

Checkpoint FW1 i Firewall Builder
– porównanie zapór sieciowych z graficznym interfejsem użytkownika

Twórcy reguł

Technologia zastosowana w zapo-
rze sieciowej, umieszczona w jądrze
Linuksa, już od jakiegoś czasu
świetnie sprawdza się w zastoso-
waniach profesjonalnych. Aby umożli-
wić wykorzystanie tych sprawdzo-
nych mechanizmów w zastoso-
waniach komercyjnych, opracowano
narzędzia z graficznym interfejsem
użytkownika.

CHRISTIAN NEY



Przedsiewzięcia korzystające z dzier-
żawionych łączy dostępowych Inter-
netu wymagają lepszej ochrony
przed atakami z zewnątrz. W większości
przypadków zapory sieciowe są pierwszą linią
obrony, umożliwiającą administratorom
systemów określenie rodzaju dopuszczalnego
ruchu do sieci wewnętrznej firmy lub do-
stępu do Internetu z tych sieci.

Rynek zapór sieciowych ciągle się rozwija,
głównie z powodu wzrostu popularności Inter-
netu. Zakres produktów obejmuje zarówno
bezpłatne oprogramowanie typu Open Source,
jak i wyjątkowo kosztowne rozwiązania kom-
ercyjne. Prawie wszystkie produkty komer-
cyjne zapewniają konfigurację w trybie gra-
ficznym i posiadają interfejs administracyjny.
Z kolei zapory typu Open Source (zwykle) są
obsługiwane z linii wiersza poleceń. Jest to je-
den z powodów, dla którego wielu adminis-
tratorów broni się przed użyciem ich w admini-
strowanych systemach.

Oczywiście dobry interfejs nie może zastą-
pić solidnych podstaw wiedzy, ale wielu z nas

woli wizualną obsługę programów niż korzy-
stanie z wiersza poleceń – po prostu łatwiej
jest zastosować opracowany wcześniej sche-
mat zabezpieczeń systemu w programie z in-
terfejsem GUI.

Sieć dobry przykład

Nasze przykłady będziemy opisywać na pod-
stawie sieci pokazanej na Rysunku 1. Poka-
żemy jak zastosować zasady bezpieczeństwa
przy użyciu programu CheckPoint Firewall
1 NG, a także Firewall Buildera oprogramo-
wania typu Open Source. Do połączenia z In-
ternetem nasza sieć korzystać będzie z route-
ra. W fikcyjnym przedsiębiorstwie, będącym
właścicielem tej sieci, podjęto decyzję o zało-
żeniu strony WWW na własnym serwerze.
Zamiast umieszczania serwera w sieci we-
wnętrznej, uruchomimy go w oddzielnej sie-
ci tzw. „strefie zdemilitaryzowanej” (DMZ).
Pracownicy firmy będą mieli dostęp do In-
ternetu przez protokoły FTP, HTTP
i HTTPS i będą chronieni wewnętrznym ser-
werem proxy. Serwer pocztowy w sieci LAN

będzie wykorzystywał protokół POP3 do od-
bierania wiadomości od zewnętrznego usłu-
godawcy. Umożliwi to centralną kontrolę an-
tywirusową wiadomości przychodzących.

Serwer pocztowy będzie jedynym kompu-
terem, przez który będzie można wysłać
pocztę protokołem SMTP. Zapora będzie do-
datkowo zabezpieczona – dostęp do niej bę-
dzie miał tylko główny komputer, który przy
ustanawianiu połączenia będzie wykorzysty-
wał protokół SSH.

Solidny strażnik

Firewall 1 NG (Next Generation) firmy
CheckPoint zdobył już reputację solidnego
strażnika sieci. Firma CheckPoint dostarcza
nie tylko zaporę o doskonałej reputacji w za-
kresie bezpieczeństwa, ale może także (za
opłatą) rozszerzyć właściwości swojego pro-
duktu, zapewniając zintegrowane rozwiąza-
nia wirtualnych sieci prywatnych (VPN),
które z kolei umożliwiają zastosowanie roz-
wiązań z półki Open Source (takich jak Fre-
eS/WAN). Niestety świat oprogramowania

Open Source nie jest obecnie w stanie zapewnić rozwiązania komplementarnego. CheckPoint udostępnia użytkownikom interfejs konfiguracyjny Smart-Dashboard.

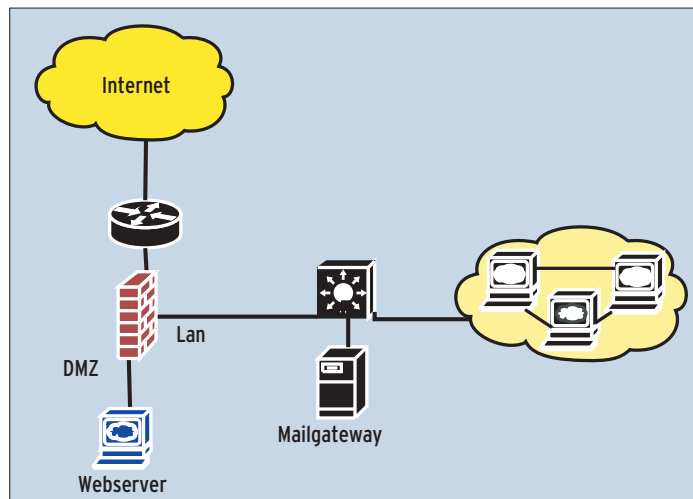
CheckPoint rozróżnia dwa moduły interfejsu:

- moduł zarządzający (ang. Management Module) jest wykorzystywany do kompilacji reguł utworzonych przy pomocy interfejsu GUI. Moduł zamienia reguły na format zrozumiały przez zaporę sieciową. Następnie przenosi je do jednego lub wielu systemów zapór i zarządza dziennikami zdarzeń zapor, obiektami użytymi przy konfiguracji i bazami danych zawierającymi użytkowników wymagających uwierzytelniania w zaporze. Kolejnym składnikiem modułu zarządzającego jest urządzenie certyfikacji. Obsługuje on certyfikaty wydane systemom uwierzytelnionym i działa na głównym komputerze zarządzającym. Takie rozwiązanie ma tę zaletę, że zapewnia centralizację narzędzi zarządzających dla wielu zapór sieciowych jednocześnie. Wszystkie inne komputery polegają na uwierzytelnianiu serwera głównego.

- Indywidualne moduły wykonawcze (ang. Enforcement Modules) odpowiadają za wdrożenie zestawu reguł w formie filtrów pakietów. Innymi słowami, moduły te są tym, co zwykle nazywamy właściwą zaporą sieciową. Automatycznie „utwardzają” system operacyjny podczas instalacji i pozostawiają administratorom do obsługi tylko niektóre funkcje, takie jak wyłączenie usług w `/etc/inetd.conf`. Zestaw reguł jest tworzony przez aplikację, która pracuje zwykle na komputerze klienta. Administrator może wykorzystać interfejs GUI do określenia zestawu reguł, ale wszelkie informacje będą zarządzane z modułu zarządzającego.

Rozdzielenie interfejsu GUI i modułów zarządzających

Oczywiście obydwa moduły, wraz z interfejsem GUI, mogą pracować na jednym komputerze – nie jest to jednak rozwiązanie zalecane ze względu na znaczny spadek wydajności. Rejestrowanie ruchu w sieci znacznie obciąża komputer, a ze względów bezpieczeństwa moduł zarządzający powinien pracować



Rysunek 1. Typowy przykład konfiguracji niewielkiej sieci, ukazujący oddzielenie strefy DMZ od reszty sieci.

na osobnym, dedykowanym komputerze. Umożliwia to zapewnienie dodatkowej ochrony przed atakami (wszystkie kluczowe z punktu widzenia bezpieczeństwa komputery powinny pracować w dedykowanej sieci).

W środowiskach produkcyjnych wiele przedsiębiorstw woli uruchamiać obydwa moduły na jednym komputerze, chociaż interfejs administracyjny GUI zwykle działa na stacji roboczej administratora. Niestety stacja robocza wymaga systemu Windows lub Solarisa, gdyż interfejs administracyjny nie jest dostępny dla platformy linuxowej.

Jedyną dystrybucją Linuksa obsługującą oficjalnie moduł zaporę sieciową Checkpoint jest Red Hat. Eksperci potrafią uruchomić zaporę także na Debianie, mimo że już sama instalacja jest sporym wyzwaniem (ze względu na mocne zorientowanie na Red Hat-a). Na listach dyskusyjnych zajmujących się CheckPointem [1] pojawiają się głosy, że jego instalacja w systemach SuSE, Mandrake i Slackware jest możliwa, ale w pewnych przypadkach należy zastosować wersje specjalne jądra systemu. Zostawmy jednak te warianty ekspertom, gdyż producent zaporę nie będzie wspierał takich instalacji.

Pulpit interfejsu GUI produktu firmy CheckPoint został domyślnie podzielony na cztery części (jak pokazano na Rysunku 2). Oczywiście możemy w dowolnej chwili dostosować widok do własnych, indywidualnych potrzeb. Po lewej stronie znajduje się lista obiektów z zaporami, komputerów, usług i wszystkiego innego, do czego administrator mógłby utworzyć regułę dla zapor. Na szczęście CheckPoint dostarcza wiele przykładów definicji, nawet dla tak rzadko spotykanych usług jak Gopher, odciążając tym samym admini-

stratorów z części ich ciężkiej pracy – umożliwia im to skoncentrowanie się na tworzeniu nowych obiektów.

Edytor zasad

Edytor zasad (prawa górna część okna programu) zajmuje większą część ekranu. Edytor posiada zakładkę zawierającą reguły, zakładkę tłumacza adresów sieciowych (NAT) oraz inne – takie jak np. VPN Manager (w zależności od wykupionej licencji). Po pierwszym uruchomieniu programu nie mamy utworzonych żadnych reguł dla zapor sieciowej. Zatem pierw-

szym krokiem jest stworzenie bramy, która będzie później reprezentować nam zaporę sieciową. Możemy skorzystać z kreatora lub stworzyć ją ręcznie. Musimy pamiętać o dodaniu tego modułu do SIC (ang. bezpieczna komunikacja wewnętrzna).

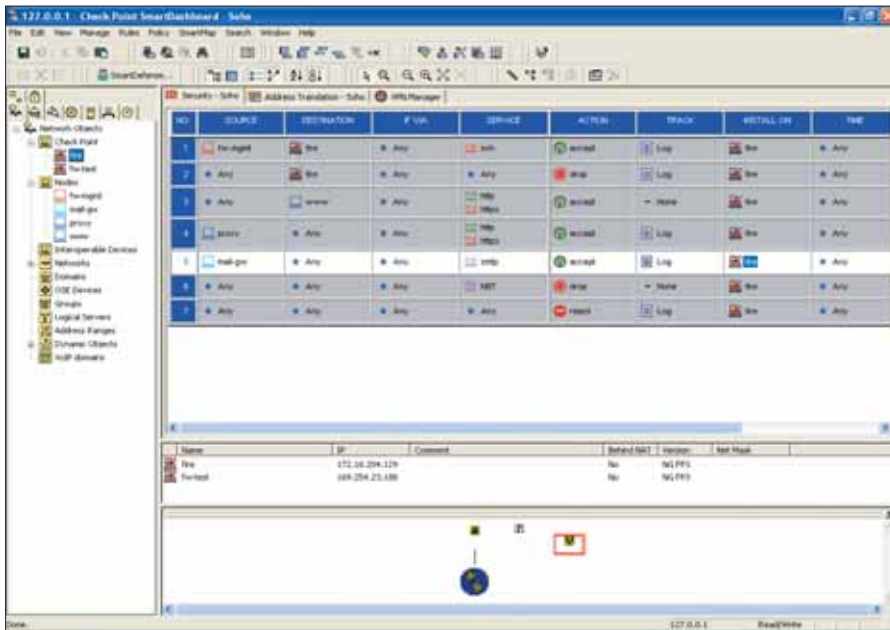
Obiekty zaporę obsługują wewnętrznie reguły zapobiegające fałszowaniu pakietów, co jest dodatkowym zabezpieczeniem samej zapor. Sprawiają one, że potencjalni agresorzy nie mogą uzyskać dostępu do adresów w sieci wewnętrznej. Każdy interfejs sieciowy nie posiadający zabezpieczenia przeciwko fałszowaniu pakietów spowoduje wyświetlenie ostrzeżenia, jeżeli oczywiście włączymy ten zestaw reguł. W tym samym czasie wygenerowany zostanie wpis w dzienniku systemu.

Po wykonaniu opisanych czynności jesteśmy gotowi do tworzenia reguł wymaganych obiektów.

Podstawowa reguła składa się z następujących elementów:

- źródło;
- przeznaczenie;
- usługa wykorzystywana do komunikacji;
- ograniczenie reguły, jeżeli połączenie wykorzystuje określoną sieć VPN;
- narzędzie rejestrujące, jeżeli reguła zostanie spełniona;
- wybór zapor, do której zastosować regułę – umożliwia to zdefiniowanie zestawów reguł dla wielu zapór sieciowych;
- okres czasu obowiązywania reguły;
- komentarz opisujący, czego dana reguła dotyczy.

Po tych elementach następuje lista obiektów, która umożliwia np. bezpośredni podgląd adresów IP. Funkcja ta może okazać się nie- zwykle przydatna przy dużej liczbie obiek-



Rysunek 2. Przykład zestawu reguł utworzonego przy pomocy CheckPoint SmartDashboard.

tów, gdy konieczny jest przegląd użytych adresów IP na określonym obszarze sieci.

Podgląd sieci

Ostatnim panelem jest Visual Policy Editor – graficzny edytor zasad, umożliwiający generalny przegląd komputerów i infrastruktury sieci. Jeżeli mamy właśnie podłączyć większą liczbę komputerów do sieci VPN, będzie to z pewnością cechą programu, którą bardzo polubimy. Ponadto edytor umożliwia administratorom zablżyście przed szefem, gdy ten chce natychmiast zobaczyć schemat pracującej obecnie sieci. Możemy eksportować schemat sieci do pliku graficznego lub zapisać go w formacie Microsoft Visio. Dzięki temu można łatwo tworzyć do archiwum aktualne mapy naszej sieci, co jest przydatne, gdy zarządzaniem zaporami zajmuje się zespół pracowników. Niestety opisane dodatki podlegają dodatkowej opłacie.

Tworzenie pojedynczych reguł jest bardzo proste. Wystarczy, że przeciągniemy i upuścimy wymagane obiekty do określonych pól. Do przenoszenia reguł możemy użyć myszy – opcja ta nie była dostępna we wcześniejszych wersjach programu. Prosta struktura interfejsu znacznie upraszcza pracę wielu administratorom o niewielkim doświadczeniu w tej dziedzinie. Jednakże mnogość dostępnych opcji, których część nie jest na pierwszy rzut oka oczywista, stwarza problem, nad którym producent powinien jeszcze popracować.

Możliwość automatycznego włączenia translacji adresów sieciowych (NAT) przy pomocy zapory jest bardzo pomocna i pozwala

zaoszczędzić wiele pracy. Zawsze jednak możemy ręcznie ustawić NAT. W zależności od tego, jaki sposób wybierzemy, możemy być zmuszeni do ustawienia przekierowania zapory tak, aby odzwierciedlała środowisko NAT (był to warunek konieczny w poprzedniej wersji programu). Przykładowo będziemy musieli zdefiniować przekierowanie z zewnętrznego, oficjalnego adresu IP, za którym będzie ukrywał się komputer użytkownika, do wewnętrznego adresu IP.

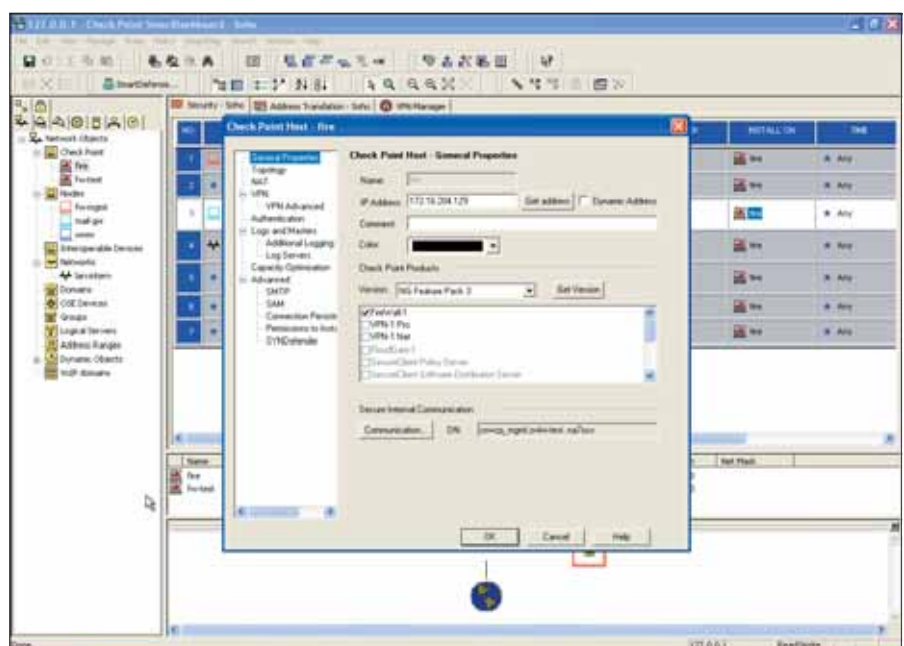
Wersje Post HotFix 3 CheckPoint potrafią monitorować ruch w sieci pod kątem nieautoryzowanego zachowania, takiego jak ataki ty-

pu Ping of Death czy nieprawidłowe polecenia SMTP. Jest to cecha o nazwie SmartDefense, współpracująca z innymi produktami oznaczonymi jako Smart. Warto uważnie śledzić dalszy rozwój aplikacji tego typu. Zakładając, że stworzyliśmy już poprawnie działającą konfigurację zapory, możemy jeszcze bardziej wzmocnić naszą linię obrony, gdyż nasza zapora pracuje nie tylko na poziomie pakietów, ale także może analizować ruch w sieci, podobnie do serwera IDS.

Sztuka tłumaczenia

Po zakończeniu tworzenia zestawów reguł i przygotowaniu do ich instalacji, należy je przenieść do zarządzającej stacji roboczej w formacie zrozumiałym dla zapory sieciowej. W żargonie CheckPoint takie działanie nazywa się kompilowaniem. Plik tekstowy z poszczególnymi regułami zapisanymi w jawnej postaci jest zamieniany na skrypt języka INSPECT, który w rezultacie zostaje zamieniony na kod INSPECT i zainstalowany w zaporach sieciowych. INSPECT jest językiem skryptowym, który firma CheckPoint stworzyła specjalnie na potrzeby własnych produktów. Dzięki temu specjaliści mogą wykonywać w nich zmiany ręcznie.

Jako że zmiany ręczne nie wpływają na obiekty wyświetlane w interfejsie GUI (odbywa się to podczas kompilacji), zalecamy zatem pozostawienie takich zmian prawdziwym ekspertom. Istnieje po prostu zbyt wielkie ryzyko, że zmodyfikowany zestaw reguł będzie niespójny, może się tak zdarzyć szczególnie przy braku odpowiedniej wiedzy.



Rysunek 3. Obiekty zapory CheckPointa posiadają duże możliwości konfiguracji.

CheckPoint korzysta z opracowanej przez siebie bezpiecznej komunikacji wewnętrznej (SIC) i przy jej pomocy przesyła zestaw reguł do modułów wykonawczych. Opiera się to na podobnej zasadzie jak w przypadku protokołów SSL/TLS, gdzie do potwierdzenia identyfikacji nadawcy/odbiorcy wykorzystywane są klucze publiczne (zapewnienie integralności danych i kodowania ruchu w sieci). Tak więc poufność i bezpieczeństwo danych są w pełni gwarantowane.

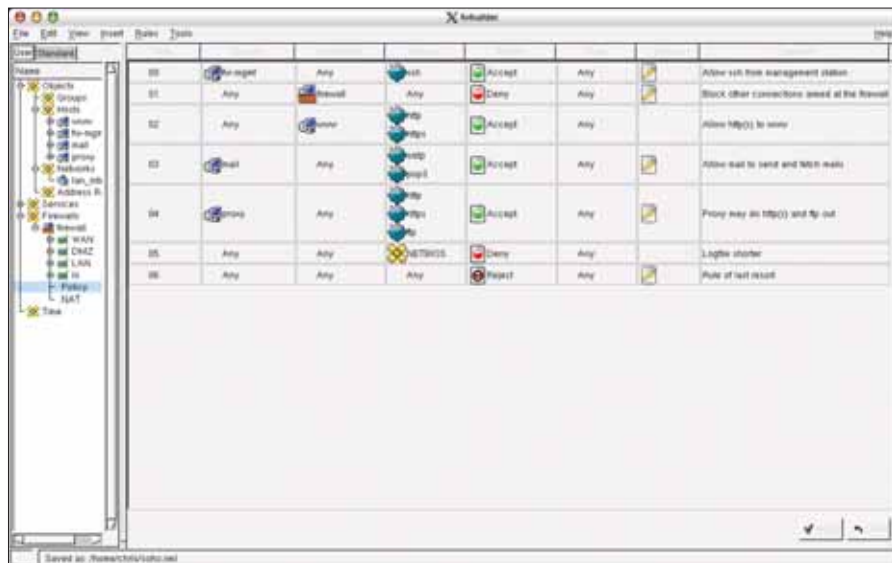
CheckPoint Firewall 1 NG jest wyjątkowo użyteczny dla osób, dla których hasło „satisfakcja gwarantowana” jest niezbędnym czynnikiem decydującym o wyborze produktu. Polecamy ten program szczególnie tym, którzy zamierzają skorzystać z funkcjonalności sieci VPN, zapewniającej narzędzia centralnego zarządzania systemem. Jeżeli interesuje nas głównie łatwa dostępność, nie mamy już zbyt dużego wyboru. Filtry pakietów użyte w systemach typu Open Source mają pewne ograniczenia, np. związane z synchronizacją tabeli stanów.

Młodszy brat

Jeżeli pracowałeś wcześniej w programie CheckPoint, poczujesz się jak u siebie w domu uruchamiając program Firewall Builder [3] – program do złudzenia przypomina wyglądem SmartDashboard. Również tutaj zastosowano zasadę rozdzielania interfejsu GUI od właściwej zapory sieciowej.

Firewall Builder umożliwia zarządzanie czterema technologiami filtrów pakietów:

- IPTables [4] (Kernel Linuksa, wersja 2.4),
- IP Filter [5] (FreeBSD i NetBSD, zastosowany w różnych odmianach Uniksa),
- pf [6] (OpenBSD),
- Cisco Pix.



Rysunek 4. Zestaw reguł utworzony przez Firewall Builder, jako przykładowe rozwiązanie dla sieci z rysunku 1.

Cisco Pix nie posiada standardowo interfejsu do zarządzania. Jeżeli chcemy używać Firewall Buildera do pracy na tej platformie systemowej, będziemy zmuszeni do zakupu niezbędnego modułu dodatkowego w cenie około 500 USD [7], co jest ceną konkurencyjną w porównaniu z innymi produktami na rynku.

Modułowość oprogramowania jest bardzo interesującą cechą, szczególnie gdy będziemy zarządzali centralnie zaporami różnych producentów lub gdy administrator z doświadczeniem w programie CheckPoint musi przesiąść się na inną platformę.

Firewall Builder, poza obsługą wielu filtrów pakietów, może być uruchamiany na różnych odmianach systemu Unix. W momencie pisania tego artykułu istnieją już pakiety dla różnych dystrybucji Linuksa:

FreeBSD, OpenBSD, a nawet dla systemu MacOS X.

Reguły dla wielu zapor sieciowych

Podobnie jak CheckPoint, Firewall Builder widzi różnicę pomiędzy komputerem z pracującym interfejsem GUI, a właściwą zaporą. Jednak w przeciwieństwie do niego, do tworzenia zestawów reguł korzysta on z systemu konfiguracji, aby następnie przesłać je do zapor sieciowych. Nie wykorzystuje mechanizmu przesyłania *bepoke* (używanego przez CheckPoint), ale zawsze możemy użyć w tym celu SSH. Na każdej zaporze instalowany jest niezależny demon, co ma swoje odzwierciedlenie w interfejsie GUI. Firewall Builder, tak jak jego starszy brat, wykorzystuje do komunikacji między

Stateful Inspection przeciwko State Tables

Listy i grupy dyskusyjne są bez ustanku zasympowane pytaniami o różnicę pomiędzy Stateful Firewall i Stateful Inspection firmy CheckPoint – nie jest to wcale dziwne, gdyż sama nazwa faktycznie może wprowadzać w błąd. Wszystkie filtry pakietów, o których mówiliśmy w tym artykule, umożliwiają uproszczenie zestawu reguł przy pomocy tzw. tabeli stanów. Daje to zwiększenie wydajności systemu filtrów pakietów przy jednoczesnym zwiększeniu bezpieczeństwa.

Stateful Firewall

Technologia opiera się na prostej zasadzie. Połączenie TCP rozpoczyna się zawsze pakietem SYN. W celu sprawdzenia, czy dany rodzaj komunikacji jest dozwolony, filtr pa-

kietów używa reguł. Jeżeli komunikacja zostanie dopuszczona, następuje ustanowienie połączenia z klientem i wprowadzenie tego połączenia do tabeli stanów. Tabele przechowywane są w pamięci jądra, co umożliwia wyjątkowo szybki dostęp. Filtr pakietów może następnie wykorzystać te dane do kontrolowania pakietów przychodzących (tam, gdzie flaga SYN nie jest ustawiona) i ustalić, czy należą one do połączenia autoryzowanego. Innymi słowy, nie ma potrzeby stosowania wszystkich reguł – komunikacja może być autoryzowana bezpośrednio.

Takie ograniczenie liczby sprawdzanych reguł może oznaczać zwiększenie wydajności systemu. Dzięki temu system rozpoznaje są-

siadnie pakiety, np. należące do połączenia FTP, które nie są bezpośrednio związane z połączeniem ustalonym wcześniej. System Linux wykorzystuje do tego celu tzw. Helper. Na pewno zwiększa to bezpieczeństwo systemu, gdyż w innym przypadku nie byłoby alternatywy otwierania wielu portów. Teraz możemy tego unikać (patrz także dyskusja pod adresem [9]).

Stateful Inspection

CheckPoint wykorzystuje tę samą technologię [10] i ponadto analizuje obciążenie pakietu (tak jak program proxy) przy użyciu specjalnych skryptów – umożliwia to wydajniejsze monitorowanie komunikacji.

komputerami klucz publiczny.

Obecna wersja zawiera program administratora zapory wraz ze skryptem powłoki o nazwie *fwb_install*. Skrypt wykorzystuje do przesyłania zestawów reguł do zapór klucz publiczny protokół SSH. Po określeniu żądanych zestawów reguł możemy wybrać polecenie Rules/Install i przesłać w ten sposób reguły do zapory.

Interfejs GUI wywołuje *fwb_install* w tle i wyświetla w nowym oknie dialogowym rezultat wykonanego polecenia. Aby uniknąć ciągłego wpisywania długiego hasła, program odwołuje się w pierwszej kolejności do agenta SSH – tam przechowywane są żądane informacje przez określony okresu czasu (lub do momentu wylogowania z systemu).

Kieszonkowy, dzięki XML

Administrator może zdefiniować liczbę opcji przesyłania dla skryptu, np. gdzie przechowywać plik z zestawem reguł dla systemu plików zapory i czy stworzyć dodatkową kopię zapasową pliku XML, użytego do kompilacji zestawu reguł w tym samym miejscu. Istnieje

także opcja zmiany konta użytkownika przy pomocy skryptu w taki sposób, aby logował się on w pierwszej kolejności na zdalnym komputerze.

Można w ten sposób uniknąć logowania do zapory jako użytkownik root. W ten sposób zwiększa się poziom bezpieczeństwa systemu. Jednak należy pamiętać, że zestaw reguł jest przenoszony i włączany przez skrypt, co wymaga uprawnień root-a. Jeżeli korzystamy z konta użytkownika o mniejszych uprawnieniach, upewnijmy się, czy nasze konto korzysta z sudo.

Podczas pierwszego uruchomienia Firewall Builder nie będzie miał ustanowionych żadnych reguł. Zawiera jednak definicje typowych protokołów TCP, UDP i ICMP oraz przestrzenie adresowe sieci prywatnych (LAN). Korzystając z kreatora możemy dodać komputery podłączone do sieci. Kreator umożliwia dostęp do pliku hosta, przeprowadzi transfer strefy DNS, użyje SNOP lub po prostu sprawdzi naszą sieć i sam pobierze niezbędne informacje.

Podczas testowania tej funkcji, program ge-

nerował błąd przy próbie odnalezienia DNS. Jeżeli użyjemy SNOP, program dodatkowo uzyska dane kontaktowe oraz miejsce i opis obiektu, który nas interesuje. Dzięki temu administrator uzyskuje niezbędne informacje o sieci.

Sobowtór GUI – przypadek?

Kreator zawarty w programie jest szczególnie pomocny dla administratorów z mniejszym doświadczeniem, którzy muszą stworzyć początkowy zestaw reguł. Kreator pozwala na ustawienie DMZ i przygotowanie konfiguracji początkowej, niezbędnej do stworzenia prostego, ale wydajnego zestawu reguł. Jest to solidny fundament do tworzenia bardziej zaawansowanych reguł. Oczywiście zawsze możemy tworzyć niezbędne obiekty ręcznie lub korzystając z innych pomocników.

Interfejs użytkownika przypomina Check-Pointa zawiera pola:

- źródło;
- przeznaczenie;
- usługa wykorzystywana do komunikacji;
- polecenie, które należy wykonać, jeżeli reguła zostanie spełniona (potwierdzenie, od-

Listing 1. Użycie zestawów reguł.

```

01 [...]
02
03 echo 1 >
   /proc/sys/net/ipv4/conf/all/rp
   _filter
04 echo 1 >
   /proc/sys/net/ipv4/conf/all/log
   _martians
05 echo 1 >
   /proc/sys/net/ipv4/icmp_echo_ignore
   _broadcasts
06 echo 0 >
   /proc/sys/net/ipv4/icmp_echo_ignore
   _all
07 echo 1 >
   /proc/sys/net/ipv4/icmp_ignore_bogus
   _error_responses
08 echo 30 >
   /proc/sys/net/ipv4/tcp_fin_timeout
09 echo 1800 >
   /proc/sys/net/ipv4/tcp_keepalive
   _intvl
10 echo 1 >
   /proc/sys/net/ipv4/tcp_syncookies
11 [...]
12
13 # Rule 0(NAT)
14 $IPTABLES -t nat -A POSTROUTING -o
   eth2 -s 192.168.0.0/24 -j MASQUERADE
15 [...]
16
17 $IPTABLES -A INPUT -m state --state
   ESTABLISHED,RELATED -j ACCEPT
18 $IPTABLES -A OUTPUT -m state --state
   ESTABLISHED,RELATED -j ACCEPT
19 $IPTABLES -A FORWARD -m state --
   state ESTABLISHED,RELATED -j ACCEPT
20 [...]
21
22 # Rule 0(eth2)
23 # anti-spoofing rule
24 $IPTABLES -N eth2_In_RULE_0
25 $IPTABLES -A INPUT -i eth2 -s
   $interface_eth2 -j eth2_In_RULE_0
26 $IPTABLES -A INPUT -i eth2 -s
   192.168.1.1 -j eth2_In_RULE_0
27 $IPTABLES -A INPUT -i eth2 -s
   192.168.0.1 -j eth2_In_RULE_0
28 $IPTABLES -A INPUT -i eth2 -s
   192.168.0.0/24 -j eth2_In_RULE_0
29 $IPTABLES -A INPUT -i eth2 -s
   192.168.1.2 -j eth2_In_RULE_0
30 $IPTABLES -A FORWARD -i eth2 -s
   $interface_eth2 -j eth2_In_RULE_0
31 $IPTABLES -A FORWARD -i eth2 -s
   192.168.1.1 -j eth2_In_RULE_0
32 $IPTABLES -A FORWARD -i eth2 -s
   192.168.0.1 -j eth2_In_RULE_0
33 $IPTABLES -A FORWARD -i eth2 -s
   192.168.0.0/24 -j eth2_In_RULE_0
34 $IPTABLES -A FORWARD -i eth2 -s
   192.168.1.2 -j eth2_In_RULE_0
35 $IPTABLES -A eth2_In_RULE_0 -m limit
   --limit 10/second -j LOG --log-level
   info
36 --log-prefix "RULE 0 -- DENY "
37 $IPTABLES -A eth2_In_RULE_0 -j DROP
38 [...]
39
40 # Rule 0(global)
41 # allow ssh from Management to the
   firewall
42 $IPTABLES -N RULE_0
43 $IPTABLES -A INPUT -p tcp -s
   192.168.0.2 -d $interface_eth2 --
   destination-port 22 -m state
44 --state NEW -j RULE_0
45 $IPTABLES -A INPUT -p tcp -s
   192.168.0.2 -d 192.168.1.1 --
   destination-port 22 -m state
46 --state NEW -j RULE_0
47 $IPTABLES -A INPUT -p tcp -s
   192.168.0.2 -d 192.168.0.1 --
   destination-port 22 -m state
48 --state NEW -j RULE_0
49 $IPTABLES -A RULE_0 -m limit --limit
   10/second -j LOG --log-level info
   --log-prefix
50 "RULE 0 -- ACCEPT "
51 $IPTABLES -A RULE_0 -j ACCEPT

```

- mowa, blokada, rejestracja);
- okres czasu obowiązywania reguły;
- opcje rejestrowania, jeżeli reguła zostanie spełniona;
- komentarz mówiący nam o tym, czego dana reguła dotyczy.

Poza tymi ogólnymi zasadami każdy interfejs sieci zapór może posiadać własne reguły określające kierunek przepływu ruchu w sieci (przychodzący, wychodzący lub w obie strony). Program CheckPoint wykorzystywał dawniej tę funkcję, ale została ona usunięta w wersji 5.

W przypadku Firewall Buildera opcja ta może okazać się pułapką, gdyż dotyczy ona wyłącznie interfejsu zapory w jednej sieci, a nie sieci poza tym interfejsem. Z drugiej strony, dzięki niej możemy symulować reguły chroniące przed fałszowaniem pakietów, które CheckPoint obsługuje całkiem sprawnie.

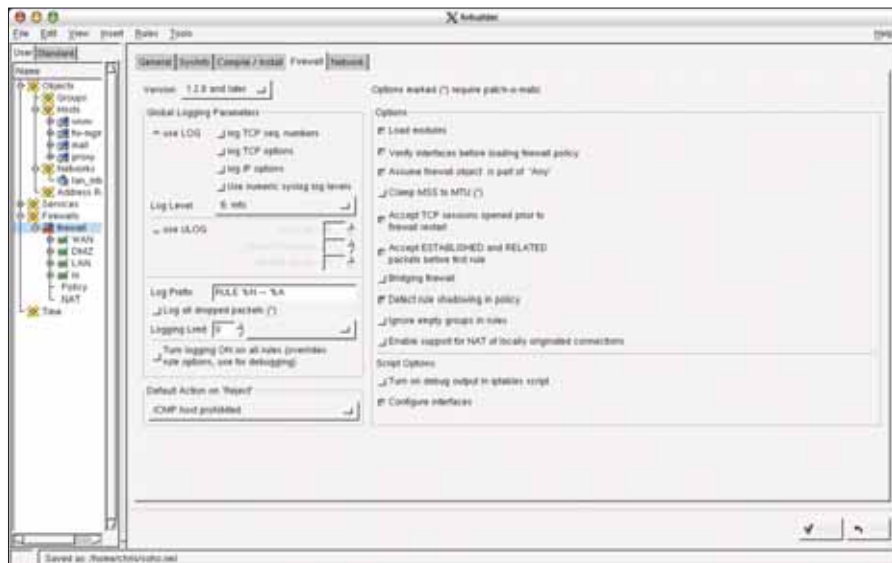
Firewall Builder obsługuje także translację adresów w formie obszaru roboczego dla translatora adresów sieciowych (NAT), tzn. zamianę adresów IP zapory sieciowej na rutowalne adresy internetowe. Maskowanie na sposób linuxowski, czyli ukrywanie całej sieci za rutowalnym adresem IP, jest tutaj wyjątkiem. W przeciwieństwie do rozwiązań zastosowanych w programie CheckPoint, takie wyjątki wymagają rutowania ręcznego (dostęp do komputera wewnątrz sieci ma być rutowany przez zewnętrzny adres IP) – program nie zapewnia automatyzacji tego zadania.

Przeciąganie i upuszczanie reguł i obiektów

Oba programy umożliwiają wstawianie i przenoszenie reguł przy pomocy przeciągania i upuszczania. Zmiany jednego obiektu w zestawie reguł pociągają za sobą natychmiastową zmianę w stosunku do innych reguł. Główne różnice znajdziemy dopiero po głębszej analizie. Mimo że Firewall Builder nie zapewnia takiego bogactwa dostępnych funkcji, wielką zaletą tego darmowego rozwiązania jest niezależność określonej warstwy filtrów.

Nasze rozwiązanie na podstawie przykładowej sieci pokazano na Rysunku 4. Reguły 0 i 1 określają, że *fw-mgmt* jest jedynym komputerem, który korzysta z SSH przy dostępie do zapory i zajmuje się rejestrowaniem tego dostępu. Reguła 2 umożliwia dostęp protokołów HTTP i HTTPS do serwera sieciowego w strefie DMZ. Logowanie nie jest tutaj wymagane, gdyż serwer sieciowy posiada własne pliki dziennika.

Reguły 3 i 4 umożliwiają komunikację z In-



Rysunek 5. Firewall Builder umożliwia bardzo szczegółową konfigurację różnych opcji zapory sieciowej.

ternetem serwerom pocztowym i proxy sieci wewnętrznej. Reguła 5 służy wyłącznie do oczyszczania plików dziennika podczas emisji NetBIOS wysyłanych do sieci lokalnej przez okna dialogowe Windows.

Ostatnia reguła służy do blokowania dostępu, który nie został nikomu innemu przydzielony przy pomocy reguł. Przeciwnie do dwóch poprzednich reguł ograniczających dostęp, ta reguła odrzuca połączenia (ale nie zamyka ich!). Umożliwia to przywrócenie połączenia przy pomocy flagi RESET. Gdyby zamknięto połączenie, nie byłoby żadnej odpowiedzi z nadchodzących pakietów i prowadziłoby to do przekroczenia czasu oczekiwania.

Prostota to podstawa!

Oczywiście reguły te mogą być następnie rozwijane, aby umożliwiły dostęp wyłącznie pakietom ICMP przekazywanym z rutera znajdującego się przed zaporą (przechwytywanie potencjalnych ataków z zewnątrz). Jest to zwykły kompromis pomiędzy ryzykiem a wartością.

Trzymanie się zasady – im prościej tym lepiej, podczas opracowywania zestawu reguł, jest bardzo zalecane. Dlaczego? Każda reguła nakazująca sprawdzanie każdego przychodzącego pakietu będzie miała znaczący wpływ na wydajność całego systemu. Ponadto zestaw reguł powinien być zrozumiały dla administratora. Nawet najlepsza reguła jest bezużyteczna, jeżeli nikt nie potrafi jej zastosować.

Firewall Builder przechowuje zestaw reguł oraz obiektów w prostym pliku XML. Kompilatory dostarczane z pakietem rozdzielają plik XML, wykorzystując go do wy-

generowania instrukcji dla programu w zaporce docelowej. Wykorzystanie XML jako formatu pośredniego umożliwia administratorowi wejście przez tylne drzwi i kontrolę procesów systemu pracujących w tle.

Ostateczny produkt nie kryje faktu, że program korzysta po prostu z wysoce zautomatyzowanych skryptów dla poszczególnych platform – oszczędza w ten sposób wiele wysiłku administratorom, którzy nie są już zmuszani do ręcznego tworzenia skryptów. Wszystkie skrypty powłoki wykazują dużą inteligencję (zawierają instrukcje dla filtrów pakietów). Skrypty obliczają także adresy dynamiczne, umożliwiając odzwierciedlenie ich w konfiguracji programu.

Konfiguracja zawiera listę czynności, o których zwykle zapominamy wykonując tą samą procedurę ręcznie (np. zasady zabezpieczające przed fałszowaniem pakietów, które uniemożliwiają potencjalnym agresorom dostęp do sieci wewnętrznej LAN przy pomocy fikcyjnego adresu IP z zewnątrz).

Zrozumiały zestaw reguł

Proces tworzy wyjątkowo wyczerpujący zestaw reguł, który korzysta zarówno z tabel stanów, jak i wielu skrótów stosowanych w IPTables. Reguły wielokrotnie to tylko jeden z przykładów. Umożliwiają one sytuację, w której jedną regułę można zastosować dla wielu usług (nie trzeba tworzyć kolejnych reguł dla każdej z usług). Zakładka Firewall umożliwia określenie, ile z takich rozwiązań jest obecnie używanych. Możemy także ustalić poziom rejestrowania dla filtra pakietu.

Na Listingu 1 pokazano kilka przykładów.

Wszystkie pozycje zostały dodane do zestawu reguł. Mimo to powinniśmy jednak wiedzieć, czego możemy się spodziewać. W najgorszym razie będziemy opracowywać reguły dla funkcji IPTables, które nigdy nie zostały użyte.

Na Listingu 1 widać wiele pomysłów, z których Firewall Builder korzysta przy tworzeniu zestawu reguł w interfejsie GUI. Skrypt, poza wyznaczeniem dynamicznych adresów IP, wykorzystuje system plików /proc, aby modyfikować jądro, a dzięki temu umożliwić przekazywanie adresów IP. Zauważmy, że kontrole

możliwości systemu umożliwiają jądro sprawdzanie, czy adres IP pochodzi z właściwego interfejsu (*rp_filter*) i ustawianie wybiórczych czasów oczekiwania dla określonych połączeń.

Reguły dla NAT jako pierwsze

Zestaw reguł rozpoczyna się ustaleniem translacji adresów. Zapobiega to potencjalnym problemom związanym z współpracą między NAT a zestawem reguł. Następne w kolejności są reguły, które zapewniają połączenia ze stałymi prawami dostępu do sieci – kolejne uproszczenie zestawu reguł. Zapora potrafi automatycznie rozpoznawać potoki danych należące do tego samego połączenia, np. FTP, więc nie musi odwoływać się do żadnych reguł. Reguły dotyczące indywidualnych interfejsów sieciowych (reguły chroniące przed utratą pakietów) określa się wcześniej w regułach globalnych. Mówiąc ściślej, można opuścić ten element procedury, a zamiast tego użyć polecenia *rp_filter*.

Następną częścią zestawu reguł są reguły globalne. Jak pokazano na przykładzie, dla każdej reguły określono łańcuch. Ułatwia to późniejszą rejestrację, mimo że metoda ta wygląda na bardziej skomplikowaną, niż jest w rzeczywistości.

Reguła 0 (globalna) pokazuje nam, że oprogramowanie bardzo ściśle współpracuje z utworzoną przez administratora zaporą. Nawet jeśli administrator będzie chciał użyć klucza SSH po stronie sieci wewnętrznej LAN, funkcja SSH będzie dostępna dla wszystkich interfejsów zapory. Jeżeli chcemy dalej zagłębiać się w prace nad zestawami reguł, powinniśmy poświęcić trochę czasu na odszukanie i zapoznanie się z konfiguracją generowaną dla naszej sieci – z pewnością ułatwi to nam przyszłą pracę.

Specjalności Netfiltera

Jedną z kolejnych interesujących funkcji Firewall Buildera jest obsługa usług klienta. Umożliwia to wykorzystanie wysoce specjalistycznych opcji programu, dostarczanych wraz z pakietem filtrów. Kombinacja Netfiltera i IPTables może mieć do zaoferowania wiele zaawansowanych właściwości filtrów pakietów (w porównaniu z prostymi filtrami, które oferuje standardowe jądro Linuksa). Poprawka dla łańcucha [8] jest jednym z przykładów. Umożliwia nam ona przeszukiwanie strumienia danych pod kątem ściśle określonych łańcuchów (tekstowych) i zastosowanie wobec nich odpowiednich reguł. Zabezpiecza nas przed blokowaniem serwera sieciowego

bezpośrednio z poziomu sieci.

Kolejną interesującą cechą, umożliwiającą rozszerzenie możliwości globalnych reguł zapory, jest definiowanie reguł specjalnych dla każdego z interfejsów zapory w sieci. Wiele opcji zastosowano w zaporze domyślnie, bez konieczności definiowania reguł przez administratora – blokowanie pakietów rutowanych ze źródła to tylko jedna z tych opcji. Porównując funkcjonalność z regułami CheckPointa stwierdzamy jednoznacznie, że darmowy produkt dostarcza nieco więcej funkcji niż CheckPoint (w zależności od platformy zapory).

Podsumowanie

Firewall Builder jest szczególnie interesujący dla administratorów nie wymagających zaawansowanych funkcji, które oferuje produkt komercyjny. Z funkcji tego rodzaju nie mogą korzystać także administratorzy pracujący w systemach niekompatybilnych ze sobą. ■

W pierwszym szeregu: GuardDog i KNetfilter

GuardDog jest programem IPTables opartym na Qt. Jest interesującą propozycją dla użytkowników z niewielkim doświadczeniem w zakresie sieci i protokołów. Umożliwia użytkownikom wybranie usług dla poszczególnych zestawów zdefiniowanych wcześniej stref sieciowych. GuardDog obsługuje większość protokołów sieciowych (szczególnie używanych w środowiskach domowych).

Mimo że taki rodzaj osobistej zapory sieciowej może wydawać się dobrym pomysłem dla komputerów typu desktop, nie sprawdza się on w bardziej zaawansowanych zastosowaniach. Na szczęście program domyślnie blokuje dostęp wszystkim programom, usługom i funkcjom, jeżeli nie określono żadnych reguł zapory – jest to bardzo przydatne dla początkujących użytkowników sieci, którzy inaczej byłiby potencjalnym źródłem nieograniczonego dostępu dla wszystkiego rodzaju agresorów z zewnątrz.

KNetfilter to kolejny program IPTables, który dodatkowo wykorzystuje zasoby TCP-Dump i Map, udostępniając narzędzia nadzoru. Program zapewnia także funkcję QoS (zapewnienie jakości usługi). Do administratora należy zdefiniowanie wszystkich reguł. Program nie zapewnia w tym zakresie żadnej pomocy. Usługi rozróżniane są dzięki źródłom i przeznaczeniom, które administrator powinien znać na pamięć lub mieć ich pełną listę.

Tak więc KNetfilter umożliwia tworzenie wyjątkowo szczegółowych reguł, jako że potrafi scalać informacje z ICMP i stanów. KNetfilter zdecydowanie nie jest narzędziem dla początkujących, gdyż wymaga solidnych podstaw wiedzy praktycznej na temat IPTables. Bardziej zaawansowani administratorzy na pewno skorzystają z możliwości generowania wyjątkowo szczegółowych i dokładnych zestawów reguł, mimo że do nietypowego interfejsu GUI trzeba się najpierw przyzwyczaić.

INFO

- [1] CheckPoint: <http://www.checkpoint.com/products/protect/firewall-1.html>
- [2] FireWall-1 Gurus Mailing List: <http://www.phoneboy.com/staticpages/index.php?page=20030517034933897>
- [3] Firewall Builder: <http://www.fwbuilder.org>
- [4] IPTables: <http://www.netfilter.org>
- [5] IP Filter: <http://coombs.anu.edu.au/~avalon/ip-filter.html>
- [6] pf: <http://www.benedrine.cx/pf.html>
- [7] PIX-Compiler for Firewall Builder: <http://www.netcitadel.com/pix.htm>
- [8] String-Patch for IPTables: <http://www.netfilter.org/documentation/pomlist/pom-extra.html#string>
- [9] Anatomy of a Stateful Firewall: http://www.giac.org/practical/gsec/Lisa_Senner_GSEC.pdf
- [10] Firewall-1 State table: <http://www.spitzner.net/fwtable.html>
- [11] Mailing Lists for Checkpoint: <http://msgs.securepoint.com/fw>

AUTOR

Christian Ney jest administratorem systemu Unix regionalnych linii lotniczych, a w wolnym czasie pracuje nad różnymi projektami typu Open Source.

