

TCPDUMP

Tcpdump jest jednym z popularniejszych analizatorów sieciowych (potocznie nazywanych snifferami) pracujących na platformie uniksowej. Głównym zadaniem tcpdumpa jest nasłuchiwanie ruchu panującego w sieci. Można wykorzystać ten program w dwóch głównych celach takich jak: analiza pakietów, które przepływają przez sieć, oraz w celu przechwytywania informacji wysyłanych przez innych użytkowników. Tcpdumpa można ściągnąć z jego oficjalnej witryny www.tcpdump.org. Do jego działania wymagana jest biblioteka libpcap, którą również możemy ściągnąć z wymienionej wcześniej strony. Ważną informacją jest to, że aby móc korzystać z tego programu należy posiadać w danym systemie konto administratora (root). To tyle tytułem wstępu, teraz zajmiemy się analizą poszczególnych funkcji oraz możliwości tego sniffera.

Zanim przyjrzymy się dokładniej możliwościom jakie oferuje nam tcpdump, należy wspomnieć o metodach zapisywania logów z tego programu, ponieważ przeglądanie ich na bieżąco nie jest wygodne oraz nie pozwana na późniejszą, dokładniejszą analizę pakietów.

Zapisanie logów do pliku poprzez tcpdump wywołujemy w następujący sposób:

```
root# tcpdump -w nazwa_pliku
```

Parametr "-w" (od ang. "write") włącza opcję logowania danych do pliku. Jednak format tych logów, uniemożliwia nam przeglądanie ich w zwykłym edytorze tekstowym, tak więc logicznie myśląc aby przeglądać logi należy użyć tego samego programu, ale z parametrem "r" (od ang. "read"). Polecenie będzie wyglądać następująco:

```
root# tcpdump -r nazwa_pliku
```

Istnieje jeszcze druga metoda zapisania logów z tcpdump różniąca się od pierwszej tym, że dzięki niej otrzymujemy plik tekstowy, zdatny do analizy w dowolnym edytorze tekstu.

Jak łatwo się domyślić, wystarczy, że przekierujemy wyjście polecenia do pliku:

```
root# tcpdump > nazwa_pliku
```

lub

```
root# tcpdump >> nazwa_pliku
```

Pierwsza z wyżej wymienionych metod (ta ze znakiem ">") utworzy plik do którego zapisywane będą logi, lub w przypadku, gdy taki plik już istnieje, jego zawartość zostanie wyzerowana. Drugie z poleceń zapisze nasze logi do danego pliku, dopisując dane na jego końcu, nie kasując poprzednio zapisanych w nim danych.

To tyle na temat zapisywania logów, teraz przejdziemy do opisu opcji, jakie posiada tcpdump. Podstawowym parametrem z jakim możemy wywołać program, jest "-i" (od ang. interface) a następnie po nim podajemy nazwę interfejsu karty na którym ma nasłuchiwać sniffer, przykładowo może to być eth0 (pierwsza karta sieciowa). Nasz sniffer z takim parametrem będzie nasłuchiwać wszystkie pakiety, które przechodzą przez ten interfejs. Kompletne wywołanie będzie wyglądać następująco:

```
root# tcpdump -i eth0
```

Należy wspomnieć także o parametrze "-n", którego zastosowanie jest bardzo pomocne, ponieważ dzięki niemu nie tracimy czasu na konwertowanie adresów IP na nazwy hostów. Dokładniej mówiąc nasz log nie będzie pokazany w postaci:

```
IP 10.0.1.247.43533 > m2.gadugadu.pl.8074 (skonwertowany adres IP
```

do DNS)

lecz w postaci:

```
IP 10.0.1.247.43533 > 217.17.41.85.8074
```

(adresy nie skonwertowany dzięki użyciu parametru "-n")

Kolejnym ważnym parametrem, z którym wywołujemy program to "-v". Pozwala on na dokładniejszą analizę pakietów. Jeśli chcemy uzyskać coraz dokładniejszą analizę, możemy zwiększać parametr pisząc "-vv" lub "-vvv". Przykładowo będzie to wyglądało następująco:

```
root# tcpdump -i eth0 -v
```

Jeśli można wywołać program z dokładną analiza pakietu, to także moż:na spowodować, by wyświetlił się nam, coraz dokładniejszy czas pakietu. Służy do tego parametr "-t" tak jak w poprzednim przykładzie, możemy stosować ten parametr w postaci "-tt", "-ttt" oraz "-tttt" co pozwoli nam na coraz dokładniejsze badanie czasu.

Zajmijmy się teraz wnętrzem pakietu, które można zobaczyć w postaci heksadecymalnej (szesnastkowej) używając parametru "x" przykładowo:

```
root# tcpdump -i eth0 -x
```

rzykładowa zawartość pakietu może wyglądać następująco:

```
0x0000: 4500 003c f74a 4000 4006 3514 0a00 01f7 E..<.J@.@.5.....
0x0010: d911 2955 aa0d 1f8a 34a1 9fe4 8b7c 4119 ..)U....4....|A.
0x0020: 8018 2d40 c3b9 0000 0101 080a 010e 37ca ..-@.....7.
0x0030: 61c1 6a09 0800 0000 0000 0000 a.j.....
```

Analogicznie, aby zobaczyć pakiet w postaci ASCII należy użyć parametry "-X" (duże X)

Tcpdump może oprócz nasłuchiwanie całego interfejsu sieciowego, przejść na tryb nasłuchiwanie poszczególnych portów, protokołów albo ruchu pomiędzy danymi komputerami w sieci itd. Zacznijmy od pierwszej metody. Aby ustawić tcpdump na nasłuchiwanie konkretnego portu należy go uruchomić w następujący sposób:

```
root# tcpdump -i eth0 port 80
```

(zamiast portu 80 możemy wpisać dowolny inny port)

Kolejna rzeczą, którą wymieniłem jest nasłuchiwanie konkretnego protokołu sieciowego:

```
root# tcpdump -i eth0 icmp
```

(zamiast protokołu icmp możemy wpisać dowolny inny)

Ostania czynnością jest ruch pomiędzy dwoma stacjami:

```
root# tcpdump -i eth0 host 10.0.1.247 and \ (10.0.1.248\)
```

Wyżej wymieniony przykład nasłuchuje cały ruch pomiędzy komputerem o numerze IP: 10.0.1.247, a komputerem o adresie 10.0.1.248

Podam jeszcze jeden przykład metody sniffingu za pomocą tcpdump, a później przejdziemy już do analizowania logów. Poniższy przykład pokazuje jak możemy podsłuchiwać dany port na zdefiniowanym komputerze w sieci:

```
root# tcpdump -i eth0 src 10.0.1.248 and port 80
```

To chyba tyle na temat najczęściej stosowanych parametrów z jakimi można wywołać ten program. Jeśli chcesz się dokładnie zapoznać ze wszystkimi parametrami jakie posiada tcpdump, wpisz w konsoli "man tcpdump".

Przechodzimy teraz do analizy logów. Omówienie tutaj dwa przykłady logów. Pierwszym będzie log z użycia programu ping, który korzysta z protokołu diagnostycznego ICMP, drugim zaś log z komunikatora internetowego.

Log1:

```
13:29:50.686670 IP 10.0.1.247 > 212.77.100.101: icmp 64: echo
request seq 3
13:29:50.704619 IP 212.77.100.101 > 10.0.1.247: icmp 64: echo
reply seq 3
```

Interpretacja:

13:29:50.686670 - czas podany w postaci HH:MM:SS:MS

10.0.1.247 - nadawca

212.77.100.101 - odbiorca

> - kierunek w którym płyną pakiety

icmp - protokół z którego korzystamy

request - zapytanie

reply - odpowiedz na zapytanie.

Log2:

```
13:37:46.832958 IP (tos 0x0, ttl 64, id 63716, offset 0, flags
[DF], length: 77) 10.0.1.247.43533 > 217.17.41.85.8074: P [tcp sum
ok] 883011855:883011880(25) ack 2340185593 win 17376
<nop,nop,timestamp 18040479 1643430965>
0x0000: 4500 004d f8e4 4000 4006 3369 0a00 01f7 E..M..@.@.3i....
0x0010: d911 2955 aa0d 1f8a 34a1 b10f 8b7c 65f9 ..)U....4....|e.
0x0020: 8018 43e0 dd07 0000 0101 080a 0113 469f ..C.....F.
0x0030: 61f4 c435 0b00 0000 1100 0000 21e7 7800 a..5.....!.x.
0x0040: 1af9 7801 0800 0000 7465 7374 00 ..x.....test.
```

Interpretacja:

13:37:46.832958 - czas podany w postaci HH:MM:SS:MS

tos - typ usługi

ttl (time to live) - potęga cyfry 2 która określa długość życia pakietu

id - jest to nic innego jak ID pakietu

[DF] - flaga don't fragment (nie fragmentuj)

length: 77 - wielkość pakietu podana w bajtach maksymalna wielkość pakietu w sieciach ethernet wynosi 1500 bajtów

> - kierunek przepływu pakietów

10.0.1.247.43533 - adres nadawcy wraz z portem na którym wysyła

217.17.41.85.8074 - adres odbiorcy wraz z portem na którym odbiera pakiet

883011855:883011880 - numer sekwencyjny

ack - potwierdzenie następnego numeru pakietu danych

win - (window size) wielkość okna podana w bajtach

nop - jest to pusta opcja

timestamp - jak sama nazwa wskazuje, jest to stempel czasu dla pakietu

Pod nagłówkiem widzimy zawartość przesyłanego pakietu w postaci heksadecymalnej.

To wszystko co chciałem wam przedstawić na temat ogólnej obsługi programu tcpdump.

Wszystkie uwagi związane z artykułem proszę kierować na: [itros\(at\)linux.pl](mailto:itros@linux.pl)

Artykuł pochodzi ze strony Słupskiej Grupy Użytkowników Linuksa (<http://linux.slupsk.pl>)