

Open Source vs Commercial Product

Porównanie dwóch zapór sieciowych ze zdolnością zestawiania prywatnych sieci wirtualnych VPN

Bartosz Krajnik
Instytut Informatyki Politechniki Warszawskiej
bmk@indigo.pl

Streszczenie

Głównym tematem pracy jest porównanie dwóch zapór sieciowych: Cisco 2621 oraz Linux Debian z kerneliem 2.4.20. Najpierw opisano ich działanie, co powinny one robić, dlaczego zapory sieciowe są potrzebne dla firm, a następnie przedstawione zostały krótkie charakterystyki wymienionych wyżej zapór. Praca zawiera także porównanie wydajnościowe opisywanych urządzeń.

1. Co to jest zaporę sieciową?

Zapora sieciowa jest urządzeniem sieciowym położonym pomiędzy dwoma różnymi sieciami (zazwyczaj pomiędzy jakąś organizacją a internetem) i chroniącą pewne określone hosty znajdujące się w sieci wewnętrznej przed atakami z zewnątrz. Często mianem zapór sieciowych określa się pojedyncze hosty z zainstalowanym oprogramowaniem mającym na celu ochronę ich samych. Tematem tej pracy są jednak komputery z dwoma (lub więcej) kartami sieciowymi stojące na granicy dwóch (lub więcej) sieci i filtrujące dane pomiędzy nimi.

2. Co powinna robić zaporę sieciową?

Zapora sieciowa zapewnia, że cała komunikacja pomiędzy dwoma sieciami (wewnętrzną i zewnętrzną) będzie poddana polityce bezpieczeństwa firmy. Zapory sieciowe kierują i kontrolują komunikacją decydując co przepuszczać a co odrzucać. Dodatkowo zabezpieczając zaufaną sieć przed internetem zapory sieciowe mogą odpowiednio chronić pewien wydzielony obszar sieci wewnętrznej lub indywidualne hosty.

3. Dlaczego firmy potrzebują zapór sieciowych?

Firmy na całym świecie wykorzystują technologie internetowe do zawierania korzystnych znajomości biznesowych. Zapory sieciowe pomagają firmom równoważyć otwarcie internetu z potrzebą zabezpieczenia prywatności i integralności komunikacji biznesowych.

4. Jak działają zapory sieciowe?

Historycznie są wykorzystywane trzy różne technologie implementacji zapór sieciowych:

filtrowanie pakietów, bramy warstwy aplikacji i kontrola stanów.

Filtrowanie pakietów – zazwyczaj implementowane na ruterach, filtrują ruch w oparciu o zawartość pakietów, np. adresy IP. Sprawdzają pakiet w warstwie sieciowej i są niezależne od aplikacji, zapewniają dobrą skalowalność i prędkość. Są najmniej zabezpieczającymi typami zapór sieciowych. Ponieważ nie są świadome aplikacji, nie mogą zrozumieć kontekstu danej komunikacji robiąc je łatwymi do złamania.

Bramy warstwy aplikacji – poprawiają bezpieczeństwo egzaminując całą warstwę aplikacji przenosząc kontekst informacji do procesu decydującego - jakkolwiek łamią zasadę modelu klient/serwer. Każda komunikacja klient/serwer wymaga tutaj dwóch połączeń: jednego od klienta do zapory sieciowej (która działa jako zastępstwo dla określonego serwera) i jednego od zapory sieciowej do rzeczywistego serwera. W dodatku każda aplikacja wymaga nowego „proxy” czyniąc skalowalność i obsługę problematyczną.

Kontrola stanów – wprowadza najwyższy poziom zabezpieczeń i przewyższa poprzednie dwie metody rozumiejąc filtrowanie warstwy aplikacji bez łamania modelu klient/serwer. Rozpoznaje stany połączeń ze wszystkich aplikacji i alokuje te informacje w dynamicznej tablicy. Wprowadza to rozwiązanie w pełni bezpiecznie i oferujące maksymalną prędkość, skalowalność oraz niezawodność.

5. Open Source przeciw komercyjnym zaporom sieciowym - porównanie funkcjonalne

Opisujemy tutaj eksperyment porównujący dwie zapory sieciowe ze zdolnością zestawiania prywatnych sieci wirtualnych (VPN) poprzez internet. Jedna jest rozwiązaniem Open Source, druga będąca produktem komercyjnym. Ponieważ rozwiązania typu Open Source są mniej drogie i oferują większą kontrolę, rozwiązania komercyjne dostarczają produkt relatywnie dobry, droższy ale i bardziej elastyczny. Rozwiązania Open Source podążają za zasadą, że im więcej użytkowników będzie pracowało z kodem, tym więcej będzie on bezpieczny i tym ciężiej będzie go skompromitować. Komercyjne

rozwiązania argumentują ukrywanie kodu źródłowego jako zmniejszenie prawdopodobieństwa na złamanie systemu. Dane dotyczące zapory sieciowej Cisco pochodzą z „Illinois State University - Department of Applied Computer Science”.

6. Porównanie wybranych zapór sieciowych

Pierwsza zapora sieciowa użyta w tym porównaniu jest skonstruowana używając rozwiązania Open Source dostępnego dla systemu Debian Woody 3.0 GNU/Linux i na procesorze Pentium z dwoma kartami sieciowymi. Druga zapora sieciowa była zbudowana używając routera Cisco 2621 z systemem Cisco IOS. Całe oprogramowanie użyte na systemie Linux jest wolno dostępne na zasadzie Open Source i może być używane na różnych platformach sprzętowych, podczas gdy Cisco jest produktem komercyjnym obejmującym zarówno sprzęt jak i oprogramowanie.

7. Filtrowanie danych sieciowych w systemie Linux

System Linux z kernelem 2.4 ma zaimplementowany filtr pakietów netfilter - nazywany iptables. Użyto wersji iptables-1.2.6a w połączeniu z systemem Debian GNU/Linux w wersji Woody 3.0. Iptables dokonywało rewizji wg następujących kryteriów: adres IP źródłowy/przeznaczenia, protokół, nazwa DNS, interfejs i stan połączenia.

8. Filtrowanie danych sieciowych w systemach Cisco

Cisco IOS ma wbudowaną obsługę filtrowania trzech typów: statyczne filtrowanie pakietów (może być użyte przy użyciu standardowej/rozszerzonej listy dostępu do filtrowania innych protokołów), zwrotnej listy dostępu (jest używane do prowadzenia filtrowania w oparciu o stany pakietów przy użyciu Content Based Access Control - CBAC). Zwrotne listy dostępu są używane aby zezwolić użytkownikom w zabezpieczonej sieci elastycznie wysyłać dane i odpowiednio przepuszczać pakiety przychodzące w odpowiedzi. System Cisco IOS ma właściwość znaną jako dostęp „Lock and Key”, która łączy autentykację hasłem z filtrowaniem pakietów opartym na stanach połączeń. Użytkownik musi wykonać poprawny proces autentykacji do routera albo innego serwera (Radius albo TACACS+) zanim zapora sieciowa zezwoli na przepuszczanie danych.

9. Porównanie filtrowania warstwy aplikacji

Celem obsłużenia filtrowania w warstwie aplikacji zapora sieciowa zbudowana na systemie operacyjnym Linux używa oprogramowania „TIS firewall toolkit”

w wersji 2.1. „TIS firewall toolkit” jest zbiorem aplikacji pośredniczących, które prowadzą dodatkową kontrolę wraz z logowaniem takich protokołów jak WWW, FTP, TELNET i SMTP. Jest również dostępna aplikacja „HTTP-Gopher proxy” i może być ona skonfigurowana do obsługi filtrowania stron WWW opartych o takie technologie jak Java, JavaScript czy ActiveX w zależności od hosta i od poziomu zabezpieczeń jego przeglądarki internetowej. Aplikacja „TIS firewall toolkit” poprzez pośredniczenie w protokole SMTP poprawia bezpieczeństwo wewnętrznego serwera SMTP czyniąc go trudnym do złamania podczas ataku. Zapora sieciowa Cisco IOS scala kilka dodatków poziomu aplikacji zwiększając bezpieczeństwo połączeń. Pierwszym z nich jest blokowanie Java w zastosowaniu do bloków appletów Javy dla ruchu webowego przy użyciu listy dostępu. Ta implementacja blokowania Javy ma jedną wadę w stosunku do Linuxa: aplikacja „TIS firewall toolkit” ma zdolność do współpracy z przeglądarką użytkownika bez potrzeby tworzenia oddzielnej listy zaufanych stron. Zapora sieciowa Cisco IOS nie ma zdolności do filtrowania aplikacji ActiveX.

10. Porównanie zdolności VPN

Dla zapory sieciowej Linux w celu implementacji VPN wybrano aplikację VPND wraz z SSH przy użyciu protokołu PPP ze względu na mocny algorytm szyfrujący i scalony schemat kompresji. Aplikacja VPND używa szyfrowania typu Blowfish ze 128 bitowym algorytmem. Jeśli wymagana jest obsługa w oparciu o IPSEC, dostępna jest aplikacja Free S/WAN. Oprogramowanie Cisco IOS jest zdolne do utworzenia VPN-ów używając technologii szyfrującej Cisco (CET) lub IPSEC. Standardowe IPSEC jest skalowalne co czyni je większą korzyścią dla firm z sieciami heterogenicznymi. Technologia CET jest własnością firmy Cisco która bazuje na standardowej sygnaturze cyfrowej (DSS), algorytmie klucza publicznego Diffie-Hellman'a (DH) i standardzie kodowania danych (DES).

11. Rezultaty eksperymentów

Dokonano serii eksperymentów bezpośrednio porównując zaporę sieciową bazującą na systemie operacyjnym Linux z zaporą sieciową Cisco IOS. Zmieniano przy tym liczbę wpisów w każdej zaporze i rozmiar pakietów. Używano transakcji TCP/UDP oraz strumieni TCP/UDP. Transakcje TCP są najbardziej miarodajną oceną wydajności zapory sieciowej biorąc pod uwagę źródło danych przepływającego przez zaporę sieciową pakietu o określonym rozmiarze (pamiętając o liczbie wpisów)

oraz pakiet powracający do źródła poprzez tą samą zaporę sieciową z tą samą liczbą wpisów. Wpisy do zapory sieciowej były przewidywane dla najgorszych przypadków tak aby wszystkie reguły były sprawdzone w każdym przypadku. Standardowa zapora sieciowa Cisco 2621 posiada 50 MHz procesor RISC z 50 M pamięci. Zapora sieciowa LINUX posiada 350 MHz procesor Pentium PC z 96 M pamięci.

Rozmiar pakietu = 1 bajt					
Ilość wpisów	0	50	100	150	200
Cisco	2541	1604	1434	1312	1216
Linux	5277	4293	4377	4033	3574
Rozmiar pakietu = 128 bajtów					
Ilość wpisów	0	50	100	150	200
Cisco	571	120	119	118	116
Linux	724	718	705	694	670

Tabela 1. Średni stosunek transakcji TCP (na sekundę).

Tabela 1 pokazuje, że zapora sieciowa Linux ma stosunkowo wyższą przepustowość transakcji dla ilości wpisów pomiędzy 0 a 200 i rozmiarów pakietów: 1 bajt i 128 bajtów. Były robione wielokrotne testy w celu osiągnięcia 99% pewności co do różnicy pomiędzy tymi wartościami.

12. Podsumowanie

Ten eksperyment pokazuje, że dla tego specyficznego przykładu zapora sieciowa Linux ma znaczącą przewagę w obsłudze transakcji, posiada także większe możliwości filtrowania w warstwie aplikacji. Zapora sieciowa Cisco IOS jest bardziej funkcjonalna przy filtrowaniu w warstwie sieciowej, jest zintegrowana z heterogenicznym wieloprotokołowym środowiskiem i skalowalna do obsługi PKI. Nie powinniśmy pomijać, że kilka projektów pod środowisko Linux znajdujących się jeszcze w fazie eksperymentalnej może dużo zmienić. Ostatecznie najbardziej efektywnym rozwiązaniem może być kombinacja obu – filtrowania w warstwie aplikacji i w warstwie sieciowej.

Bibliografia

- Cheswick W. R., Bellovin S. M.: *Firewalls and Internet Security: Repelling the Wily Hacker*. Reading, Mass., Addison-Wesley 1994.
- Garfinkel S. L., Spafford E. H.: *Practical UNIX and Internet Security*. Wyd. 2, Sebastopol Calif., O'Reilly & Associates 1996.

Kent S. T.: *U.S. Department of Defense Security Options for the Internet Protocol*. RFC 1108, Niev. 1991.

Muffet A.: *FAQ: Computer Security Frequently Asked Questions*. Usenet, alt.security, comp.security.misc, comp.security.unix, news.answers, Dec. 1993.

Postel J. B.: *Internet Protocol*. RFC 791, Sept. 1981.

Postel J. B.: *Transmission Control Protocol*. RFC 793, Sept. 1981.

Postel J. B.: *User Datagram Protocol*. RFC 768, Aug. 1980.

www.cisco.com

www.debian.org

www.firewallguide.com

www.freeswan.org

www.insecure.org

www iptables.org

Patton S., Doss D., Yurcik W.: *Open Source Versus Commercial Firewalls: Functional Comparison*. Annual IEEE Conference on Local Computer Networks (LCN 2000), Tampa FL, USA, Niev. 2000.