Knockd ile portlarinizi güvenli acin

İçindekiler

Knockd Nedir?

Knockd istemci-sunucu mantığı ile çalışan sisteminizde bir port açık olmadığı halde uzaktan vereceğiniz belirli port yoklamaları ile istediğiniz portu açabilecek yada sisteme bağlanmadan istediğiniz komutu çalıştırabilecek bir programdır. Knockd'yi kullanabilmek için aşağıda linki verilen iki yazılımı da indirmelisiniz,sunucu uygulamasını sunucu makinenize istemci uygulamasını da sunucuya bağlanmak istediğiniz herhangi bir makineye kurabilirsiniz

Kurulum

Sunucu ve istemci tarafı için gerekli paketleri http://www.zeroflux.org/knock/knock-0.3.tar.gz adresindne indirebilirsiniz, aynı zamanda Red Hat kullananlar aşağıdaki paketleri imzalarını control ederek[*] sitemden indirebilirler.Paketler Red Hat 7.3-Fedora Linux 1 için test edilmiştir.

Sunucu RPM paketi;

http://cc.kou.edu.tr/huzeyfe/tools/knock-server-0.3-2.i386.rpm
MD5 özeti 829b8b9661b691d06ccb07cd25046883

İstemci RPM Paketi;
http://cc.kou.edu.tr/huzeyfe/tools/knock-0.3-2.i386.rpm
MD5 özeti 7ae6485e8b2457077bf8440b19ca026b

istemci tarafına kurulum için,

```
[root@cc i386]# rpm -ivh knock-0.3-2.i386.rpm
Preparing...                ########################################### [100%]
   1:knock                  ########################################### [100%]
```

Sunucu tarafına kurulum için,

bash-2.05b# rpm -ivh knock-server-0.3-2.i386.rpm

Preparing... ### [100%]

1:knock-server ### [100%]

Çalışma Mantığı

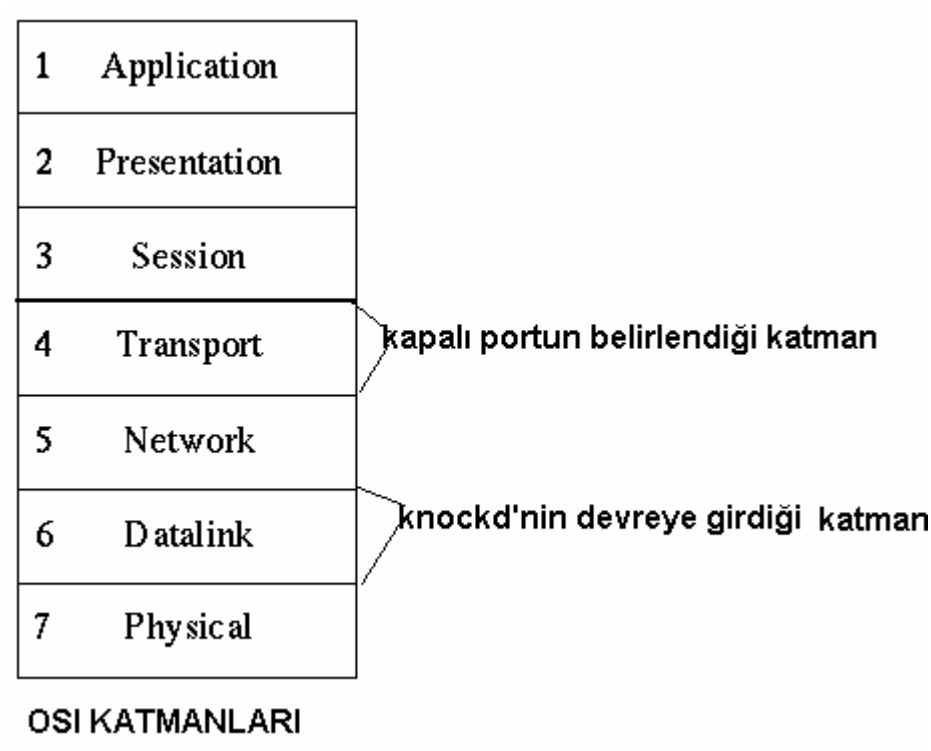
Karmaşık gözükmesine rağmen oldukça basit bir mantıkla çalışıyor "knockd", yaptığı iş ağ arabirimine (Ethernet,ppp) gelen tüm trafiği dinlemek ve konfigürasyon dosyasında belirtilen uyarlamalara uygun bir paket geldiginde yine konfigürasyon dosyasında belirtildiği komutları çalıştırmak.

Istemci belli sirada port yoklaması yapar, istemci port yoklamasını tcp ya da udp portlarda yapabilir yoklama icin istemci paketini sisteme kurup kullanabilirsiniz. Ve en önemliside yoklama yaptığınız portun açık olup olmamasının öneminin olmadığıdır, normalde kapalı bir porta göndereceğiniz paketlerden eğer filtrelenmemişse RST cevabı döner , aşağıda bir hosttan diğerinin 3000.portuna yapılan bir bağlantı isteğinin reddedildiğini gösteren tcpdump çıktısı yer alıyor.

```
194.27.72.80.51661 > 194.27.72.72.3000: S 689686850:689686850(0) win 5840 <mss
1460,sackOK,timestamp 1998810522 0,nop,wscale 0> (DF) [tos 0x10]
194.27.72.72.3000 > 194.27.72.80.51661: R 0:0(0) ack 689686851 win 0 (DF) [tos 0x10]
```

resim-1a

peki nasıl oluyorda « knockd « bu işi yapabiliyor ? Bunun için biraz TCP/IP bilgilerimizi güncelleyelim bir paket bir hosttan diğerine gönderildiğinde OSI katmanının en üst seviyesinden an altına kadar yolculuk yapar ve fiziksel ortama bırakılır , karşı tarafa ulaştığında ise OSI katmanının en alt katmanından en üst (Application Level)katmanına kadar ters bir yolculuk geçirir, bu katmanlardan herbirinin kendine göre farklı bir görevi vardır, port kavramının sözünün geçtiği katmanda 4.katmandır. Detaylı gösterim için **resim-2a** ya bakabilirisiniz. İşte knockd paketi daha 4.katmana ulaşmadan data Link layer(2)da yakalıyor, okuyor ve içeriğine gore işlemi gerçekleştiriyor, nasıl iyi fikir değil mi?



resim-2a

Sunucu Kullanımı

Sunucu kullanımda kullanabileceğimiz seçenekler

-i ile hangi arabirimin dinleneceğini belirliyoruz,herhangi bir değer belirtmezsek varsayılan olarak ilk Ethernet kartını(Linux'lar için eth0)dinlemeye alacaktır.

-d knockd nin bir servis olarak hizmet vermesini belirliyoruz.

-c <dosya_ismi> opsiyonu ile de kullanılacak yapılandırma dosyasını belirliyoruz varsayılan olarak bu değer **/etc/knockd.conf** tur,
-D parametresi ile de sunucu programının debug modda çalışmasını sağlıyoruz

-V ile calışan programın versiyonunu öğrenebiliriz.

-h parametresi kullanılabilecek parametreleri göstermeye yarar.
Data detaylı kullanım için man knockd komutunu kullanabilirsiniz.

Istemci Kullanımı

Istemci kullanımının seçeneklerine ulaşmak için
huzeyfe@cc tools~>$ knock komutunu vermeniz yeterlidir,bu seçeneklerin ne işe yaradığı aşağıda belirtilmiştir.

usage: knock [options] <host> <port> [port] ...
options:

-u, --udp	UDP paketi yollamak için,varsayılan değer TCP dir.
-v, --verbose	Detaylı bilgi için
-V, --version	Versiyon öğrenmek için.
-h, --help	Bu menu için.

Yapılandırma Dosyası

Programla birlikte gelen varsayılan yapılandırma dosyasının /etc/knockd.conf olduğundan bahsetmiştik, şimdi de bu dosyanın içeriğine bakarak yorumlayalım sonrada çeşitli örneklerle nasıl kullanılacağını anlamaya çalışalım.

bash-2.05b# cat /etc/knockd.conf

```
[options]
        UseSyslog

[opencloseSSH]
        sequence      = 2222:udp,3333:tcp,4444:udp
        seq_timeout   = 15
        tcpflags      = syn,ack
        start_command = /sbin/iptables -A INPUT -s %IP% -p tcp --dport ssh -j
ACCEPT
        cmd_timeout   = 10
        stop_command  = /sbin/iptables -D INPUT -s %IP% -p tcp --dport ssh -j
ACCEPT
```

Bu basit öntanımlı yapılandırma dosyası ile knockd'nin loglama mekanizması olarak sistemin kendi log mekanizmasını kullanmasını söylüyoruz, Eğer system log mekanizmasında farklı bir yere loglanmasını istersek bunu

```
logfile = /var/log/knockd.log
```

şeklinde belirtebiliriz. Tabii bu dosyayı sistemde oluşturup gerekli hakları atamak

suretiyle kullanabiliriz.

bash-2.05b# touch /var/log/knockd.log

izinlerini knockdyi çalıştıran kullanıcı olarak ayarlamanız lazım , farklı bir durum gözetmezseniz root kullanıcısı ile çalıştığından dosya erişim izinleri de ona göre ayarlanmış olur.

*alt tarafta [**opencloseSSH**] ile yeni bir kural için tanımlayıcı isim belirleyip seçeneklerini yazıyoruz.*

Sequence ile istemcinin port yoklama sırasını belirtiyoruz,

seq_timeout

bu değişkenle istemcinin port yoklama işlemini yaparken iki yoklama arasında max ne kadar sure bekleyebileceğini bildiriyoruz.

start_command

Uygun port yoklaması oluştuktan sonra başlatılacak komut

cmd_timeout

komut sonrası ne kadarlık bir sure bekleneceği

stop_command

cmd_timeout sonrası işletilecek komut

Aşağıdaki örnekte yapılandırma dosyasındaki seçeneklerin değerleri ve ne işe yaradıkları konusunda geniş bilgilendirme bulabilirsiniz.

Örnek;

Sistemin 100, 200 ve 300.portlarına sırası ile SYN paketi gönderilmesi halinde /tmp dizininde deneme-123 adında bir dosya oluşturmasını isteyelim.

Bunun için **/etc/knockd.conf** dosyasını herhangi bir editörle açıp dosyanın sonuna

[dosya ac]
sequence = 100,200,300
protocol = tcp
timeout = 15
command = mkdir /tmp/deneme-1-2-3
tcpflags = SYN

satırlarını ekleyelim

[dosya ac]

satırı ile açıklama belirtiyoruz, bu başlık loglarda görünecek olan başlıktır

sequence = 100,200,300
ile hangi sıra ile portların yoklanacağını belirtiyoruz

protocol = tcp
protocol tipini belirliyoruz

timeout = 15
zaman aşımını ne kadar olacağını belirliyoruz.

command = mkdir /tmp/deneme-123
yoklamalar sonrası hangi komutun çalıştırılacağını belirliyoruz

tcpflags = SYN
protocol olarak TCP belirledikten sonra hangi TCP bayrağı ile yoklama yapılacağının belirtilmesi
bunları yazdıktan sonra dosyayı kaydedip çıkalım ve istemci tarafındaki knock programcığını aşağıdaki gibi çalıştıralım.

[root@cc root]# knock yubam -v 100 200 300
hitting tcp 194.27.72.88:100
hitting tcp 194.27.72.88:200
hitting tcp 194.27.72.88:300

tekrar /tmp dizinine bakalım

[root@yubam tmp]# ls
deneme-1-2-3 *mc-root*
gconfd-root *orbit-root*
GSLhtmlbrowser5493 *sess_33c22dbdd8f5fbf788f8ccf9ecbb519a*
kde-root *sess_4d0f6ce5ca90d758cc126ad992219b85*
ksocket-root *splint-3.1.1*
mapping-root *splint-3.1.1.Linux.tgz*
mcop-root
görüldüğü gibi deneme-1-2-3 dosyası oluştu

Örnek;

```
[options]
        logfile = /var/log/knockd.log

[opentelnet]
        sequence    = 7000,8000,9000
        seq_timeout = 10
        tcpflags    = syn
        command     = /usr/sbin/iptables -A INPUT -s %IP% -p tcp --
dport 23 -j ACCEPT

[closetelnet]
        sequence    = 9000,8000,7000
        seq_timeout = 10
        tcpflags    = syn
        command     = /usr/sbin/iptables -D INPUT -s %IP% -p tcp --
dport 23 -j ACCEPT
```

bu örnekte yukarıdaki örnekten farklı olarak iki farklı seçenek kullandık biri [opentelnet] diğeri [closetelnet], [opentelnet] ile 7000,8000,9000 portlarına sırası ile gelecek SYN paketleri karşılığında sistemin ne yapacağını , [closetelnet] ile de 9000,8000,7000 portlarına sırası ile gelecek SYN paketi ile sistemin ne yapacağını belirttik, aşikarki [opentelnet] ile sisteme belirli aralıklarda syn paketi yollayan IP ye Firewall dan 23.port için geçiş hakkı tanıdık aynı şekilde [closetelnet]ile de tam tersi bir işlem yaparak o IP ye 23.portu kapadık.Yani bir önceki örneğe göre işimizi zamana bırakmadık, işimizin bittiğini ve sunucunun telnet portunu kapatmasını elle sağlamış olduk.

Sunucu Yazılımın Çalışma Modları

Sunucuyu debug ve verbose modda çalıştıralım ve aralarındaki farklılıkları görelim.

Şimdide istemci tarafında knock programını çalıştırıp sunucu tarafındaki değişiklikleri inceleyelim, çalışma parametrelerinin ne işe yaradığı yukarıda anlatılmıştı.

Sunucunun Debug Modda Çalıştırılması

Sunucu yazılımını debug modda çalıştırmak için knockd'ye verilen parametrelere ek olarak –D parametresini eklememiz gerekir.Bir önceki oturumu debug modda başlatırsak aşağıdaki gibi bir çıktı alırız.

[root@yubam tmp]# knockd -i eth1 -D

config: new section: 'options'

config: usesyslog

config: new section: 'opentelnet'

config: opentelnet: sequence: 7000,8000,9000

config: opentelnet: protocol: tcp

config: opentelnet: timeout: 15

config: opentelnet: cmd: /sbin/iptables -A INPUT -s %IP% -p tcp --dport telnet -j ACCE

PT

config: tcp flag: SYN

config: new section: 'closetelnet'

config: closeSSH: sequence: 9000,8000,7000

config: closeSSH: protocol: tcp

config: closeSSH: timeout: 15

config: closeSSH: cmd: /sbin/iptables -D INPUT -s %IP% -p tcp --dport telnet -j ACC

EPT

config: tcp flag: SYN

config: new section: 'dosya ac'

config: dosya ac: sequence: 100,200,300

config: dosya ac: protocol: tcp

config: dosya ac: timeout: 15

config: dosya ac: cmd: mkdir /tmp/deneme-1-2-3

config: tcp flag: SYN

2004-04-30 23:55:21: tcp: 81.214.131.55:2706 -> 194.27.72.88:22 106 bytes

packet is not SYN, ignoring...

packet is not SYN, ignoring...

packet is not SYN, ignoring..

Sunucu Yazılımını Verbose Modda Çalıştırmak

knockd yi verbose modda çalıştırıp istemcide de

[root@cc root]# knock yubam -v 100 200 300

hitting tcp 194.27.72.88:100

hitting tcp 194.27.72.88:200

hitting tcp 194.27.72.88:300

komutunu verdiğimizde aşağıdaki çıktıyı alırız.

[root@yubam tmp]# knockd -i eth0 -v
listening on eth0...
194.27.72.80: dosya ac: Stage 1
194.27.72.80: dosya ac: Stage 2
194.27.72.80: dosya ac: Stage 3
194.27.72.80: dosya ac: OPEN SESAME
running command: mkdir /tmp/deneme-1-2-3
mkdir: `/tmp/deneme-1-2-3' dizini oluÅturulamÄ±yor: Dosya var
dosya ac: command returned non-zero status code (1)

yukarıdaki hata biraz once o dosyanın oluşturulumuş olduğundan veriliyor,

knockd'yi sonlandırmak isterseniz aşağıdaki komutları çalıştırmanız yeterlidir.

 # ps axu|grep knockd
root **15345** 0.0 0.0 1640 460 ? S 23:27 0:00 knockd -i eth0 -d
root 15373 0.0 0.1 5132 584 pts/7 S 23:34 0:00 grep knockd

 # kill 15345

[*][huzeyfe@cc tools]$ md5sum knock-0.2.1-1.i386.rpm
7ae6485e8b2457077bf8440b19ca026b knock-0.2.1-1.i386.rpm
şeklinde control edebilirsiniz.

Huzeyfe ÖNAL huzeyfe[at]cc.kou.edu.tr

Kaynaklar:
http://www.zeroflux.org/knock/
http://www.packetfactory.net/