SmbClient Format String

简单分析

- 这个漏洞只是发生在samba的组件smbclient中，所以我们在配置samba服务的时候，可以取巧，直接使用系统中提供的samba软件包就可以了！
- 需要源码编译的只有smbclient的代码，但是smbclient的代码在samba代码里面，没办法直接单独编译，所以最简单的办法就是编译整个samba，然后使用的时候只是使用smbclient即可。
- 配置环境：Ubuntu 14.04LTS

安装

- sudo apt-get install samba samba-client
  # service will automate start
- 注释：为什么我要在这里安装samba-client，因为使用系统中smbclient可以用于排除是我们配置samba出的问题，而不是因为smbclient 中的漏洞而导致的问题。

Configuration

- configure share folder:
- add the following content to your /etc/samba/smb.conf
- [vmware]
- comment = vmware
- path = /home/mudongliang/vmware/
- # path should exist
- browseable = yes
- writable = yes # writable must be yes

Configuration

- add user :
- smbpasswd -a mudongliang
- #your own user name and set password
- connect to the samba server:
- smbclient //127.0.0.1/vmware/ -U mudongliang
- Comment：use smbclient provided by samba-client package to test your configuration about samba.

Source Code Compiling

- Download the source code of samba 3.2.12.
- cd source/
- ./autogen.sh
- ./configure
- make
- sudo make install
- Comment: 默认 samba，samba-client会安装到/usr/local/samba中。

Problems

- 这里会出现缺少共享库的问题，直接从 /usr/local/samba/lib/ 里面拷贝到lib库目录下，就可以了

```
$ cd /usr/local/samba/bin

$ ./smbclient //127.0.0.1/vmware -U mudongliang
./smbclient: error while loading shared libraries: libtalloc.so.1: cannot open shared
object file: No such file or directory

$ ldd smbclient
    linux-vdso.so.1 =>  (0x00007ffd7813b000)
        libresolv.so.2 => /lib/x86_64-linux-gnu/libresolv.so.2 (0x00007fc053cba000)
        libnsl.so.1 => /lib/x86_64-linux-gnu/libnsl.so.1 (0x00007fc053aa0000)
        libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2 (0x00007fc05389c000)
        libtalloc.so.1 => not found
            libtdb.so.1 => /usr/lib/x86_64-linux-gnu/libtdb.so.1 (0x00007fc05368a000)
        libwbclient.so.0 => /usr/lib/x86_64-linux-gnu/libwbclient.so.0
        (0x00007fc05347e000)
        libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007fc0530b9000)
        /lib64/ld-linux-x86-64.so.2 (0x00007fc054472000)
        libwinbind-client.so.0 => /usr/lib/x86_64-linux-gnu/samba/libwinbind-client.so.0
        (0x00007fc052eb6000)
        libbsd.so.0 => /lib/x86_64-linux-gnu/libbsd.so.0 (0x00007fc052ca7000)
        libpthread.so.0 => /lib/x86_64-linux-gnu/libpthread.so.0 (0x00007fc052a89000)

$ cd ../lib
$ sudo ln /usr/local/samba/lib/libtalloc.so.1 /lib/x86_64-linux-gnu/libtalloc.so.1
```

Problems

- 源码编译运行时的配置文件 smb.conf 不是在 /etc/samba/ 目录下，直接拷贝过去一份就好！

```
$ ./smbclient //127.0.0.1/vmware -U mudongliang
params.c:OpenConfFile() - Unable to open configuration file
"/usr/local/samba/lib/smb.conf":
    No such file or directory
    ./smbclient: Can't load /usr/local/samba/lib/smb.conf - run testparm to debug it
```

```
$ sudo cp /etc/samba/smb.conf /usr/local/samba/lib/
```

模拟成功

- 使用已经添加的账户登录进去之后，使用exploit提供的攻击例子(put aa%3Fbb),提示是文件不存在，所以需要先创建对应的文件，然后上传。
- Comment: smb.conf 配置文件中的path 最好不要是家目录，就是因为上传的操作！

```
$ /usr/local/samba/bin/smbclient //127.0.0.1/vmware -U mudongliang

# According to the logic of smbclient, file to be put must exist.
$ smb: \> !touch aa%3Fbb

$ smb: \> put aa%3Fbb
  putting file aa%3Fbb as \aa0.000000bb (0.0 kb/s) (average 0.0 kb/s)
```

缺陷代码位置

- 使用source insight 去查看 samba 的源码：
- 这个format string会出现很多命令里面，exploit里面使用的是put命令，我们就直接看put命令的源代码。

```
static int cmd_put(void)
{
    TALLOC_CTX *ctx = talloc_tos();
    char *lname;
    char *rname;
    char *buf;

    rname = talloc_strdup(ctx, client_get_cur_dir());
    if (!rname) {
        return 1;
    }

    if (!next_token_talloc(ctx, &cmd_ptr,&lname,NULL)) {
        d_printf("put <filename>\n");
        return 1;
    }

    if (next_token_talloc(ctx, &cmd_ptr,&buf,NULL)) {
        rname = talloc_asprintf_append(rname, buf);
    } else {
        rname = talloc_asprintf_append(rname, lname);
    }
```

缺陷代码分析

- 看着这个函数的第二个参数 fmt，我们就很明显地可以看到这个函数会将 lname 这个变量作为 fmt ，即format string来使用。

```
/*
 Realloc @p s to append the formatted result of @p fmt and return @p
 s, which may have moved.  Good for gradually accumulating output
 into a string buffer.
 */
char *talloc_asprintf_append(char *s, const char *fmt, ...)
{
    va_list ap;

    va_start(ap, fmt);
    s = talloc_vasprintf_append(s, fmt, ap);
    va_end(ap);
    return s;
}
```

其他位置:mkdir

```
/*****************************************************************************
 Make a directory.
*****************************************************************************/

static int cmd_mkdir(void)
{
    TALLOC_CTX *ctx = talloc_tos();
    char *mask = NULL;
    char *buf = NULL;

    mask = talloc_strdup(ctx, client_get_cur_dir());
    if (!mask) {
        return 1;
    }

    if (!next_token_talloc(ctx, &cmd_ptr,&buf,NULL)) {
        if (!recurse) {
            d_printf("mkdir <dirname>\n");
        }
        return 1;
    }
    mask = talloc_asprintf_append(mask, buf);
    if (!mask) {
        return 1;
    }
```

其他位置:mget

```
/****************************************************************************
 Do a mget command.
****************************************************************************/

static int cmd_mget(void)
{
    TALLOC_CTX *ctx = talloc_tos();
    uint16 attribute = aSYSTEM | aHIDDEN;
    char *mget_mask = NULL;
    char *buf = NULL;

    if (recurse) {
        attribute |= aDIR;
    }

    abort_mget = false;

    while (next_token_talloc(ctx, &cmd_ptr,&buf,NULL)) {
        mget_mask = talloc_strdup(ctx, client_get_cur_dir());
        if (!mget_mask) {
            return 1;
        }
        if (*buf == CLI_DIRSEP_CHAR) {
            mget_mask = talloc_strdup(ctx, buf);
        } else {
            mget_mask = talloc_asprintf_append(mget_mask,
                                               buf);
        }
```

其他位置:get

```
/****************************************************************************
 Get a file.
****************************************************************************/

static int cmd_get(void)
{
    TALLOC_CTX *ctx = talloc_tos();
    char *lname = NULL;
    char *rname = NULL;
    char *fname = NULL;

    rname = talloc_strdup(ctx, client_get_cur_dir());
    if (!rname) {
        return 1;
    }

    if (!next_token_talloc(ctx, &cmd_ptr,&fname,NULL)) {
        d_printf("get <filename> [localname]\n");
        return 1;
    }
    rname = talloc_asprintf_append(rname, fname);
    if (!rname) {
        return 1;
    }
    rname = clean_name(ctx, rname);
    if (!rname) {
        return 1;
    }
```

其他位置:del

- Comment：这个del命令就有点意思了，如果本身samba 上面有一个文件名叫做aa%3Fbb,那么就算你有权限，你就没有办法删除删除这个文件。

```
/*******************************************************************
 Delete some files.
*******************************************************************/

static int cmd_del(void)
{
    TALLOC_CTX *ctx = talloc_tos();
    char *mask = NULL;
    char *buf = NULL;
    uint16 attribute = aSYSTEM | aHIDDEN;

    if (recurse) {
        attribute |= aDIR;
    }

    mask = talloc_strdup(ctx, client_get_cur_dir());
    if (!mask) {
        return 1;
    }
    if (!next_token_talloc(ctx, &cmd_ptr,&buf,NULL)) {
        d_printf("del <filename>\n");
        return 1;
    }
    mask = talloc_asprintf_append(mask, buf);
    if (!mask) {
        return 1;
    }
```