# Análisis de Firmware

# Introducción

Este proyecto trata sobre el análisis de firmware de la cámara DCS-935L. Se llevó a cabo la extracción del binario y la exploración de sus carpetas, archivos y scripts en busca de credenciales y posibles vulnerabilidades.

El principal objetivo del proyecto es comprender en qué consiste el análisis de firmware, qué tipos de herramientas se pueden utilizar, qué tipo de información se puede encontrar y cómo un ciberdelincuentes podría aprovechar estas vulnerabilidades para llevar a cabo un ataque.

---

# Metodología de análisis de firmware

En este caso voy a tratar de guiarme por la metodología de OWASP FTSM
La metodología consisten en 9 pasos pero en mi caso voy a reducir los pasos por falta de conocimiento.

1. **Recolección de información:** Obtener detalles técnicos del dispositivo.
2. **Obtención del firmware:** Adquirir el firmware.
3. **Análisis del firmware:** Estudiar sus características.
4. **Extracción del sistema de archivos:** Extraer los contenidos del firmware.
5. **Análisis del sistema de archivos:** Realizar análisis estáticos.
6. **Emulación del firmware:** Emular los componentes.
7. **Análisis dinámico:** Pruebas de seguridad dinámicas.
8. **Análisis en tiempo de ejecución:** Analizar binarios en ejecución.
9. **Explotación binaria:** Explorar vulnerabilidades encontradas.

---

# Recolección de información

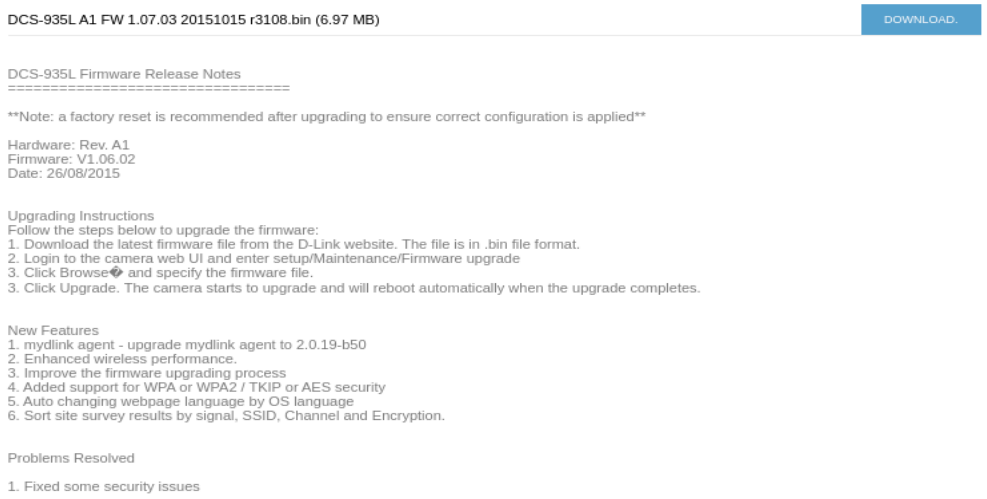
La recolección de información es un proceso amplio que puede incluir detalles sobre el hardware, el software y las vulnerabilidades conocidas del dispositivo. Sin embargo, en este caso, omitiré esa parte y me centraré en explorar el contenido del binario del firmware.
Sitio oficial: DCS-935L

---

# Obtener Firmware

Existen distintas manera de conseguir el firmware en mi caso lo descargo porque es publico,pero en muchos casos no lo vas a poder encontrar y vas a tener que buscar la forma de extraerlo manualmente manipulando el hardware del dispositivo.

Dejo un video de ejemplo para que explore: Extraction Firmware


DCS-935L A1 FW 1.07.03 20151015 r3108.bin (6.97 MB) DOWNLOAD.

DCS-935L Firmware Release Notes
==========

**Note: a factory reset is recommended after upgrading to ensure correct configuration is applied**

Hardware: Rev. A1
Firmware: V1.06.02
Date: 26/08/2015

Upgrading Instructions
Follow the steps below to upgrade the firmware:
1. Download the latest firmware file from the D-Link website. The file is in .bin file format.
2. Login to the camera web UI and enter setup/Maintenance/Firmware upgrade
3. Click Browse and specify the firmware file.
3. Click Upgrade. The camera starts to upgrade and will reboot automatically when the upgrade completes.

New Features
1. mydlink agent - upgrade mydlink agent to 2.0.19-b50
2. Enhanced wireless performance.
3. Improve the firmware upgrading process
4. Added support for WPA or WPA2 / TKIP or AES security
5. Auto changing webpage language by OS language
6. Sort site survey results by signal, SSID, Channel and Encryption.

Problems Resolved

1. Fixed some security issues

---

# Extracción del firmware y análisis

La guía recomienda usar algunos comando para obtener información del binario, pero en mi caso, no apareció nada demasiado relevante. Así que en lugar de detenerme ahí, voy a seguir adelante y extraer su contenido.

Según la guía, a veces un análisis de binario no arroja datos interesantes por varias razones:

- Puede ser BareMetal, es decir, diseñado para ejecutarse directamente en el hardware sin un sistema operativo.
- Puede ser para un sistema operativo en tiempo real (RTOS) con un sistema de archivos personalizado.
- Puede estar encriptado, lo que dificulta la extracción de información.

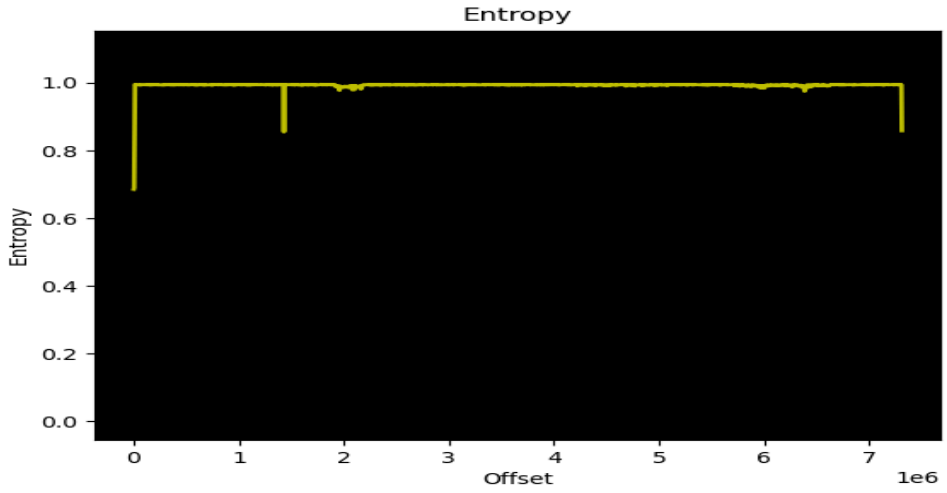
Para verificar si un binario está cifrado, se puede analizar su entropía con binwalk:

```
binwalk -E firmware.bin
```

La entropía se mantiene mayormente alta, con algunas caídas en ciertas áreas. Esto podría indicar que:

- El firmware no está completamente cifrado.

- Algunas partes pueden estar comprimidas o contener código legible.




Extraer contenido del binario.

```
binwalk -ev firmware.bin
```

Su contenido es este.

```
└─$ ls
15D822.squashfs  2818  2818.7z  squashfs-root  squashfs-root-0
```

Por el momento solo explorare la primer carpeta.

```
└─$ ls -lah squashfs-root

total 72K
drwxr-xr-x 18 thor thor 4.0K Oct 15  2015 .
drwxrwxr-x  4 thor thor 4.0K Mar 26 12:00 ..
drwxr-xr-x  2 thor thor 4.0K Oct 15  2015 bin
drwxr-xr-x  3 thor thor 4.0K Mar 26 12:00 dev
drwxr-xr-x  7 thor thor 4.0K Oct 15  2015 etc
drwxr-xr-x  2 thor thor 4.0K Oct 15  2015 home
drwxr-xr-x  3 thor thor 4.0K Oct 15  2015 lib
drwxr-xr-x  6 thor thor 4.0K Oct 15  2015 mnt
drwxr-xr-x  2 thor thor 4.0K Oct 15  2015 mydlink
drwxr-xr-x  2 thor thor 4.0K Oct 15  2015 proc
drwxr-xr-x  2 thor thor 4.0K Oct 15  2015 root
drwxr-xr-x  2 thor thor 4.0K Oct 15  2015 sbin
drwxr-xr-x  2 thor thor 4.0K Mar 26 12:00 server
drwxr-xr-x  2 thor thor 4.0K Oct 15  2015 share
drwxr-xr-x  2 thor thor 4.0K Oct 15  2015 sys
lrwxrwxrwx  1 thor thor    7 Oct 15  2015 tmp → var/tmp
drwxr-xr-x  6 thor thor 4.0K Oct 15  2015 usr
drwxr-xr-x  2 thor thor 4.0K Oct 15  2015 var
drwxr-xr-x  3 thor thor 4.0K Mar 26 12:00 web
```

El contenido del archivo 2818.

```
strings 2818 | grep -iE
"password|passwd|admin|user|key|credential|secret|username|pass"
```

Su salida no muestra algo tan relevante para mi pero si puede profundizar mas, se encontró una gran cantidad de cadenas relacionadas con credenciales y claves.

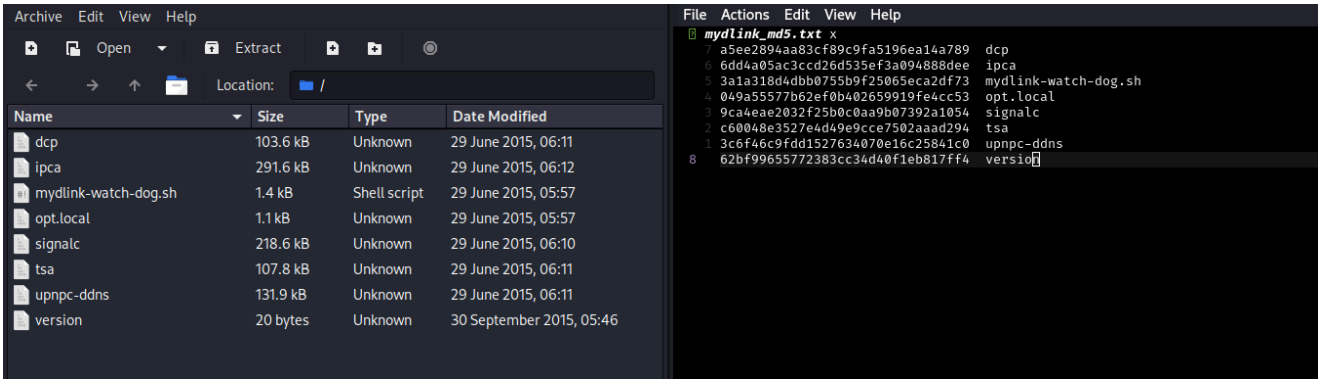
```
└─$ strings 2818 | grep -iE "password|passwd|admin|user|key|credential|secret|username|pass"

cifs_user_read
cifs_user_write
jffs2_build_inode_pass1
No init found.  Try passing init= option to kernel.
<6>User-defined physical RAM map:
 user
user.
<4>%s: %s passed in a files arraywith an index of 1!
NFS: mount program didn't pass any mount data
NFS: mount program didn't pass remote address
lkey
lkeyi
lkeyp
<3>lockd_down: no users! task=%p
<4>lockd_up: no pid, %d users??
<7> fs/cifs/cifsfs.c: Empty cifs superblock info passed to unmount
,username=%s
,nouser_xattr
<3> CIFS VFS: Server requests plain text password but client support disabled
MultiuserMount
  Local Users To Server: %d SecMode: 0x%x Req On Wire: %d
user_xattr
nouser_xattr
user
<4>CIFS: invalid or missing username
<4>CIFS: username too long
pass
<4>CIFS: no memory for password
credentials
<7> fs/cifs/connect.c: null user
<7> fs/cifs/connect.c: Username: %s
<3> CIFS VFS: No username specified
<3> CIFS VFS: Null inode passed to cifs_writeable_file
<7> fs/cifs/misc.c: Multiuser mode and UID did not match tcon uid
<7> fs/cifs/misc.c: Null buffer passed to cifs_small_buf_release
<7> fs/cifs/misc.c: Null buffer passed to tconInfoFree
<7> fs/cifs/misc.c: Null buffer passed to sesInfoFree
<3> CIFS VFS: Null parms passed to AllocOplockQEntry
```

Como objetivo principal explore la carpeta home y este contiene 3 archivos.

```
└─$ ls
mydlink_md5.txt  mydlink.tgz  RTS5826_FW.bin
```

El text contiene hash y el comprimido contiene script y ejecutables.




En la carpeta server se encuentra varios archivos pero por ahora no buscare mucho.

```
└─$ ls
accepted6.ini  aviheader.bin  camsvr.ini  httpd.ini     ipfilter.ini  profile.ini  schedule.ini  ulawfull.tbl  usr.ini    xver.ini
accepted.ini   camsvr         event.ini   ipfilter6.ini motion.ini    pt.ini       server.ini    url.ini       video.ini
```

```
 grep -iE
"password|username|admin|key|secret|pass|http|local|user|pass|cam|firmware
|dlink|monitor|server|services|ip" *.ini

xver.ini:Cabname =UltraRTCamX.cab
xver.ini:Cabname64 =UltraRTCamX64.cab
```

Navegando en la carpeta `/etc/ssl/certs` encontré certificados.

```
└─$ cat ca-bundle.crt
##
## ca-bundle.crt -- Bundle of CA Root Certificates
##
## Certificate data from Mozilla as of: Tue Apr 22 08:29:31 2014
##
```

En la carpeta `/etc/stunnel` encontré mas certificados pero en este caso son privados.

```
cat stunnel.pem
```

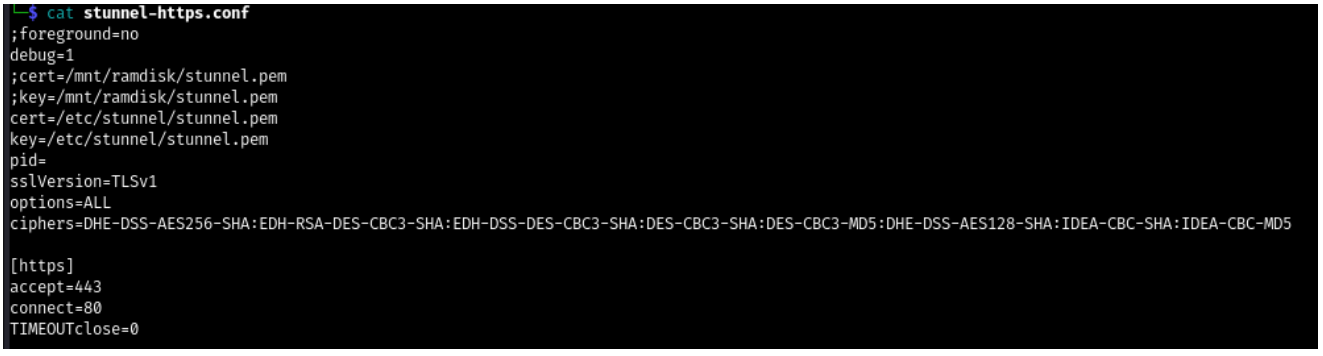
En la carpeta `/etc/Wireless` encontré configuraciones.


Realizando un grep con algunos parámetros de interés encontré algunas cosas interesante:

```
grep -iE
"password|username|admin|key|secret|credential|user|firmware|version|cam|p
ort|ip|tcp|udp|open|close" *
```

Se puede ver puertos,servicios,canales y algunos usuario.Luego de usar grep decidí explorar con cat y encontré algunas cosas mas,como la configuración de servicios ,direcciones url,configuración de tunnel y configuración onvif.





Este es todo el contenido de la carpeta etc:


Por ultimo explorare la carpeta web.


---

# Emulación del firmware

Encontré la carpeta web y decidí montar todo el sistema de archivos para poder visualizar el sitio web de la cámara.

Permite identificar las estructuras dentro del firmware.
`binwalk firmware.bin`

Para extraer el contenido del binario, utilicé el siguiente comando:
`dd if=DIR850L_REVB.bin bs=1 skip=1704084 of=dir.squashfs`

Luego, descomprimí el sistema de archivos SquashFS con el siguiente comando:
`unsquashfs dir.squashfs`

SquashFS es un sistema de archivos altamente comprimido. Esto permite almacenar una gran cantidad de datos en un espacio reducido.

Finalmente, accedí a la carpeta squashfs-root, donde se encuentran los archivos extraídos. Dentro de esta carpeta, ejecuté el siguiente comando para verificar que el sistema contenía busybox:
`cat bin/busybox`

Una vez verificado monte tres carpetas principales `proc` , `dev` y `sys` :

- **/proc**: Proporciona datos sobre el estado del sistema y los procesos en tiempo real.
- **/dev**: Contiene archivos que representan los dispositivos hardware, permitiendo su acceso y manipulación.
- **/sys**: Ofrece detalles sobre el kernel y los dispositivos, permitiendo inspeccionar y modificar parámetros relacionados con el hardware.

```
sudo mount --bind /proc
/home/thor/Desktop/Analisis_Firmware/Firmware/squashfs-root/proc

sudo mount --bind /dev
/home/thor/Desktop/Analisis_Firmware/Firmware/squashfs-root/dev
```

```
sudo mount --bind /sys
/home/thor/Desktop/Analisis_Firmware/Firmware/squashfs-root/sys
```

Una vez montado todo se procede a iniciar la shell:

```
sudo chroot . /bin/sh
```

```
└─$ sudo chroot . /bin/sh
[sudo] password for thor:
# ls
bin      etc      lib      mydlink  root     server   sys      usr      web
dev      home     mnt      proc     sbin     share    tmp      var
#
```

Para comprobar el user use el comando `whoami` y busque el archivo `passwd`.

```
# whoami
admin
#
```

```
# cat passwd
admin::0:0:root:/:/bin/sh
```

Ahora se ejecuta el script `rcS` que se inicia al arranque para configurar el sistema, montar sistemas de archivos, preparar el entorno, y ejecutar otros scripts que son necesarios para que el sistema esté listo para su uso.

```
# uname -m
mips
# /etc/rc.d/
init.d/  rcK.d/   rcS      rcS.d/
# /etc/rc.d/rcS
Tue Jul  1 00:00:00 UTC 2014
mount: mounting sysfs on /sys failed: Device or resource busy
mount: mounting /dev/mtdblock1 on /mnt/flash failed: No such file or directory
[ERROR] Can't mount /dev/mtdblock1, erase it ...
/usr/bin/flash_eraseall: /dev/mtd1: No such file or directory
Try to re-mount /dev/mtdblock1 ...mount: mounting /dev/mtdblock1 on /mnt/flash fail
ed: No such file or directory
 failed
mount: mounting /dev/mtdblock4 on /mydlink failed: No such file or directory
[ERROR] Can't mount /dev/mtdblock4, erase it ...
/usr/bin/flash_eraseall: /dev/mtd4: No such file or directory
Try to re-mount /dev/mtdblock4 ...mount: mounting /dev/mtdblock4 on /mydlink failed
: No such file or directory
 failed
check mydlink agent ...
    dcp [fail]
    ipca [fail]
    mydlink-watch-dog.sh [fail]
    opt.local [fail]
    signalc [fail]
    tsa [fail]
    upnpc-ddns [fail]
    version [fail]
re-install mydlink agent ... qemu: uncaught target signal 4 (Illegal instruction) -
 core dumped
```

Hice una comprobación de puertos con `netstat`.

```
# netstat -tln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:443             0.0.0.0:*               LISTEN
tcp        0      0 :::80                   :::*                    LISTEN
tcp        0      0 :::8088                 :::*                    LISTEN
# logread: can't find syslogd buffer: No such file or directory
```

Tiene tres puertos abierto y realice la comprobación con nmap para ver las versiones de los puertos.

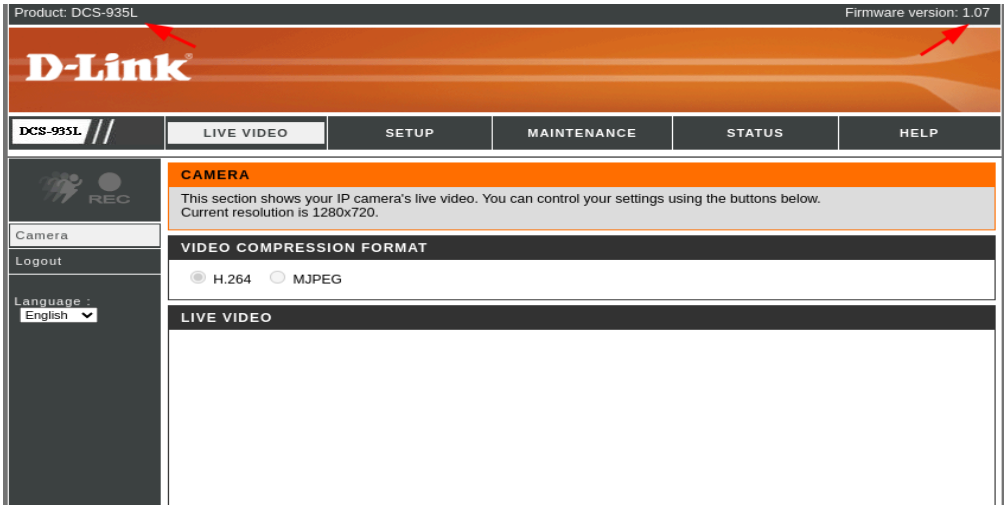
Escaneo del puerto con nmap para ver la version.

```
nmap -sV  0.0.0.0 -p 80,8080,443 -vvv
```

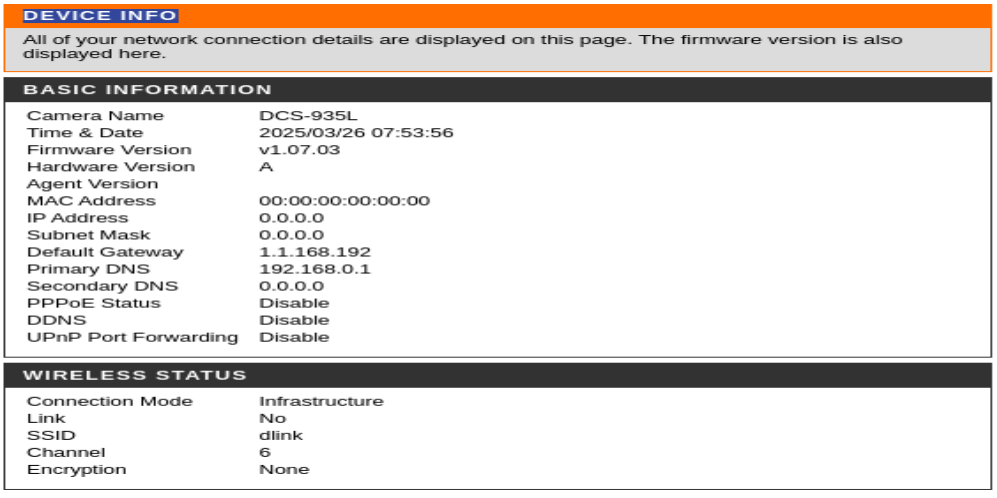
```
PORT     STATE    SERVICE    REASON         VERSION
80/tcp   open     http       syn-ack ttl 64
443/tcp  open     ssl/https? syn-ack ttl 64
8080/tcp filtered http-proxy no-response
1 service unrecognized despite returning data. If you know the service/version
ce :
SF-Port80-TCP:V=7.95%I=7%D=3/26%Time=67E459FF%P=x86_64-pc-linux-gnu%r(GetR
SF:equest,2EB,"HTTP/1\.1\x20200\x20OK\r\nConnection:\x20close\r\nDate:\x20
SF:Wed,\x2026\x20Mar\x202025\x2019:48:10\x20GMT\r\nContent-Type:\x20text/h
SF:tml\r\nExpires:\x20Mon,\x2001\x20Jul\x201980\x2000:00:00\x20GMT\r\nCach
SF:e-Control:\x20no-cache,\x20no-store,\x20must-revalidate\r\nPragma:\x20n
SF:o-cache\r\nContent-Length:\x20516\r\n\r\n<html\x20xmlns=\"http://www\.w
SF:3\.org/1999/xhtml\">\r\n<head>\r\n<link\x20rel=\"Bookmark\"\x20type=\"i
SF:mage/x-icon\"\x20\x20href=\"/favicon\.ico\">\r\n<link\x20rel=\"icon\"\x
SF:20href=\"/favicon\.ico\"\x20type=\"image/x-icon\">\r\n<link\x20rel=\"sh
SF:ortcut\x20icon\"\x20href=\"/favicon\.ico\">\r\n<meta\x20http-equiv=\"Co
SF:ntent-Type\"\x20content=\"text/html;\x20charset=utf-8\">\r\n<title>Unti
SF:tled\x20Document</title>\r\n<script\x20language=\"JavaScript\"\x20type=
SF:\"text/JavaScript\">\r\nfunction\x20redirect\(\)\r\n{\r\n\tdocument\.lo
SF:cation\.href\x20=\x20\"/home\.asp\";\r\n}\r\n</script>\r\n</head>\r\n<b
SF:ody\x20onload=\"redirect\(\);\">\r\n</body>\r\n</html>\r\n")%r(HTTPOpti
SF:ons,2EB,"HTTP/1\.1\x20200\x20OK\r\nConnection:\x20close\r\nDate:\x20Wed
SF:,\x2026\x20Mar\x202025\x2019:48:15\x20GMT\r\nContent-Type:\x20text/html
SF:\r\nExpires:\x20Mon,\x2001\x20Jul\x201980\x2000:00:00\x20GMT\r\nCache-C
SF:ontrol:\x20no-cache,\x20no-store,\x20must-revalidate\r\nPragma:\x20no-c
SF:ache\r\nContent-Length:\x20516\r\n\r\n<html\x20xmlns=\"http://www\.w3\.
SF:org/1999/xhtml\">\r\n<head>\r\n<link\x20rel=\"Bookmark\"\x20type=\"imag
```

Claramente nos muestra que contiene el sitio web de la cámara, para entrar al sitio me dirigí a la dirección `0.0.0.0:80`.

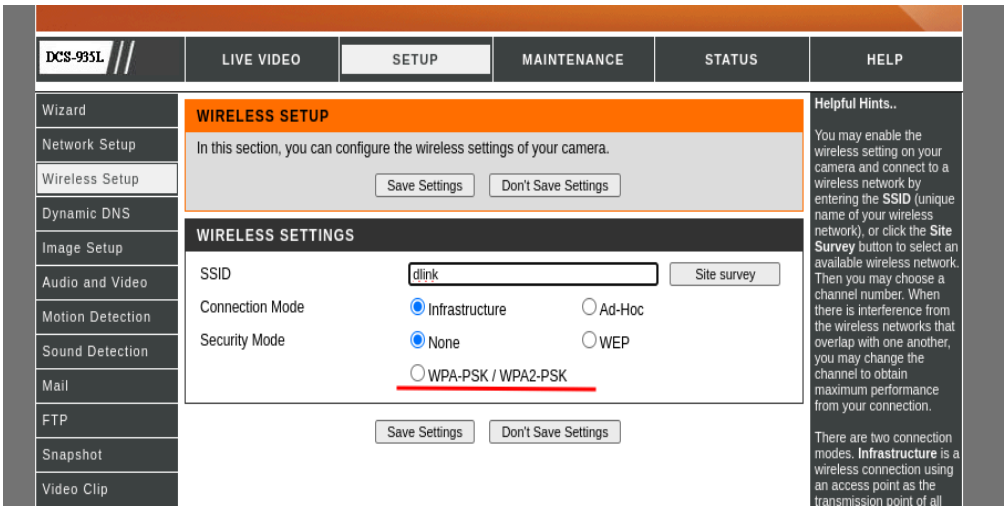
Se puede ver el la pagina principal del sitio y tipo de producto y su version de firmware.




Explorando un poco mas el sitio puedo ver mas información del dispositivo.




En otro apartado del sitio se puede ver en Wireless que se puede configurar con dos modos de seguridad.




---

# Vulnerabilidades

Estuve realizando búsqueda de vulnerabilidades en internet y no logre encontrar ninguna pero posiblemente algunas vulnerabilidades de versiones mas alta a la del firmware 1.07 puedan ser probada en esta version.

De igual forma dejare algunas de otras versiones:
CVE-2019-17146
CVE-2019-17146
CVE-2019-17146

---

# Conclusion

Se analizó el firmware de la cámara D-Link DCS-935L (versión 1.07). Aunque no se encontraron vulnerabilidades, el proceso ayudó a entender cómo analizar firmware en dispositivos IoT.

En resumen, el proyecto proporcionó conocimientos sobre cómo auditar dispositivos IoT y mejorar su seguridad.

---