

Ingeniería inversa: Identificación de Brechas de Seguridad en Cámara de Videovigilancia

Reverse Engineering: Identifying Security Gaps in Video Surveillance Cameras

Juan Felipe Pérez Roldán, Ana Claudia Zenteno Vázquez, María del Carmen Santiago Díaz, Judith Pérez Marcial, Gustavo Trinidad Rubín Linares

Facultad de Computación, Avenida San Claudio y 14 Sur, Colonia San Manuel,
Ciudad Universitaria, C.P. 72570, Puebla, Puebla. México.
juan.perezrold@alumno.buap.mx, {ana.zenteno, marycarmen.santiago, judith.perez, gustavo.rubin}@correo.buap.mx

Fecha de recepción: 19 de septiembre de 2025

Fecha de aceptación: 4 de marzo de 2026

Resumen. El presente trabajo se enfoca en el análisis de seguridad del firmware de un modelo conocido en los hogares por su precio, con el objetivo de identificar vulnerabilidades que puedan comprometer la integridad, confidencialidad o disponibilidad del dispositivo. Se realiza ingeniería inversa sobre el firmware extraído, permitiendo el análisis del sistema de archivos, donde se identificaron componentes críticos como credenciales embebidas, scripts de arranque, servicios en ejecución y binarios propietarios. Además, se llevaron a cabo pruebas de penetración orientadas al reconocimiento del dispositivo mediante técnicas como escaneo de puertos, identificación de servicios abiertos y análisis de posibles vectores de ataque. Se evalúa también el comportamiento del dispositivo en diferentes entornos de red, lo que permite identificar riesgos asociados a una configuración por defecto poco segura. Este estudio proporciona evidencia técnica de los riesgos asociados al software embebido en dispositivos IoT, y sugiere posibles mitigaciones a nivel de firmware para fortalecer su seguridad.

Palabras clave: Cámaras IP, Ciberseguridad, Vulnerabilidades, Análisis de firmware, Pruebas de penetración.

Summary. This thesis focuses on the security analysis of the TP-Link TAPO C310 IP camera firmware, aiming to identify vulnerabilities that may compromise the device's integrity, confidentiality, or availability. Reverse engineering techniques were applied to the extracted firmware image, allowing for a thorough examination of its internal file system. During the analysis, several critical elements were identified, including embedded credentials, startup scripts, configuration files, running services, and proprietary binaries. These components represent potential security risks if left unprotected or misconfigured. Furthermore, penetration testing techniques were conducted in a controlled network environment, including port scanning, service enumeration, and analysis of exposed interfaces. These tests revealed open services that could be exploited by attackers to gain unauthorized access. The device's behavior was also assessed under different operational contexts to evaluate how default settings impact overall security. This study contributes to understanding the risks associated with embedded IoT firmware and proposes firmware-level recommendations to enhance the robustness of similar devices in real-world deployments.

Keywords: IP Cameras, Cybersecurity, Vulnerabilities, Firmware Analysis, Penetration Testing.

1 Introducción

El uso de dispositivos de videovigilancia ha abierto puertas a la mejora tanto como individuos en la seguridad de los hogares, así como en empresas donde es crucial mantener una continua supervisión como lo son entidades bancarias, almacenes o plantas de montaje, pero estas nuevas oportunidades no están solo a nuestro alcance pues pueden ser tomadas como ventanas para vulnerar la privacidad de los usuarios a manos de los llamados ciber delincuentes.

El modelo Tapo C310 de TP-Link, por su amplia adopción, representa un caso relevante para estudiar su seguridad. Esta investigación busca identificar las principales brechas de seguridad presentes en estos dispositivos junto a recomendaciones propuestas por los proveedores para mitigar estas vulnerabilidades.

2 Estado del arte

El predecesor de los sistemas de videovigilancia basados en cámaras IP corresponde a los sistemas basados en cámaras análogas conectadas mediante cable coaxial al sistema de grabación DVR (Digital Video Recorders), siendo estos los únicos sistemas de videovigilancia existentes hasta la incorporación de la cámara “Neteye” de la empresa Axis en 1992, ellos combinaron la cámara de video analógica con un microprocesador especializado (Procesador RISC de 32 bit “ETRAX”) que controlaba la cámara e interactuaba con la red mediante los protocolos ARP, RARP, BOOTP, TCP, IP, HTTP ICMP y FTP. Marcando un antes y un después en el mercado de los sistemas de videovigilancia [1].

Tabla. 1. Tabla comparativa entre protocolos usados inicialmente y protocolos actuales [2]

Categoría	Protocolo anterior	Protocolo actual
Resolución de direcciones IP	RARP Y BOOTP No permiten restricciones en la asignación de direcciones IP.	DHPC Soporta asignación dinámica y permite más configuraciones como DNS y Gateway.
Comunicación WEB	HTTP Transmite datos en texto plano, lo que lo hace vulnerable a ataques de interceptación.	HTTPS Usa cifrado TLS/SSL para proteger la comunicación entre el cliente y el servidor.
Transferencia archivos	FTP No cifra los datos ni las credenciales, lo que permite ataques como la interceptación de contraseñas.	SFTP Basado en SSH, cifra la transferencia de datos, protegiendo credenciales y archivos.
Streaming video	RTP Transmite video y audio en tiempo real, pero sin cifrado, lo que permite espionaje o manipulación.	SRTP Añade cifrado y autenticación para garantizar que la transmisión no sea interceptada o modificada.

La base de datos CVE (Common Vulnerabilities and Exposures) contiene en sus registros la vulnerabilidad con ID CVE-2022-37255 donde se consigue acceso a la transmisión de video haciendo uso de credenciales conocidas previamente, siendo este vector de credenciales una de las vulnerabilidades más conocidas, simples de ejecutar y por lo tanto de mayor importancia. [3]

Diversos proveedores al descubrir nuevas vulnerabilidades realizan parches de seguridad en el firmware de las cámaras, pero esta no es la única forma de mantener seguros estos dispositivos, existen diversas recomendaciones que el usuario puede llevar a cabo para añadir capas extras de seguridad, dentro de estas recomendaciones encontramos:

- Crear usuarios con credenciales seguras: En el momento de configuración del dispositivo, se recomienda crear el usuario evitando nombres comunes como lo son “root” o “admin”, además de usar contraseñas seguras con las siguientes recomendaciones: usar más de 8 caracteres, hacer uso de combinaciones con caracteres alfanuméricos y caracteres especiales y evitar contraseñas conocidas como “12345678” o “qwerty1234”.
- Seguir paso a paso las recomendaciones propuestas por los proveedores de las cámaras en las guías de instalación del dispositivo. Cada fabricante proporciona manuales y guías de instalación para sus dispositivos. Es fundamental seguir estas instrucciones al pie de la letra, ya que incluyen configuraciones recomendadas para optimizar el funcionamiento y la seguridad de la cámara.
- Crear segmentos de red destinados únicamente a estos dispositivos. Para mejorar la seguridad de las cámaras IP, se recomienda aislarlas en una red separada del resto de los dispositivos conectados a internet.
- No descargar software o actualizaciones no autorizadas por el proveedor. Es importante asegurarse de que todas las actualizaciones y software instalados en la cámara provengan únicamente del fabricante oficial.[4]

3 Metodología

El desarrollo del presente trabajo se realiza bajo una metodología de tipo aplicada, siguiendo un enfoque experimental y analítico. Cada una de las etapas propuestas permite abordar de manera estructurada el estudio de las vulnerabilidades en cámaras IP.

3.1 Investigación: cámaras, software, protocolos.

Se lleva a cabo una investigación documental y técnica para comprender el funcionamiento de las cámaras IP, su comportamiento en la red y los diferentes protocolos que utilizan para la transmisión de datos, la autenticación y la configuración remota. Este primer paso permitirá sentar las bases para un correcto análisis posterior del firmware además de que facilita la identificación de vectores de ataques más comunes.

3.2 Análisis de software firmware de cámara Tapo c310

Se obtiene el firmware, en medio oficiales o se obtiene desde dispositivo físico haciendo uso de la técnica “*Dumping*”, vía pines UART, del modelo especificado de cámara utilizado en el estudio. A través de herramientas como “*Binwalk*”, binwalk es una herramienta para buscar en una imagen binaria archivos embebidos y código ejecutable [5], y “*usquashfs*”, *SquashFS* es un sistema de archivos de solo lectura altamente comprimido para Linux y *unsquashfs* permite descomprimir estos archivos [5], se realiza el análisis a los ficheros para identificar configuraciones por defectos, presencia de credenciales, servicios expuestos y otras vulnerabilidades. Esta etapa expondrá las debilidades del sistema desde su núcleo.

3.3 Pruebas de penetración al equipo físico

En esta etapa se realizan pruebas de penetración directamente al equipo físico (cámara IP modelo Tapo c310) para validar vulnerabilidades documentadas en pasado y descubrir nuevas vulnerabilidades presentes.

Estas pruebas se llevan a cabo en un entorno controlado, respetando las prácticas éticas con fines académicos. Los resultados que se obtienen permiten determinar el nivel de exposición de la cámara en configuración de fábrica conectada a una red doméstica y justificar la necesidad de aislarla en una red segura.

3.4 Análisis de implementaciones de seguridad de la cámara Tapo c310

En esta etapa se analizan implementaciones de seguridad de la cámara Tapo c310, una vez identificado el comportamiento del dispositivo antes de las pruebas de penetración, se analizan las medidas y configuraciones de seguridad que se ofrecen por defecto; autenticación con factor de dos pasos, actualizaciones de firmware y uso de aplicación móvil. Se evalúan elementos como autenticación, cifrado, servicios y actualizaciones.

4 Resultados experimentales

El análisis del firmware de un dispositivo IoT requiere una serie de pasos estructurados que permitan descomponer la imagen binaria del sistema en diferentes componentes. En esta investigación se utilizó la siguiente metodología descrita en la figura 1 para estudiar el firmware de la cámara TAPO C310.

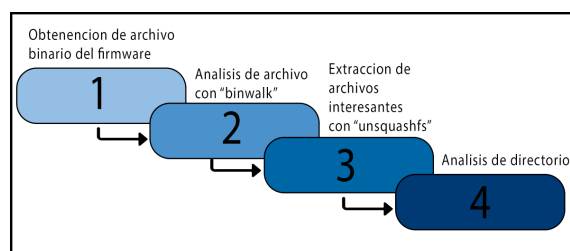


Figura 1. Metodología de análisis.

El primer paso consiste en obtener la imagen de firmware del dispositivo. En este caso, se localizó el archivo binario oficial correspondiente al modelo TAPO C310, versión 1.3.2, a través de los canales públicos del proveedor. Alternativamente, este archivo puede ser extraído físicamente desde la memoria flash del dispositivo utilizando técnicas de dumping. La imagen binaria contiene el sistema operativo embebido junto con todas las configuraciones y servicios integrados.

Una vez obtenida la imagen binaria, se procede a su análisis utilizando la herramienta *binwalk*, la cual permite identificar firmas de sistemas de archivos y cabeceras de archivos comprimidos. Esta etapa resulta fundamental para localizar estructuras que encapsulan el contenido del firmware.

Con base en la información proporcionada por *binwalk*, se hace uso de la herramienta *unsquashfs* para descomprimir el sistema de archivos. Esta operación permite acceder a la estructura de carpetas del sistema operativo del dispositivo, incluyendo directorios como */etc*, */bin*, */sbin*, y */lib*, entre otros. La extracción es esencial para realizar un análisis estático del firmware, identificando configuraciones de seguridad, credenciales por defecto y servicios habilitados.

Una vez se extrae la estructura de archivos, se realiza un análisis manual de los directorios y ficheros más críticos desde el punto de vista de la ciberseguridad.

En la figura 2 se presenta el desglose de directorios que conforman al sistema de la cámara correspondiente a la versión 1.3.2.

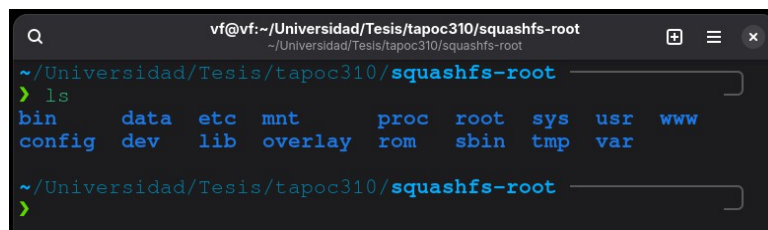


Figura 2. Directorios del sistema firmware de la cámara tapo c310.

En la tabla 2 se describe el contenido de los directorios, destacando aquellos que contienen archivos de configuración.

Tabla 2. Contenido del árbol de directorios.

Directorio	Contenido
Squashfs-root/	Raíz del proyecto
bin/	Contiene ejecutables esenciales del sistema, como busybox, ash, ping, mount, entre otros comandos estándar de UNIX.
config/	Archivos de configuración de bajo nivel. Incluye archivos como board.ini, mmap.ini, y herramientas para detección de sensores.
data/	Almacena información de configuración (cfg), paquetes (ipkg) y plugins.
dev/	Archivos de dispositivos (especiales), como graphics, necesarios para la interacción con el hardware.
etc/	Configuraciones generales del sistema como passwd, inittab, scripts de inicio, certificados, y configuraciones de red.
lib/	Bibliotecas compartidas del sistema, como libc, libbus.so, libmi_*. Son utilizadas por los ejecutables
mnt/	Punto de montaje temporal para dispositivos externos (vacío por defecto).

overlay/	Sistema de archivos en capa superior para persistencia de configuraciones y es dependiente de ejecución.
proc/	Sistema virtual que contiene información sobre procesos y kernel.
rom/	Contiene la base inalterable del sistema operativo.
root/	Directorio home del usuario root y vacío en este caso.
sbin/	Binarios del sistema relacionados con la administración como init, reboot, uci, wifi, sysctl, entre otros.
sys/	Sistema de archivos virtual que expone información y control del kernel.
tmp/	Directorio temporal utilizado por el sistema durante la ejecución
usr/	Contiene subdirectorios bin, lib, sbin, y share con utilidades adicionales y bibliotecas.
var/	Directorio para archivos temporales del sistema y sockets (ssh-*, tmpaddon, etc.).
www/	Vacío en esta versión.

En la figura 3 se muestra el uso del algoritmo RSA como método de encriptación, mostrando la posible vulnerabilidad al detectar el directorio donde son guardadas estas claves.

RSA es un sistema de cifrado asimétrico que utiliza un par de llaves: una pública para cifrar y una privada para descifrar. Su seguridad se basa en la dificultad de factorizar números primos grandes, y permite intercambiar información segura sin necesidad de compartir previamente una clave secreta. [7]

```

/home/vf/Universidad/Tesis/tapoc310/squashfs-root/etc/tp_manage # cat *.
-----BEGIN RSA PRIVATE KEY-----
MIICWwIBAAKBAQD0s2Un6S2yAQsgZ2KMPceExtFSbYpIdprq90Fdd7wc3JEVfGb8Z
PCxDFSxLXo+EYVDr/law/pfas6n2xyxJI f3Iz20nzjFmwmLa6h0bcBgd/YMR7YeX
GQy+mE6qtI7GzZcpIk3Lb+FkkC09j1R3kmVPq9e9k18/65yspmNxo9KPSQIDAQAB
AoGAQ0c+bXdTsUYbnXGPJYFYe412DeqSDcwLq1KRvnlU9dhn=Nx0ZF6s5t0uwyb
mXQtf63/xZr0tCC9t6FW0medKm4Zk6wtNQdhhpB6/SHc3SWayyM0sD5fQa1Nqb1J
bIEFev/t1J25/08IGySKgRxtZdBZR2Ta/+jtt0eU0X6VKECQQD+27HVRQ826fS
tu7TektM03gPz11tn7JsAv6dUiWjexv1r6NFEPF8nWcbzR3rfFhdh9vo4NV21Jd+
YwC6gq5AKEA7ejwURcL0cN/a5Lmnjcx3ty8/L3e/cIQ18LRDj60fVzIpyNawVK
iRdvBwoDf2OGQWYqV3RiRrLyvb7rxhChEQJAK6PdrpAK/hvonoNKR0biLP7J7/P
VHg4d1NXPQRQyoejKzY+jlv0JnuRk8Le10G+DUPkL5w8is8kw9e1ls+QJAMXjK
4FhmZ7Cbgsge1FmIjpueo08mmZ0TNlt+kvb8NKTpDKUliAwLMSQ2ybMhMseUs0daa
5WlyMvLrOMHGzX0PwQJAQc7RF2/8G+SQ7aQ18XIGxQKpJWvwHJq2fUf5tM8B+wIf
p0nLkChw49d41RldKj52LLHDx8hEEa11YrVKtaWjdg==
-----END RSA PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
MIICDCCAf2gAwIBAgIJAPV4gV2XmGvMA0GCSqGSIb3DQEBBQUAMGMxCzAJBgNV
BAYTAkNOMRIwEAYDVQQIDAlHdWUzZ2RvbmVmcxETAPBgNVBACMCFoZm56aG9uMRAw
DgYDVQQKDAduUUC1MSUSLHQ0wCwYDVQQLDART0hPMQwCgYDVQ0DDANTTFwHhcn
MTQxMTEyMDIyMTA2ZWhcNMTQxMjE5MDYyMTA2ZWhcNMTQxMjE5MDYyMDYyMDYy
A1UECAwJR3Vhbmdkb25nMREwDwYDVQQHDAhTaGVuemh1b1EQA4GA1UECgwHVFAA
TELOSzENMAzGA1UECwwEU09ITzENMAoGA1UEAwU0U09IMQI0GA0GCSqGSIb3DQEB
AQQUAA4GNADCBiQKBAQD0s2Un6S2yAQsgZ2KMPceExtFSbYpIdprq90Fdd7wc3JEV
fGb8ZPCxDFSxLXo+EYVDr/law/pfas6n2xyxJI f3Iz20nzjFmwmLa6h0bcBgd/YMR
7YeXGQy+mE6qtI7GzZcpIk3Lb+FkkC09j1R3kmVPq9e9k18/65yspmNxo9KPSQID
AQAB01AwTjAdBgNVHQ4EFgQUFjHKzR+LqrMbHyBEw0L7xn9F9dgwHwYDVR0jBBgw
FoAUFjHKzR+LqrMbHyBEw0L7xn9F9dgwDAYDVR0TBAUwAwEB/zANBgkqhkiG9w0B
AQUFAA0BgQDQ756wLon08WgRFKjZjwJR6Pbo1+eo+c+1UNTq5duXxwX6bQDMaH48
TTMnUKgoZ1A2NzYLSBjFkS3AVHcmYhW6XjQKbLj8vLVH/b7X180Cw7aNUtloPucK
i0MJMQG0jaDd001D/mLWU55Z0V8fLkove3iwQiyCmZQHgonzxQJg==
-----END CERTIFICATE-----
/home/vf/Universidad/Tesis/tapoc310/squashfs-root/etc/tp_manage #

```

Figura 3. Llave de encriptación RSA.

Posteriormente a esto se realiza un escaneo de puertos y servicios usando la herramienta *nmap*.

Tabla 3: Servicios y puertos detectados.

Puerto	Estado	Servicio
443	Open	ssl/https
554	Open	Rtsp
2020	Open	Xinupageserver?
8445	Open	ssl/https-alt?
8800	Open	Sunwebadmin?

Durante el análisis de superficie de ataque mediante escaneo de puertos, se identificaron varios servicios en ejecución en la cámara TAPO C310 mostrados en la tabla 3. El puerto 443, comúnmente utilizado para

conexiones *HTTPS* seguras, sugiere la existencia de una interfaz web o API protegida por TLS. El puerto 554, correspondiente al protocolo *RTSP (Real-Time Streaming Protocol)*, confirma la presencia de un servicio de transmisión de video en tiempo real, lo cual es esperado en dispositivos de videovigilancia. Sin embargo, la exposición del puerto 2020 asociado tentativamente a *XinuPageServer* y el 8800, potencialmente vinculado con *SunWebAdmin*, podría representar un riesgo si dichos servicios no están autenticados o correctamente configurados, ya que son menos comunes y podrían contener vulnerabilidades no documentadas. Por otro lado, el puerto 8445 está relacionado con una versión alternativa o personalizada de *HTTPS*, indicando la existencia de múltiples interfaces de administración o control [8].

La exposición de múltiples puertos en un dispositivo IoT como una cámara IP representa un área significativa de posibilidades de ataque. Cada puerto abierto puede convertirse en un vector de entrada para potenciales atacantes, especialmente si los servicios asociados no están debidamente protegidos mediante autenticación robusta, cifrado adecuado o actualizaciones de seguridad. La presencia de puertos poco comunes o no documentados incrementa aún más el riesgo, ya que pueden ocultar funcionalidades no intencionadas o interfaces administrativas olvidadas por el fabricante. Este análisis resalta la importancia de realizar un cierre o filtrado estricto de puertos, así como la necesidad de configurar los servicios activos bajo principios de mínimo privilegio y exposición.

5 Evaluación del impacto de las vulnerabilidades detectadas

Para valorar el riesgo que representan las vulnerabilidades identificadas la empresa Forum of Incident Response and Security Teams (FIRST) ofrece el sistema Common Vulnerability Scoring System Version 4.0 (CVSS v4.0), considerando factores complejidad del ataque, nivel de privilegios requeridos, impacto sobre la confidencialidad, integridad y disponibilidad como se muestra en la tabla 4.

Tabla 4: Evaluación del impacto.

Vulnerabilidad	Vector de ataque	Puntaje CVSS	Impacto Principal
Credenciales embebidas	9.3 / Crítico	ssl/https	Acceso no autorizado
Puerto RTSP sin autenticación	8.7 / Alto	Rtsp	Intercepción de video
Claves RSA almacenadas sin cifrado	5.9 / Medio	Xinupageserver?	Exposición de comunicaciones cifradas

El sistema CVSS evalúa la severidad de una vulnerabilidad con base en un conjunto de métricas conocidas como “Base Metrics”, que determinan qué tan fácil es explotarla y cuál sería el impacto en el sistema si se llegara a ejecutar. Estas métricas se dividen en dos grandes grupos: las “Exploitability Metrics”, que consideran factores como el vector de ataque (es decir, si la vulnerabilidad puede ser explotada remotamente o requiere acceso físico), la complejidad del ataque, los privilegios necesarios y si se requiere o no interacción del usuario; y las “Impact Metrics”, que evalúan si la vulnerabilidad afecta la confidencialidad, la integridad o la disponibilidad del sistema [9]. Además, en algunos escenarios se consideran métricas de impacto en sistemas subsecuentes, si la vulnerabilidad pudiera extenderse a otros dispositivos.

Las credenciales embebidas en el firmware de la cámara TAPO C310 representan una vulnerabilidad explotable remotamente, ya que un atacante con acceso a la red puede autenticarse sin dificultad. La explotación es sencilla, no requiere privilegios ni interacción del usuario, y permite comprometer la confidencialidad, integridad y disponibilidad del dispositivo, al otorgar control total sobre su configuración y funcionamiento.

La exposición del puerto RTSP sin autenticación también es explotable por red y de baja complejidad, ya que cualquier atacante puede acceder al flujo de video con herramientas básicas. No requiere privilegios ni intervención del usuario. Sin embargo, su impacto se limita a la confidencialidad, pues el atacante no puede modificar ni desactivar el sistema a través de este canal.

En cuanto a las llaves RSA visibles en el firmware, esta vulnerabilidad requiere acceso físico o administrativo al dispositivo, lo que la clasifica como local y de alta complejidad. Aunque no necesita privilegios adicionales ni interacción del usuario, su impacto recae principalmente en la confidencialidad, al permitir descifrar comunicaciones si las claves son reutilizadas, sin comprometer directamente otros aspectos del sistema.

6 Conclusiones y trabajos futuros

El análisis del firmware de la cámara ha revelado múltiples vulnerabilidades que podrían ser explotadas por actores malintencionados, comprometiendo así la seguridad de la red y la privacidad de los usuarios. A través del uso de herramientas se identificaron configuraciones inseguras, servicios expuestos y mecanismos de autenticación débiles. Estos hallazgos ponen en evidencia la necesidad de que los fabricantes adopten prácticas más rigurosas en el desarrollo de firmware para dispositivos IoT.

La actualización del firmware en dispositivos IoT es una medida para garantizar su seguridad y funcionamiento confiable. Los fabricantes suelen liberar parches que corrigen vulnerabilidades descubiertas posteriormente al lanzamiento del producto, por lo que mantener el firmware actualizado reduce el riesgo de explotación por parte de atacantes. En muchos casos, las brechas de seguridad son aprovechadas debido a que los usuarios no aplican dichas actualizaciones, dejando expuestos sus dispositivos. Además, las actualizaciones no solo mejoran la seguridad, sino que también optimizan el rendimiento, corrigen errores y amplían funcionalidades. [10]

Para futuras investigaciones, se sugiere extender el análisis a otros modelos de cámaras IP de diferentes fabricantes, con el fin de evaluar si presentan vulnerabilidades similares. Asimismo, como medida de mitigación, se propone el desarrollo de un entorno de red segura que integre segmentación de red, cifrado de extremo a extremo y supervisión del tráfico mediante sistemas IDS. Esta propuesta busca reducir significativamente el riesgo de intrusiones y accesos no autorizados, fortaleciendo así la seguridad de los entornos donde se implementan cámaras IP.

References

- [1] AXIS NetEyeTM 200. (s. f.). En Axis.com. AXIS. Recuperado 11 de abril de 2025, de <https://www.axis.com/dam/public/eb/70/f6/axis-neteyetm-200-en-US-196579.pdf>
- [2] Redes informáticas - Nociones fundamentales (5a edición). (s. f.). Google Books. <https://books.google.com.mx/books?hl=es&lr=&id=Huwy1LOPEq8C&oi=fnd&pg=PA4&dq>
- [3] =Protocolos+de+red&ots=N0-o8ubScw&sig=5mB7APr485tfdRtZV7tL5VVCX3Y&redir_esc=y#v=onepage&q=Protocolo s%20de%20red&f=false
- [4] NVD - CVE-2022-37255. (s. f.). <https://nvd.nist.gov/vuln/detail/CVE-2022-37255>
- [5] Ribero-Corzo, S. M., & Prieto-Guerrero, Y. A. (2021). Identificación de riesgos en la seguridad de la información de cámaras de vigilancia domésticas en entornos IOT. Repositorio Institucional Universidad Católica de Colombia -RIUCaC. <https://repository.ucatolica.edu.co/entities/publication/64904c63-3163-4033-87e8-f8753b49e5c8>
- [6] binwalk | Kali Linux Tools. (s. f.). Kali Linux. <https://www.kali.org/tools/binwalk/>
- [7] squashfs-tools-ng - Kali Linux Package Tracker. (s. f.). <https://pkg.kali.org/pkg/squashfs-tools-ng>
- [8] Pérez, R. A., & Gálvez, J. A. S. (2001, 1 enero). RSA: Criptosistema de llave pública . https://repositorio-uapa.cuaed.unam.mx/repositorio/moodle/pluginfile.php/2176/mod_resource/content/3/contentido/index.html
- [9] Perez, G. A. H., & Fernandez, H. A. F. (2009). GESTIÓN Y SEGURIDAD EN PUERTOS.
- [10] Vinculos, 6(1), 40-45. <https://doi.org/10.14483/2322939x.4143>
- [11] CVSS v4.0 Specification Document. (n.d.). FIRST — Forum of Incident Response and Security Teams. <https://www.first.org/cvss/v4-0/specification-document>
- [12] Jorge Alberto, M. G. (2019). LA IMPORTANCIA DE LA GESTIÓN DE RIESGOS Y SEGURIDAD EN EL INTERNET DE LAS COSAS (IOT) [Artículo, Universidad Piloto de Colombia]. <http://repository.unipiloto.edu.co/handle/20.500.12277/6754>
- [13] Mejía, D. F. (2022, 27 enero). Ana Karina Soto: Hackean cámaras de seguridad de casa de presentadora. RCN Radio. <https://www.rcnradio.com/entretenimiento/hackean-cameras-de-seguridad-de-casa-de-presentadora-ana-karina-soto>
- [14] López Rodríguez, J. C. L. (2007). Estructura, funcionamiento y aplicación de las cámaras IP [Monografía]. Universidad Autónoma del estado de Hidalgo.

- [15] Bernal Garnica, M. B. G. (2015). SEGURIDAD EN PROFUNDIDAD “Los pro y los contra” [Universitaria Piloto de Colombia]. <https://repository.unipiloto.edu.co/bitstream/handle/20.500.12277/2832/Trabajo%20de%20grado.pdf?sequence=1&isAllowed=y>
- [16] ¿Qué es la segmentación de la red? (s. f.). Palo Alto Networks. <https://www.paloaltonetworks.es/cyberpedia/what-is-network-segmentation> (accedido el 6 de abril de 2025).
- [17] ¿Qué es una lista de control de acceso a la red (ACL)? | Fortinet. (s. f.). Fortinet. <https://www.fortinet.com/lat/resources/cyberglossary/network-access-control-list> (accedido el 6 de abril de 2025).
- [18] ¿Qué es un firewall? Definición y tipos de firewall | Fortinet. (s. f.). Fortinet. <https://www.fortinet.com/lat/resources/cyberglossary/firewall#:~:text=Un%20firewall%20es%20un%20dispositivo,y%20redes%20externas%20no%20confiables.> (accedido el 6 de abril de 2025).
- [19] %20un%20dispositivo,y%20redes%20externas%20no%20confiables. (accedido el 6 de abril de 2025).
- [20] Ibm. (2024, 15 julio). Sistema de detección de intrusiones. ¿Qué es un sistema de detección de intrusiones (IDS)? <https://www.ibm.com/mx-es/topics/intrusion-detection-system> (accedido el 6 de abril de 2025).
- [21] Redacción. (2019, 13 diciembre). El hacker que le habló a una niña por una cámara Ring colocada en su habitación. BBC News Mundo. <https://www.bbc.com/mundo/noticias-50781444>