjnych. Decyzja o wyborze uzależniona jest od wielu czynników, w tym również finansowych, ale przede wszystkich powinna być zgodna z polityką bezpieczeństwa teleinformatycznego firmy oraz zapisami w planie bezpieczeństwa teleinformatycznego.

Wykrywanie włamań jest podstawowym celem stosowania tychże systemów. Ich skuteczność, pomijając architekturę samego narzędzia, jest uzależniona od zdefiniowania akceptowalnego ruchu sieciowego. Rozwiązania tego typu doskonale sprawdzają się w niewielkich sieciach, które wymagają wysokiego poziomu zabezpieczeń. Jednak aktywny system IDS nie zawsze musi stanowić najlepsze rozwiązanie dla sieci. Może spowolnić przepływ danych nawet do poziomu niedopuszczalnego. Dodatkowo błąd w konfiguracji może doprowadzić do zakłócenia poprawnej pracy sieci. Pakiety połączeń nie stanowiące zagrożeń mogą być pomyłkowo odrzucane, co może spowodować brak komunikacji między jednostkami sieci.

W dużych sieciach korporacyjnych, które mają co najmniej jedno łącze internetowe o dużej wydajności, stosuje się wysoko wydajne firewalle przetwarzające znaczne ilości danych w krótkim czasie. Mają one specjalne reguły gwarantujące szybkie i efektywne przenoszenie ruchu. Aktywne IDSy, ich reguły mogą ten ruch spowolnić.

Możliwym, ciekawym zastosowaniem aktywnych IDS są tzw. **pułapki**. Te równoległe rozwiązanie ze standardowymi konfiguracjami sieciowymi, umożliwia wykrywanie nowych form ataków i dokładniejsze sprawdzenia działania firewalla.

Innym przykładem wykorzystania aktywnego IDS jest objęcie jego zasięgiem działania jedynie pewnego fragmentu działania sieci, w którym przetwarzane są szczególnie krytyczne z punktu widzenia dane.

Za pomocą pakietów LogSentry i PortSentry można chronić system na poziomie operacyjnym. LogSentry zbiera i porządkuje wiadomości dzienników zdarzeń, które mogą wskazywać na próby włamania, natomiast PortSentry podejmuje aktywny udział w ochronie systemu przed włamaniami z poziomu sieci. Wykorzystanie aplikacji SNORT, pracującej w trybie aktywnym (*in-line*), umożliwia selektywne odrzucanie pakietów w zależności od zapisanych w nim adresów stacji, numerów portów i zawartości pola danych.

BIBLIOGRAFIA

1. Cole E., *Bezpieczeństwo sieci*, Helion, Gliwice 2005
2. Liderman K., *Podręcznik administratora bezpieczeństwa teleinformatycznego*, MIKOM, Warszawa 2003
3. Liderman K., *Bezpieczeństwo teleinformatyczne*, WAT, Warszawa 2006
4. Lockhart A., *100 sposobów na bezpieczeństwo sieci*, Helion, Gliwice 2004
5. Negus C., *Fedra Core 2*, Helion, Gliwice 2005
6. Shimorski R. J., *Wielka Księga Firewalli*, Helion, Warszawa 2004
7. Stallings W., *Ochrona danych w sieci i intersieci*, WNT, Warszawa 1997
8. Rash M., *IPS zapobieganie i aktywne przeciwdziałanie intruzom*, Mikom, Warszawa 2005

NETOGRAFIA

9. http://pl.docs.pld-linux.org/uslugi_snort.html
10. http://easyfwgen.morizot.net/gen
11. http://www.netfilter.org
12. http://www.howtoforge.com/intrusion_detection_base_snort_p4
13. http://www.snort.org
14. http://base.secureideas.net
15. http://www.hakin9.org
16. http://www.pckurier.pl
17. http://wiki.kis.p.lodz.pl
18. http://pl.docs.pld-linux.org/uslugi_snort.html
19. http://www.opsec.com/solutions/sec_intrusion_detection.html
20. http://www.clico.pl
21. http://www.nfsec.pl/publikacje/bezpieczenstwo_w_teorii/ids_ips.html
22. http://hack.pl/aktualnosci/dziurawy_snort_527
23. http://www.howtoforge.com/intrusion_detection_base_snort_p4
24. http://www.bezpieczna-siec.com/ataki2-opisy.htm#wirus
25. http://www.networld.pl/artykuly

W projekcie **Informatyka +**, poza wykładami i warsztatami, przewidziano następujące działania:

- 24-godzinne kursy dla uczniów w ramach modułów tematycznych
- 24-godzinne kursy metodyczne dla nauczycieli, przygotowujące do pracy z uczniem zdolnym
- nagrania 60 wykładów informatycznych, prowadzonych przez wybitnych specjalistów i nauczycieli akademickich
- konkursy dla uczniów, trzy w ciągu roku
- udział uczniów w pracach kół naukowych
- udział uczniów w konferencjach naukowych
- obozy wypoczynkowo-naukowe.

Szczegółowe informacje znajdują się na stronie projektu

www.informatykaplus.edu.pl