

☀ Chengwei's Words

{hello world}

为什么 netstat 对某些服务只显示了 tcp6 监听端口

tags: linux

最近偶尔发现一个比较奇怪的现象，netstat 查看监听的服务端口时，却只显示了 tcp6 的监控，但是服务明明是通过 tcp4 的 ipv4 地址访问的，那为什么没有显示 tcp4 的监听呢？

以 sshd 监听的 22 端口为例：

```
# netstat -tlnp | grep :22
tcp        0      0 0.0.0.0:22          0.0.0.0:*          LISTEN     1444/sshd
tcp6       0      0 :::22              :::*                LISTEN     1444/sshd
```

可以看到，netstat 显示表示 sshd 既监听在 ipv4 的地址，又监听在 ipv6 的地址。

而再看看 httpd 进程：

```
# netstat -tlnp | grep :80
tcp6       0      0 :::80              :::*                LISTEN     19837/httpd
```

却发现只显示了监听在 ipv6 的地址上，但是，通过 ipv4 的地址明明是可以访问访问的。

下面来看下怎样解释这个现象。

首先，关闭 ipv6 并且重启 httpd：

```
# sysctl net.ipv6.conf.all.disable_ipv6=1
# systemctl restart httpd
```

现在，看下 httpd 监听的地址：

```
# netstat -tlnp | grep :80
tcp        0      0 0.0.0.0:80          0.0.0.0:*          LISTEN     33697/httpd
```

可以看到，已经只监听到 ipv4 地址了。

那为什么在 ipv6 开启的时候，netstat 只显示了 tcp6 的监听而非像 sshd 那样既显示 tcp 又显示 tcp6 的监听呢？

我们下载 httpd 的源码看一看，在代码 server/listen.c 的 open_listeners() 函数中，有相关注释：

```
/* If we have the unspecified IPv4 address (0.0.0.0) and
 * the unspecified IPv6 address (::) is next, we need to
 * swap the order of these in the list. We always try to
 * bind to IPv6 first, then IPv4, since an IPv6 socket
 * might be able to receive IPv4 packets if V6ONLY is not
 * enabled, but never the other way around.
 * ... 省略 ...
 */
```

上面提到，ipv6 实际上是可以处理 ipv4 的请求的当 V6ONLY 没有开启的时候，反之不然；那么 V6ONLY 是在什么时候开启呢？

继续 follow 代码到 make_sock() 函数，可以发现如下代码：

```

#if APR_HAVE_IPV6
#ifdef AP_ENABLE_V4_MAPPED
    int v6only_setting = 0;
#else
    int v6only_setting = 1;
#endif
#endif

```

在这个函数中，可以看到如果监听的地址是 ipv6，那么会去设置 IPV6_V6ONLY 这个 socket 选项，现在，关键看 AP_ENABLE_V4_MAPPED 是怎么定义的。

在 configure（注意，如果是直接通过代码数获取的，可能没有这个文件，而只有 configure.ac/in 文件）文件中，可以找到：

```

# Check whether --enable-v4-mapped was given.
if test "${enable_v4_mapped+set}" = set; then :
    enableval=$enable_v4_mapped;
    v4mapped=$enableval
else
    case $host in
        *freebsd5*|*netbsd*|*openbsd*)
            v4mapped=no
            ;;
        *)
            v4mapped=yes
            ;;
    esac
    if ap_mpm_is_enabled winnt; then
        v4mapped=no
    fi
fi

if test $v4mapped = "yes" -a $ac_cv_define_APR_HAVE_IPV6 = "yes"; then
    $as_echo "#define AP_ENABLE_V4_MAPPED 1" >>confdefs.h

```

所以，在 Linux 中，默认情况下，AP_ENABLE_V4_MAPPED 是 1，那么 httpd 就会直接监听 ipv6，因为此时 ipv6 的 socket 能够处理 ipv4 的请求；另外，bind() 系统调用会对用户空间的进程透明处理 ipv6 没有开启的情况，此时会监听到 ipv4。

而如果我们在编译 httpd 的时候使用 --disable-v4-mapped 参数禁止 ipv4 mapped，那么默认情况下，httpd 会分别监听在 ipv4 和 ipv6，而非只监听 ipv6，如下所示：

```

# netstat -tlnp | grep :80
tcp        0      0 0.0.0.0:80          0.0.0.0:*          LISTEN     40576/httpd
tcp6       0      0 :::80              :::*                LISTEN     40576/httpd

```

而，如果在 /etc/httpd/conf/httpd.conf 中将 Listen 设置为只监听 ipv6 地址，如下：

```
Listen :::80
```

那么，将可以看到 netstat 只显示 tcp6 的监听：

```
# systemctl restart httpd
# netstat -tlnp | grep :80
tcp6      0      0 :::80          :::*           LISTEN     40980/httpd
```

并且，你会发现现在不能通过 ipv4 地址访问 httpd 了。

```
# telnet 192.168.1.100 80
Trying 192.168.1.100...
telnet: Unable to connect to remote host: Connection refused
```

所以，**netstat** 只是很真实的显示监听的端口而已，但是需要注意 **ipv6** 实际上在 **Linux** 上也支持 **ipv4**。

Posted on Mar 05, 2017

Copyright © 2021 [Chengwei Yang](#). Powered by [Jekyll](#), theme by [Matt Harzewski](#)