
# TCPDUMP İLE TRAFİK ANALİZİ(SNİFFİNG)

## Tcpdump-I

**Huzeyfe ÖNAL <huzeyfe@lifeoverip.net>**

**4/13/2010**

Bu yazı tcpdump'a ait sık kullanılan parametreleri örnekleriyle birlikte açıklayıp konuya yabancı olanlara tcpdump'a giriş niteliğinde bir belge sunmayı amaçlamaktadır.

## İçerik Tablosu

# Tcpdump

## Tcpdump Nedir?

Tcpdump Linux/UNIX sistemlerde de-fakto paket yakalama ve analiz aracıdır. Tcpdump pcap paket yakalama kütüphanesini(libpcap) kullanır ve ağ arabiriminden geçen paketleri (TCP/IP protokollerini) kaydedip, pcap destekli herhangi bir araç kullanarak kaydedilmiş paketleri okuma işine yarar.

Özellikle ağ üzerinden yakaladığı paketleri pcap formatındaki sniffer araçlarının okuyabileceği formatta kaydetme özelliği, yoğun trafiğe sahip ağlarda sorunsuz paket yakalama becerisi tcpdump'ı ağ güvenliği yöneticilerinin vazgeçilmezi kılmaktadır.

### Windows için Tcpdump

Tcpdump'ın Windows işletim sistemlerindeki eşdeğeri Windump aracıdır. http://www.winpcap.org/ adresinden indirilecek ikili dosyalar sisteme kurularak tcpdump benzeri kullanım imkanı elde edilebilir.

# Tcpdump Kullanımı

Tcpdump klasik Linux/UNIX araçları gibi komut satırından çalışır ve tüm özelliklerini parametre olarak alır. Parametresiz çalıştırıldığında sistemde bulduğu ilk aktif ağ arabirimini dinlemeye alır(root izni varsa*). Tcpdump'ın çeşitli amaçlarla kullanılacak onlarca parametresi vardır ve sıradan bir ağ yöneticisinin bu parametreleri ezberlemesi gereksizdir.

Bu yazı tcpdump'a ait sık kullanılan parametreleri örnekleriyle birlikte açıklayıp konuya yabancı olanlara tcpdump'a giriş niteliğinde bir belge sunmayı amaçlamaktadır.

Tcpdump kullanmaya başlamadan sistem hakkında bilinmesi gereken bir iki husus vardır. Bunlar;

## Promiscious mod

Bir makinenin hedefi kendisi olmayan paketleri alabilmesi için ağ arabiriminin promiscious modda olması gerekir. Tüm snifferlar otomatik olarak ağ arabirimini promiscious moda geçirir ve sniffer durdurulduğunda tekrar arabirimi normal moda döndürür.

Arabirimin prosmic modda olup olmadığı ifconfig komutu çıktısında gözükecektir.

```
# ifconfig
bce0: flags=28902<BROADCAST,PROMISC,SIMPLEX,MULTICAST,PPROMISC> metric 0 mtu 1500

options=1bb<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,JUMBO_MTU,VLAN_HWCSUM,TSO
4>
```

Tcpdump komutu çalıştırıldığında ağ arabirimini otomatik olarak promisc moda geçirir ve tcpdump'ı sonlandırdığınızda yine ağ arabirimini promisc moddan çıkarır.

## Yetki

**Linux/UNIX** altında tcpdump programını kullanabilmek için ya root haklarına sahip olmak lazım ya da tcpdump programının suid olarak çalışması lazım

**Tcpdump**, paketleri kernel'a giriş-çıkış yapmadan yakalar bu sebeple iptables(Linux için) ile yazdığınız kurallar tcpdump'ı etkilemez.

Tcpdump'ın en basit kullanımı parametresiz kullanımdır.

```
# tcpdump
tcpdump: listening on ste0, link-type EN10MB
19:25:39.148459 arp who-has 192.168.123.100 (33:68:9c:b1:fc:bb) tell
192.168.123.100
19:25:39.156276 open.edu.tr.2000 > 88.235.74.210.kpop: P
1769688136:1769688220(84) ack 2940700931 win 17640 (DF) [tos 0x10]
19:25:39.156490 open.edu.tr.2000 > 88.235.74.210.kpop: P 84:136(52) ack 1 win
17640 (DF) [tos 0x10]
19:25:39.165021 88.235.74.210.kpop > open.edu.tr.2000: . ack 0 win 16072 (DF)
19:25:39.183084 88.235.74.210.kpop > open.edu.tr.2000: . ack 136 win 15936 (DF)
19:25:40.148429 arp who-has 192.168.123.100 (33:68:9c:b1:fc:bb) tell
192.168.123.100
19:25:40.157596 open.edu.tr.2206 > bim.open.edu.tr.domain: 55035+ PTR?
100.123.168.192.in-addr.arpa. (46)
19:25:40.247597 bim.open.edu.tr.domain > open.edu.tr.2206: 55035 NXDomain 0/1/0
(123)
```

Tcpdump çıktısı ilk bakışta anlaşılır gelmese de çıktıları oluşturan bileşenler tanındıkça çıktılar da anlaşılır olacaktır.

Aşağıda tcpdump için TCP, UDP ve ICMP protokollerine ait çıktılarının bileşenleri açıklanmıştır.

## Tcpdump TCP Paket Formatı

| Değer | Açıklaması |
|---|---|
| 16:21:24.174 | Zaman Damgası |
| 192.168.60.3 | Kaynak IP Adresi |
| 34720 | Kaynak Port numarası |
| > | Yön Belirteci |
| 10.10.10.3 | Hedef IP Adresi |
| 3389 | Hedef Port Numarası |
| S | TCP Bayrağı (SYN Bayrağı set edilmiş) |
| 2354677536 | TCP başlangıç seri numarası (ISN) |
| 2354677536 | Bir sonraki byte icin beklenen sıra numarası |
| (0) | Bu segmentin içerdiği uygulama verisi hesabı |
| win 5840 | Byte cinsinden Window size. |
| mss 1460 | Maximum Segment Size (MSS) |
| sackOK | Selective acknowledgement |
| (DF) | Paketin DF(Parçalanmaması) özelliğinde olduğunu |

## Tcpdump UDP Paket Formatı

| Değer | Açıklaması |
|---|---|
| 10:20:21.17 | Zaman Damgası |
| 172.27.20.4 | Kaynak IP Adresi |
| 41197 | Source port |
| > | Yön Belirteci |
| 192.168.60.5 | Hedef IP |
| 24 | Destination port |
| udp 300 | Byte cinsinden udp datagram boyutu |

## Tcpdump ICMP Paket Formatı

| Değer | Açıklaması |
|---|---|
| 10:20:04.92 | Zaman Damgası |
| 172.27.20.4 | Kaynak IP Adresi |
| > | Yön Belirteci |

| Değer | Açıklaması |
|---|---|
| 192.168.60.3 | Hedef IP |
| icmp: echo request | ICMP mesaj tipi |

## Sık Kullanılan Parametreler

### Arabirim Seçimi( -i )

Sistemimizde birden fazla arabirim varsa ve biz hangi arabirimini dinlemesini belirtmezsek tcpdump aktif olan ağ arabirimleri arasında numarası en düşük olanını dinlemeye alır, mesela 3 adet aktif Ethernet ağ arabirimimiz var; eth0, eth1, eth2[Linux için geçerlidir,diğer unix çeşitlerinde farklıdır, şeklinde biz bu makinede tcpdump komutunu yalın olarak kullanırsak tcpdump eth0 arabirimini dinlemeye alacaktır.

Eğer ilk arabirimi değilde istediğimiz bir arabirimi dinlemek istiyorsak -i parametresi ile bunu belirtebiliriz

**# tcpdump -i eth2**

komutu ile sistemimizdeki 3.Ethernet kartını dinlemeye alıyoruz.

Sistemde bulunan ve tcpdump tarafından dinlemeye alınabilecek arabirimlerin listesini almak için –D parametresi kullanılabilir.

```
[root@netdos1 ~]# tcpdump –D
1.em0
2.pflog0
3.em1
4.lo0
```

### İsim Çözümleme ( -n )

Eğer tcpdump ile yakalanan paketlerin dns isimlerinin çözülmesi istenmiyorsa -n parametresini kullanılabilir. Özellikle yoğun ağlarda tcpdump her gördüğü ip adresi-isim için dns sorgusu göndermeye çalışıp gelen cevabı bekleyeceği için ciddi yavaşlık hissedilir.

```
Normal kullanım;
# tcpdump
17:18:21.531930 IP huzeyfe.32829 > erhan.telnet: S 3115955894:3115955894(0) win
5840
17:18:21.531980 IP erhan.telnet > huzeyfe.32829: R 0:0(0) ack 3115955895 win 0
```

-n parametresi ile kullanım;

**# tcpdump -n**

```
17:18:53.802776 IP 192.168.0.100.32835 > 192.168.0.1.telnet: S
3148097396:3148097396(0) win 5840
17:18:53.802870 IP 192.168.0.1.telnet > 192.168.0.100.32835: R 0:0(0) ack
3148097397 win 0
```

burada huzeyfe makinesi 192.168.0.100, erhan makinesi 192.168.0.1 IP adresine sahiptir. İsimlerin yanında protocol ve port numaralarınında isimlere çevrimi de istenmiyorsa -nn parametresini kullanılabilir.

**# tcpdump -nn**

yukarıda (-n için)verdiğimiz örnekte -n yerine -nn koyarsanız hem isim hemde port çözümlemesi yapılmayacaktır,yani telnet yerine 23 yazacaktır.

### -Zaman Damgası Gösterimi ( -t )

Eğer tcpdump'ın daha sade bir çıktı vermesini isteniyorsa ekrana bastığı satırların başındaki timestamp(zaman damgası, hangi paketin hangi zaman aralığında yakalandığını belirtir) kısmı iptal edilebilir.

Çıktılarda timestamp[zaman damgası]leri istenmiyorsa -t parametresi kullanılabilir.

#### *Timestamp li çıktı*

**# tcpdump**

```
15:32:13.479577 cc.open.edu.tr.200 > 212.174.108.162.29157: . 68:1528(1460) ack
53 win 20440 (DF) [tos 0x10]

15:32:13.479582 cc.open.edu.tr.200 > 212.174.108.162.29157: P 1528:2456(928)
ack 53 win 20440 (DF) [tos 0x10]
```

#### *Timestamp(Zaman damgası)sız çıktı*

**# tcpdump –t**

```
2.174.108.162.29157 > cc.huzeyfe.net.2000: P 3329:3381(52) ack 11236 win 17520
(DF) [tos 0x20]
cc.huzeyfe.net.2000 > 2.174.108.162.29157: . ack 2289 win 8576 (DF) [tos 0x10]
```

### Yakalanan Paketleri Kaydetme ( -w )

Tcpdump'ın yakaladığı paketleri ekradan değilde sonradan incelemek üzere bir uygun bir şekilde dosyaya yazması istenirse -w parametresi kullanılabilir. Kaydedilen dosya cap uyumlu olduğu için sadece tcpdump ile değil birçok network snifferi tarafından okunup analiz edilebilir.

**# tcpdump -w dosya_ismi**

-r /Kaydedilmiş Paketleri Okuma

-w ile kaydedilen paketler -r parametresi kullanılarak okunabilir.

**# tcpdump -r dosya_ismi**

**Not!** *-w ile herhangi bir dosyaya kaydederken filtreleme yapılabilir. Mesela sadece şu tip paketleri kaydet ya da timestampleri kaydetme gibi, aynı şekilde -r ile paketlerie okurken filtre belirtebiliriz. Bu filtrenin -w ile belirtilen filtre ile aynı olma zorunluluğu yoktur.*

```
# cd /tmp/

# tcpdump -w log icmp

tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes

ctrl c

# tcpdump -r log -nn

reading from file log, link-type EN10MB (Ethernet)
17:31:01.225007 IP 192.168.0.100 > 192.168.0.1: icmp 64: echo request seq 0
17:31:01.225119 IP 192.168.0.1 > 192.168.0.100: icmp 64: echo reply seq 0
17:31:02.224988 IP 192.168.0.100 > 192.168.0.1: icmp 64: echo request seq 1
17:31:02.225111 IP 192.168.0.1 > 192.168.0.100: icmp 64: echo reply seq 1
```

### Yakalanacak Paket Sayısını Belirleme ( -c )

tcpdump'a -c parametresini vererek ne kadar paket yakalayıp duracağını söyleriz.

```
# tcpdump -i eth0 -c 5

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
00:59:01.638353 IP maviyan.net.ssh > 10.0.0.2.1040: P
1010550647:1010550763(116) ack 774164151 win 8576
00:59:01.638783 IP 10.0.0.2.1040 > maviyan.net.ssh: P 1:53(52) ack 116 win 16520
00:59:01.638813 IP maviyan.net.ssh > 10.0.0.2.1040: P 116:232(116) ack 53 win
8576
00:59:01.639662 IP 10.0.0.2.1040 > maviyan.net.ssh: P 53:105(52) ack 232 win
16404
00:59:01.640377 IP maviyan.net.ssh > 10.0.0.2.1040: P 232:380(148) ack 105 win
8576
5 packets captured
5 packets received by filter
0 packets dropped by kernel
```

Tcpdump, -c sayı ile belirtilen değer kadar paket yakaladıktan sonra çalışmasını durduracaktır.

### Yakalanacak Paket Boyutunu Belirleme ( -s )

-s parametresi ile yakalancak paketlerin boyutunu byte olarak belirtilebilir.

#tcpdump –s 1500 gibi. Öntanımlı olarak 96 byte kaydetmektedir.

```
# tcpdump -i eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
```

### Detaylı Loglama (-v)

-v parametresi ile tcpump'dan biraz daha detaylı loglama yapmasını istenebilir. Mesela bu parametre ile tcpdump çıktılarını TTL ve ID değerleri ile birlikte edinebilir.

```
# tcpdump -i eth0 -n -c 5

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
01:19:25.361595 IP 91.93.119.77.ssh > 78.186.137.157.epc: P
3417325832:3417325948(116) ack 2217260129 win 8576
01:19:25.361882 IP 91.93.119.77.ssh > 78.186.137.157.epc: P 116:232(116) ack 1
win 8576
01:19:25.372120 IP 78.186.137.157.epc > 91.93.119.77.ssh: . ack 0 win 16072
01:19:25.372300 IP 91.93.119.77.ssh > 78.186.137.157.epc: P 232:528(296) ack 1
win 8576
01:19:25.372913 IP 91.93.119.77.ssh > 78.186.137.157.epc: P 528:644(116) ack 1
win 8576


# tcpdump -i eth0 -n -c 5 -v
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
01:19:48.408084 IP (tos 0x10, ttl  64, id 44909, offset 0, flags [DF], proto: TCP (6),
length: 92) 91.93.119.77.ssh > 78.186.137.157.epc: P 3417327392:3417327444(52)
ack 2217260441 win 8576
01:19:48.409330 IP (tos 0x10, ttl  64, id 44910, offset 0, flags [DF], proto: TCP (6),
length: 156) 91.93.119.77.ssh > 78.186.137.157.epc: P 52:168(116) ack 1 win 8576
01:19:48.419563 IP (tos 0x0, ttl 120, id 53010, offset 0, flags [DF], proto: TCP (6),
length: 40) 78.186.137.157.epc > 91.93.119.77.ssh: ., cksum 0x744a (correct), ack
52 win 16056
01:19:48.419801 IP (tos 0x10, ttl  64, id 44911, offset 0, flags [DF], proto: TCP (6),
length: 496) 91.93.119.77.ssh > 78.186.137.157.epc: P 168:624(456) ack 1 win 8576
01:19:48.420978 IP (tos 0x10, ttl  64, id 44912, offset 0, flags [DF], proto: TCP (6),
length: 268) 91.93.119.77.ssh > 78.186.137.157.epc: P 624:852(228) ack 1 win 8576
```

### Promisc Moddan Kaçış ( -p )

-p parametresi ile de sniff yaptığımız arabirimin promisc moddan çıkmasını sağlanabilir.

Promisc moddan çıkmak ne sağlar?

Promisc moddan çıkmakla sadece o arabirime gelen ve o arabirimi ilgilendiren paketler işlenir ki bu paketlerde ya broadcast ya da direct o arabirimin adresi olması demektir. Daha çok tcpdump'ın çalıştığı makineye ait bir paket analizi yapmak istediğimiz zaman kullanılabilecek türden bir parametredir.

```
# tcpdump -p -i eth0
```

### Layer 2 Başlıklarını Yakalama ( -e )

Tcpdump kullanarak ethernet başlık bilgileri de yakalanabilir. Özellikle yerel ağlarda yapılan trafik analizlerinde MAC adresleri önemli bilgiler vermektedir.

```
# tcpdump -t -nn –e
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes

00:0b:db:1c:4b:61 > 00:02:44:27:73:79, ethertype IPv4 (0x0800), length 52: IP
192.168.0.100.32768 > 192.168.0.1.33435: UDP, length 10
00:0b:db:1c:4b:61 > 00:02:44:27:73:79, ethertype IPv4 (0x0800), length 52: IP
192.168.0.100.32768 > 192.168.0.1.33436: UDP, length 10
00:02:44:27:73:79 > 00:0b:db:1c:4b:61, ethertype IPv4 (0x0800), length 80: IP
192.168.0.1 > 192.168.0.100: icmp 46: 192.168.0.1 udp port 33436 unreachable
00:0b:db:1c:4b:61 > 00:02:44:27:73:79, ethertype IPv4 (0x0800), length 52: IP
192.168.0.100.32768 > 192.168.0.1.33437: UDP, length 10
```

## BPF(Berkley Packet Filter)

Tcpdump ile gelişmiş paket yakalama için BPF kullanılabilir(sadece X hostunun Y portundan gelen paketleri yakala gibi).

BPF üç ana kısımdan oluşur

### Type

Host, net, port parametreleri.

### Direction

Src, dst parametreleri.

### Protocol

Ether, fddi, wlan, ip, ip6, arp, rarp parametreleri.

### Host Parametresi

Sadece belli bir host a ait paketlerin izlenmesini isteniyorsa host parametresi kullanılabilir.

**# tcpdump host 10.0.0.21**

bu komutla kaynak ya da hedef ip adresi 10.0.0.21 olan paketlerin alınmasını istiyoruz.

### dst host (Hedef Host Belirtimi)

dst host ;hedef host olarak belirtilen adrese ait paketleri yakalar,

**# tcpdump -i eth0 dst host 10.0.0.1**

yukarıdaki komutla makinemizin eth0 arabirimine gelen ve hedefi 10.0.0.1 olan tüm paketler yakalanacaktır.

**# tcpdump -i eth0 dst host 10.0.0.1**
tcpdump: listening on eth0
10:47:20.526325 10.0.0.21 > 10.0.0.1: icmp: echo request

ile de hedef ip si 10.0.0.1 olan ip adreslerini izlemiş oluyoruz.

### src host (Kaynak Host Belirtimi)

src host tanımı ilede kaynak hostu belirterek dinleme yapabiliriz, mesela kaynak hostu 10.0.0.21 olan paketleri (10.0.0.21 makinesinde )dinlemeye alalim.

```
# tcpdump -i eth0 src host 10.0.0.21

tcpdump: listening on eth0

10:52:00.620897 10.0.0.21.3409 > baym-cs253.msgr.hotmail.com.1863: P
1541540362:1541540367(5) ack 3598940393 win 17484 (DF)
10:52:01.025286 10.0.0.21.3409 > baym-cs253.msgr.hotmail.com.1863: . ack 9 win
17476 (DF)
10:52:14.758635 10.0.0.21.4013 > 10.0.0.1.telnet: S 3499731684:3499731684(0)
win 16384 (DF)
```

sadece ip adresi değil host ismide belirtilebilir.

**# tcpdump host hotmail.com**

dst ve src i aynı komuttada kullanabiliriz.

Örnek:

kaynak ip si 10.1.0.59 hedef hostu 10.1.0.1 olan paketleri izlemek istersek

**# tcpdump src host 10.1.0.59 and dst host 10.1.0.1**

komutunu verebiliriz.

burada dikkatimizi çeken ufak bir değişiklik oldu. src host ve dst host arasına 'and' geldi, evet tcpdump ile kompleks kurallar yazarken sıkça kullanacağımız kelimelerden biri de 'and' dir, ilerleyen bölümlerde 'and' in yerine hangi dizimler gelebilir onlarıda göreceğiz.

Host parametresi ile de aynı şekilde bir sonuca ulaşabiliriz host parametresi ile kaynak ya da hedef hosttan herhangi biri uygunsa paket yakalanır.

### port Parametresi (Port Belirtimi)

Belirli bir portu dinlemek istediğimizde kullanacağımız parametredir. Host gibi src ve dst ön ek alabilir.

src ile kaynak portu dst ile hedef portu belirtilir. dst ya da src ön eki kullanılmazsa hem kaynak hemde hedef portu verilmiş olur.

**# tcpdump port 23** ile

Kaynak veya hedef portu 23 olan paketler

**# tcpdump dst port 23** ile hedef portu 23 olanlar

**# tcpdump src port 23** ile de kaynak portu 23 olan paketler izlemeye alınır.

Aşağıdaki örnekte belirli ip ve belirli port numaralarını içeren paketleri port ve isim çözümleme yapmamasını(-nn)söylüyoruz.

```
# tcpdump -nn host 192.168.2.165 and port 23


tcpdump: listening on eth0

19:20:00.804501 192.168.2.10.1221 > 192.168.2.165.23:

S2565655403:2565655403(0) win 16384 (DF)
```