CHROOT Nedir?

Gökhan ALKAN, gokhan@enderunix.org

İÇİNDEKİLER

1. Motivasyon

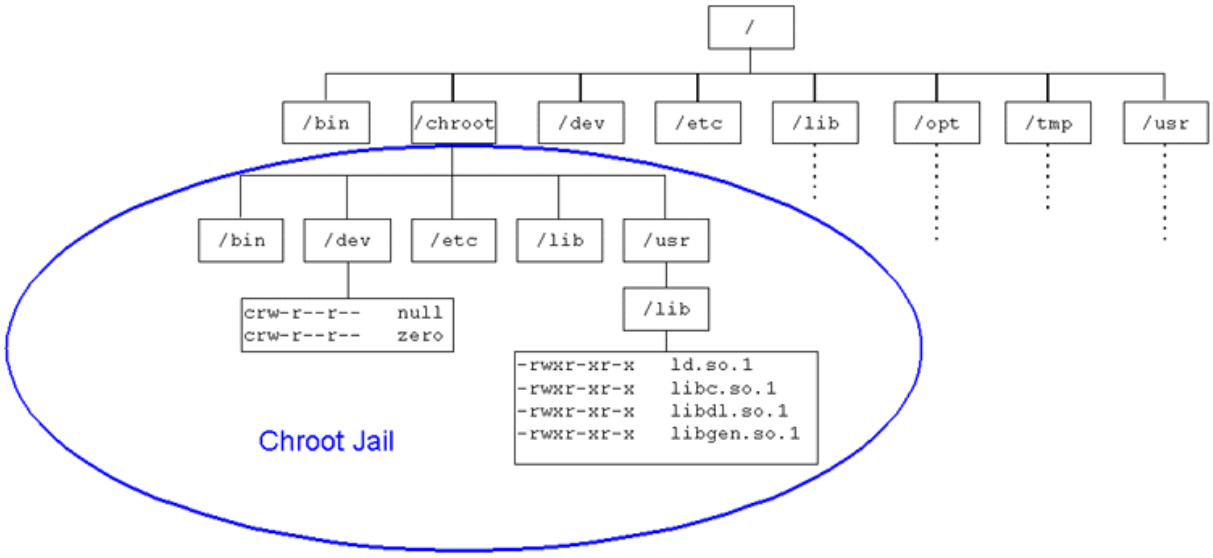
Bu makalede özellikle sunucu servis yazılımlarının ve uygulamaların *chroot* ortamında çalıştırılması, kademeli güvenlik anlayışı gibi yaklaşımlardan ziyade basit olarak *chroot nedir* ve neden *chroot* kullanmak gerekli sorusuna cevap aranmaya çalışılacaktır.

2. Chroot Nedir?

Chroot yazılımlar için yeni bir kök (/) dizini tanımlar. Kısaca çalıştırılacak olan servis ya da uygulama için gerekli kütüphane, yapılandırma, sürücü dosyaları (device file), bu servis için belirlenen kök dizinde bulunan ilgili yollara (path) kopyalanır ve hapsedilmiş olarak çalıştırılır.

Peki *chroot* kullanmanın getirdiği avantajlar ve dezavantajlar nelerdir ve neden kullanmak gerekli sorusunu kendinize sormuş olabilirsiniz. Kısaca bu sorunun cevabı, eğer sistem üzerinde *chroot* ortamında çalışan bir servis ya da uygulamadan kaynaklanan bir açıklığı kullanarak sisteme sızan bir saldırgan, bütün dosya sistemine ve işletim sistemi kaynaklarına erişim sağlayamayacak ve sadece *chroot* için belirlenen kök dizine ve burada bulunan kaynaklara erişim sağlayabilecektir. Ayrıca kısıtlı dosya sistemi erişimi sağlaması yanında kısıtlı komutlara erişim ile de sistem üzerinde bulunan komutlar kümesine erişim de kısıtlanmış olacaktır.

Aşağıdaki şekilde (Şekil 1 **Chroot Kullanımı İle Yeni Dosya Sistemi Hiyerarşisi**) *chroot* kullanımı ile dosya sistemine erişim şematize edilmiştir.



Şekil 1 Chroot Kullanımı İle Yeni Dosya Sistemi Hiyerarşisi

3. Chroot Nasıl Çalışır?

Chroot komutu '*chroot /chroot* [komut seti]' şeklinde çalıştırıldığında artık yeni kök dizini olarak '/chroot' görülecektir.

NOT: Burada "*/chroot*" dizini özellikle belirtilmemiştir. Bu dizin yerine isteğe göre herhangi bir dizin seçilebilir.

```
# mkdir /chroot
# chroot /yeni_dizin
```

Bütün bu işlemler için *chroot* komutu, *chroot* sistem çağrısını kullanır. Kısaca önce belirlenen yeni kök dizini içerisine girilir ve arkasından *chroot* sistem çağrısı çalıştırılır ve son olarak belirlenen kullanıcı kimliğine bürünülerek *root* kullanıcı haklarından vazgeçilir.

```
# chdir("/chroot");
# chroot("/chroot");
# setuid(500);
```

NOT: Sistem üzerinde 500 uid'sine sahip bir kullanıcı olduğu varsayılmıştır.

4. Örnek Chroot Ortamının Oluşturulması

Daha öncede belirtildiği gibi sadece saldırganın bütün dosya sistemine ve işletim sistemi kaynaklarına erişimi engellenmiş olacaktır. Aşağıda *chroot* kullanıma basit bir örnek verilmiştir. *Bash* kabuğu *chroot* ortamı altında çalıştırılacak ve birkaç örnek komut *chroot* ortamı altında çalıştırılacaktır.

```
# mkdir /chroot
# which bash
/bin/bash
#
# cd /chroot
# mkdir /bin
# mkdir /lib
# ldd /bin/bash
        linux-gate.so.1 =>  (0x00712000)
        libtermcap.so.2 => /lib/libtermcap.so.2 (0x00c49000)
        libdl.so.2 => /lib/libdl.so.2 (0x00c43000)
        libc.so.6 => /lib/libc.so.6 (0x00afe000)
        /lib/ld-linux.so.2 (0x00ae0000)
#
```

Bu çıktıya göre *bash* için gerekli kütüphane dosyaları teker teker belirlenmeli ve ilgili dizinlere kopyalanmalıdır. Hangi kütüphane dosyalarına ihtiyaç olduğu '*ldd*' komutu aracılığı ile öğrenilebilir.

```
# cp –p /lib/libtermcap.so.2 /home/ga/chroot/lib
# cp –p  /lib/libdl.so.2 /home/ga/chroot/lib
# cp –p  /lib/libc.so.6 /home/ga/chroot/lib
# cp –p  /lib/ld-linux.so.2 /home/ga/chroot/lib
```

Ardından *chroot* komutu ile *bash* kabuğu *chroot* olarak belirtilen dizinde çalışmaya başlayacaktır. *Chroot* ortamı altında çalıştırılmak istenen komutlar ve bu komutlar için gerekli kütüphane dosyaları '*ldd*' komutu yardımı ile bulunarak ilgili dizinlere kopyalanmalıdır.

```
# cp –p /bin/bash /chroot/bin
# cp –p /bin/pwd /chroot/bin
# chroot /chroot/ /bin/bash
bash-3.2# pwd
/
bash-3.2# exit
```

Görüldüğü gibi '*pwd*' komutu çalıştırıldığında '*/chroot*' dizinini kök *(/)* dizini olarak görülmektedir. *Chroot* ortamından çıkmak için '*exit*' komutu kullanılabilir. '*pwd*' gibi çalıştırılmak istenen komutlar gerekli kütüphane dosyaları ile birlikte ilgili dizinlere kopyalanarak çalıştırılabilirler.

Burada verilen basit bir örnektir. Asıl *chroot* kullanım amacı dosya sistemi hiyerarşisi üzerinde dağıtık biçimde erişim sağlayabilen servisler için kullanımı olmalıdır. Bu şekilde saldırgan, sistem üzerinde çalışan bir servisten ya da uygulamadan kaynaklanan bir açıklık ile sisteme sızmayı başardığında bütün dosya sistemine erişimi engellenmiş olacaktır.

5. Kaynaklar

[1] http://www.sun.com/bigadmin/content/developer/howtos/images/chrooted_fig1.gif

[2] man chroot