Komunikacja TCP/IP, Firewalle, Routery i NAT

łó › ›

Jakiś czas temu zostałem zmuszony do napisania referatu nt. firewalli, routerów, NAT-u i maskarady do szkoły na przedmiot o nazwie Systemy Operacyjne i Sieci Komputerowe. Ponieważ widziałem tutaj również teoretyczny wpis na temat systemu binarnego, postanowiłem wrzucić na bloga mój referat. :) Zapraszam do lektury.

Komunikacja w Internecie, protokoły sieciowe i porty komunikacyjne

Każdy program zainstalowany w komputerze który chce wysyłać lub odbierać dane przez sieć, musi do tego celu wykorzystać jakiś protokół warstwy aplikacji ze stosu TCP/IP. Wiadomość tego protokołu muszą być następnie upakowane w segment TCP lub datagram UDP (w warstwie transportowej), a następnie przesłane dalej, w dół stosu do warstwy internetowej, gdzie następuje podzielenie segmentu na kilka mniejszych części (fragmentacja datagramów IP), oraz upakowanie ich w datagram IP. Dalej dane są jeszcze pakowane w ramkę i zamieniane na odpowiedni sygnał przesyłany przez medium transmisyjne np. skrętkę kategorii 5 w sieci 100Base-TX Ethernet, do przełącznika, routera i dalej przez Internet do celu.

Podczas komunikacji klient/serwer, każdy serwer danego protokół warstwy aplikacji oczekuje na odebranie połączenia TCP lub datagramu IP na danym porcie. W przypadku serwera HTTP, czyli serwera stron WWW, jest to najczęściej port 80 protokołu TCP. Klient może rozpocząć komunikacje na dowolnym porcie i połączyć się z portem TCP 80 serwera. Porty z których klient nawiązuje połączenie to wysokie porty np. 52044.

Po nawiązaniu połączenia TCP, możliwe jest przesyłanie nim danych, do czasu jego zamknięcia. Takie połączenie może trwać bardzo długo. Protokół UDP jest protokołem bezpołączeniowym. Przesyła on tylko datagramy, nie nawiązując żadnych połączeń.

Na serwerach z systemem Unix, aktywne połączenia TCP możemy poznać wydając w terminalu polecenie `netstat -n` . Poniżej, przykładowy wynik tego polecenia:

```
# Text
root@e00:~# netstat -n
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.1:9000          127.0.0.1:60487         TIME_WAIT
tcp        0      0 127.0.0.1:9000          127.0.0.1:60479         TIME_WAIT
tcp        0      0 127.0.0.1:9000          127.0.0.1:60478         TIME_WAIT
tcp        0      0 127.0.0.1:9000          127.0.0.1:60484         TIME_WAIT
tcp        0      0 127.0.0.1:9000          127.0.0.1:60507         TIME_WAIT
tcp        0      0 127.0.0.1:9000          127.0.0.1:60505         TIME_WAIT
tcp        0      0 127.0.0.1:9000          127.0.0.1:60495         TIME_WAIT
tcp        0      0 127.0.0.1:9000          127.0.0.1:60493         TIME_WAIT
tcp        0      0 127.0.0.1:9000          127.0.0.1:60504         TIME_WAIT
tcp        0      0 77.55.240.61:80         78.88.21.237:57766      ESTABLISHED
tcp        0      0 127.0.0.1:9000          127.0.0.1:60485         TIME_WAIT
```

```
tcp        0      0 127.0.0.1:9000          127.0.0.1:60481         TIME_WAIT
tcp        0      0 77.55.240.61:80         78.88.21.237:57765      ESTABLISHED
tcp        0      0 77.55.240.61:22         78.88.21.237:52000      ESTABLISHED
tcp        0      0 127.0.0.1:9000          127.0.0.1:60483         TIME_WAIT
tcp        0      0 127.0.0.1:9000          127.0.0.1:60491         TIME_WAIT
tcp        0      0 127.0.0.1:9000          127.0.0.1:60482         TIME_WAIT
tcp        0      0 127.0.0.1:9000          127.0.0.1:60480         TIME_WAIT
tcp        0      0 127.0.0.1:9000          127.0.0.1:60498         TIME_WAIT
tcp        0      0 127.0.0.1:9000          127.0.0.1:60506         TIME_WAIT
tcp        0      0 127.0.0.1:9000          127.0.0.1:60500         TIME_WAIT
tcp        0      0 77.55.240.61:80         78.88.21.237:57770      ESTABLISHED
tcp        0      0 127.0.0.1:9000          127.0.0.1:60488         TIME_WAIT
tcp        0      0 127.0.0.1:9000          127.0.0.1:60489         TIME_WAIT
tcp        0      0 127.0.0.1:9000          127.0.0.1:60494         TIME_WAIT
tcp        0      0 127.0.0.1:9000          127.0.0.1:60496         TIME_WAIT
tcp        0      0 77.55.240.61:80         78.88.21.237:57764      ESTABLISHED
tcp        0      0 127.0.0.1:9000          127.0.0.1:60492         TIME_WAIT
tcp        0      0 127.0.0.1:9000          127.0.0.1:60499         TIME_WAIT
tcp        0      0 127.0.0.1:9000          127.0.0.1:60501         TIME_WAIT
tcp        0      0 77.55.240.61:80         78.88.21.237:57759      ESTABLISHED
tcp        0      0 127.0.0.1:9000          127.0.0.1:60502         TIME_WAIT
tcp        0      0 127.0.0.1:9000          127.0.0.1:60490         TIME_WAIT
tcp        0      0 77.55.240.61:80         78.88.21.237:57756      ESTABLISHED
tcp        0      0 127.0.0.1:9000          127.0.0.1:60497         TIME_WAIT
tcp        0      0 127.0.0.1:9000          127.0.0.1:60503         TIME_WAIT
tcp        0      0 77.55.240.61:80         78.88.21.237:57749      ESTABLISHED
tcp        0      0 127.0.0.1:9000          127.0.0.1:60486         TIME_WAIT
tcp        0      0 127.0.0.1:9000          127.0.0.1:60477         TIME_WAIT
tcp        0      0 127.0.0.1:9000          127.0.0.1:60508         TIME_WAIT
tcp        0      0 77.55.240.61:80         78.88.21.237:57769      ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  7      [ ]         DGRAM                    162548653 /dev/log
unix  2      [ ]         DGRAM                    163935546
unix  3      [ ]         STREAM     CONNECTED     163732856
unix  3      [ ]         STREAM     CONNECTED     163732855
unix  2      [ ]         DGRAM                    162645686
unix  2      [ ]         DGRAM                    162548926
unix  2      [ ]         DGRAM                    162548862
unix  2      [ ]         DGRAM                    162548685
```

Na powyższym listingu widać że komputer o adresie IP 78.88.21.237 nawiązał 7 połączeń TCP z serwerem o adresie IP 77.55.240.61 na porcie 80 z różnych wysokich portów. Z tego samego komputera, zostało nawiązane także połączenie TCP z serwerem na porcie 22, z portu wysokiego nr. 52000. Na serwerze, na porcie 80 uruchomiłem serwer HTTP - nginx, a na porcie 22 uruchomiony jest serwer SSH - OpenSSH.

Na liście znajduje się również bardzo dużo połączeń lokalnych z adresu IP 127.0.0.1 z różnych wysokich portów, na adres IP 127.0.0.1 na port 9000. Adres 127.0.0.1 to adres loopback, czyli adres wirtualnej karty sieciowej - tak zwanej, pętli zwrotnej. Wszystkie pakiety wysłane na ten adres, zostaną zwrócone do naszego komputera. Do działania tego adresu nie jest potrzebne połączenie z Internetem ani żadną siecią.

Na moim serwerze, na porcie 9000 uruchomiony jest serwer FastCGI PHP. Po wysłaniu żądania HTTP o przesłanie pliku z rozszerzeniem *.php, serwer HTTP przesyła najpierw zawartość tego pliku do lokalnego serwera FastCGI, a do klienta żądającego pliku, przesyłany jest wynik działania programu - serwera FastCGI, czyli przetworzona już strona WWW w formacie HTML.

Ściana ognia

Firewall, czyli zapora ogniowa służy do blokowania różnych typów komunikacji wchodzącej lub wychodzącej do lub z komputera lub sieci lokalnej. Zabezpiecza on komputery przed atakami i włamywaczami z zewnątrz. Przykładowo, wirus Sasser łączył się z losowymi komputerami na portach TCP 445 i 139 (usługa NetBIOS) i powodował przepełnienie bufora, dzięki czemu mógł załadować się na dany komputer i w ten sam sposób rozprzestrzeniać się dalej oraz powodować automatyczne restartowanie się komputera zaraz po jego uruchomieniu.

Firewall może być programowy lub sprzętowy. Najczęściej firewalle blokują komunikację/połączenia przychodzące lub wychodzące na określonych portach danego protokołu warstwy transportowej (TCP lub UDP). Firewalle mogą też blokować pakiety pochodzące z danych adresów IP, a niektóre, potrafią także blokować pakiety zawierająca jakąś ustaloną wartość w nagłówku protokołu warstwy aplikacji, np. żądania HTTP pochodzące z wybranej przeglądarki stron WWW lub przechodzące przez serwer proxy który pozostawił po tym ślad.

Firewalle programowe, na komputerach klienckich zazwyczaj służą do blokowania możliwości przyjmowania połączeń przychodzących na wszystkich portach i w ten sposób uniemożliwiają ich "odebranie" aplikacjom zainstalowanym na komputerze. Podobnie, firewalle programowe konfiguruje się na serwerach. Wtedy zazwyczaj stosuje się jednak wyjątki, pozwalające na odebranie połączeń na określonych portach. Np. na moim serwerze ustawiłem wyjątki, tak aby mógł on odbierać połączenia przychodzące protokołów SSH, HTTP, FTP, SMTP, POP3, IMAP, SMTPS, POP3S, IMAPS. Na tych portach działają uruchomione przeze mnie serwery - SSH - OpenSSH, HTTP - nginx, FTP - ftpd, SMTP i SMTPS - Postfix, POP3, POP3S, IMAP i IMAPS - Dovecot.

"Polityka", która zakłada że firewall zablokuje całą komunikacje z pewnymi wyjątkami nazywana jest Default Deny Policy - Domyślna Polityka Blokowania. Istnieje też druga "polityka" zakładająca że domyślnie, ruch sieciowy na wszystkich portach będzie przekazywany dalej, z wykluczeniem kilku ustalonych wyjątków np. blokowanie tylko ruchu przychodzącego na port 80. Jest ona nazywana Default Allow Policy.

Firewalle sprzętowe, oprócz blokowania ruchu na określonych portach, bardzo często używane są do blokowania całego ruchu z danych adresów IP. Jest to bardzo przydatne podczas ataków DDoS. Atak DDoS polegają na wysyłaniu bardzo dużej ilości zapytań/żądań np. HTTP do serwera bez oczekiwania na odpowiedź, przez dużą ilość komputerów. Powoduje to przeładowanie bufora w serwerze i jego blokadę. Firewalle sprzętowe potrafią wykrywać źródła dużego natężenia nawiązywanych połączeń/wysyłanych żądań i blokować adresy IP z których pochodzi ten ruch

Firewalle sprzętowe potrafią także równoważyć obciążenie. Odbierają ruch przychodzący do jednego adresu IP serwera i przesyłają go do dowolnego serwera znajdującego się w puli. Serwery te świadczą identyczne usługi i komunikacja wygląda tak jakby każdy użytkownik łączył się z tym samym fizycznym serwerem.

Firewalle sprzętowe nie są z reguły podłączane przed pojedynczymi serwerami, ale przed całymi sieciami komputerowymi.

Routery i maskarada

Prawdziwe routery nie mają nic wspólnego z routerami domowymi służącymi do dzielenia łącza. Routery te mają po kilka kart sieciowych (interfejsów) i za pomocą każdego z nich, przyłączane są do innych sieci. Mogą one wymieniać pakiety IP między podłączonymi do nich sieciami. Jeśli dany router nie jest podłączony do sieci do której ma przesłać dany pakiet IP, przesyła go do innego routera który jest następnym "hopem" na najkrótszej trasie do celu. Trasa ta jest zapisana w tablicy routingu, a tablice tą tworzą protokoły routingu.

Domowe routery tak naprawdę składają się z 3 urządzeń zintegrowanych w jedno:

- Czegoś co producenci nazywają routerem, ale z routerem nie ma faktycznie nic wspólnego i bardziej przypomina firewall. Ma on 2 karty sieciowe i jest podłączony z jednej strony do portu WAN/modemu a z drugiej do wbudowanego przełącznika;
- Switcha/Przełącznika Ethernetowego, który pozwala podłączyć wiele komputerów do "routera";
- oraz Access Pointa, podłączonego do przełącznika i pozwalającego przyłączać do sieci, bezprzewodowo komputery wyposażone w karty WiFi.

Czasami routery też zawierają wbudowane modemy, np. ADSL lub DOCSIS, umożliwiający podłączenie go bezpośrednio do kabla dostarczonego przez dostawcę Internetu - ISP.

Moduł "routera" w routerze nie łączy sieci tak jak prawdziwy router, a udaje podłączony do sieci dostawcy komputer z przypisanym pojedynczym publicznym adresem IP i pozwala, komputerom z sieci lokalnej, komunikować się z Internetem, zamieniając ich prywatne adresy na swój adres publiczny, podczas wysyłania oraz odwrotnie, w trakcie odbierania danych. Komputery w sieci lokalnej mają przypisane prywatne adresy IP które mogą się powtarzać w każdej z takich sieci i nie są publicznie trasowane w Internecie. Aby router wiedział, które dane przychodzące ma przesłać do którego z lokalnych komputerów, dla każdego połączenia wychodzącego zmienia port z którego wychodzi to połączenie na inny losowy (wysoki) i zapisuje go w "tabelce". W trakcie odbierania danych, odczytuje port z tabelki i zamienia go na ten z którego połączenie przyszło od prawdziwego celu oraz zmienia IP w pakiecie, na to lokalne. Tak więc, połączenie może być zainicjowane jedynie przez komputer wewnątrz sieci lokalnej. Nie możliwe jest przysłanie pakietu do komputera w sieci lokalnej, z zewnątrz.

Takie zachowanie routera nazywane jest translacją adresów sieciowych (NAT) wiele (adresów wewnętrznych) do jednego (adresu zewnętrznego) lub maskaradą.

Strefa zdemilitaryzowana i przekierowanie portów

Aby rozwiązać problem braku możliwości przyjmowania połączeń, powstało przekierowanie portów oraz strefa zdemilitaryzowana. Na większości domowych routerów, możliwe jest przekierowanie pojedynczych portów z publicznego adresu IP przypisanego do routera, na pojedynczy port, wybranego komputera w sieci lokalnej. DMZ, czyli strefa zdemilitaryzowana, to możliwość przekierowania całego ruchu przychodzącego na publiczny adres routera, na dowolny wybrany adres IP z sieci lokalnej.

Na routerach TP-LINK, aby przekierować port z publicznego adresu routera, wchodzimy w panel administracyjny routera, przechodzimy do zakładki Forwarding -> Virtual Servers, klikamy Add New, podajemy port publiczny, port prywatny, lokalny adres IP komputera, na który ma zostać utworzone przekierowanie, protokół - TCP, UDP lub oba na raz i zatwierdzamy, klikając przycisk Save.

Po przekierowaniu portu, warto zarezerwować sobie, wybrane lokalne IP, tak aby port był zawsze przekierowywany na nasz komputer. W tym celu, wchodzimy w zakładkę DHCP -> Address Reservation, klikamy New, podajemy parę adresów - nasze lokalne IP oraz adres sprzętowy naszej karty sieciowej który nigdy się nie zmienia i jest na stałe zakodowany w pamięci ROM karty i klikamy Save.

Aby przekierować cały ruch na wybrane IP w sieci lokalnej, czyli utworzyć strefę zdemilitaryzowaną, wchodzimy w zakładkę Forwarding -> DMZ, zaznaczamy Enabled i podajemy adres IP naszego lokalnego komputera, na który chcemy przekierować ruch, a następnie klikamy Save.

ę

ó ś ę

„ ó ś ę