RESULTADOS DEL ANÁLISIS DE VULNERABILIDADES Y PRUEBAS DE SEGURIDAD - EH

DOCUMENTO CONFIDENCIAL

Elaborado por:

Versión 1.0

Febrero, 2026 Bogotá D.C

Historial de versiones

Id	Fecha (dd/mm/aa)	Versión	Descripción	Elaborado por	Revisado por
1	25/02/2026	1.0	Resultados de las pruebas de seguridad a los activos web y aplicación móvil de TRANSMILENIO S.A	Rodrigo Diaz	Password Consulting Services S.A.S

TABLA DE CONTENIDO

1. INTRODUCCIÓN....3
2. OBJETIVO....3
3. ALCANCE....3
4. DEFINICIONES....4
5. CONTENIDO....5
5.1 Metodología a Utilizar 5
5.1.1. FASE 1. Identificación de vulnerabilidades puertos y servicios....6
5.1.2. FASE 2. Análisis técnico de vulnerabilidades....6
5.1.3. FASE 3. Análisis de vulnerabilidades asistido con Inteligencia Artificial....8
5.1.4. FASE 4. Pruebas de seguridad (Hacking Ético)....9
5.1.5. FASE 5. Generación y socialización de resultados....10
5.1.6. FASE 6. Verificación final del estado de las vulnerabilidades (Retest)....10
5.2 Criterios de clasificación de vulnerabilidades 11
5.3 Tiempos de remediación 12
5.4 Herramientas para análisis de vulnerabilidades 12
6. RESULTADOS....13
6.1 Resultados del análisis de vulnerabilidades de infraestructura local 13
6.2 Resultados del análisis de vulnerabilidades de infraestructura web 15
6.3 Resultados del análisis de vulnerabilidades de la aplicación móvil 16
7.Pruebas de seguridad....18
7.1 Resultados de las pruebas de seguridad en infraestructura local 18
7.1.1 SMB Signing not required y envenenamiento LLMNR/NBT-NS – No exitosa....18
7.1.2 Enumeración de Sesiones Nulas (Null Session) y Auditoría de SMB Anónimo – No exitosa.19
7.1.3 IPv6 DNS Takeover (mitm6) – No exitosa....21
7.2 Resultados de las pruebas de seguridad en aplicaciones web 22
7.2.1 Inyecciones SQL – No exitosa....22
7.2.2 Cross-Site Scripting – No exitosa....23
7.2.23 Hallazgos Proactivos / Fuera de Alcance (Out-of-Scope Insights)....24
7.3 Resultados de las pruebas de seguridad en TransmiApp 28
8. CONCLUSIONES....32
9. RECOMENDACIONES....32

1. INTRODUCCIÓN

La fase de Hacking Ético ha finalizado con algunos hallazgos importantes que requieren revisión y ajustes en algunos casos, en la revisión de la aplicación móvil también se observaron algunas oportunidades de mejora, a nivel de equipos de la red local no se observaron debilidades de alto riesgo.

2. OBJETIVO

Presentar los resultados de estas pruebas y relacionar todas las recomendaciones de seguridad pertinentes para llevar al mínimo posible la superficie de ataque que se documento en esta fase del proyecto de seguridad para TRANSMILENIO.

3. ALCANCE

Comprende pruebas éticas de manera manual para corroborar los resultados de la fase de análisis (fase anterior) y el uso de herramientas que faciliten el análisis de comportamiento de las aplicaciones ante diferentes situaciones.

Ataque a realizar	Objetivo del ataque
Ataque hombre en el medio "MitM" (SMB Signing not required)	Interceptación de tráfico SMB para ejecutar comandos en hosts remotos sin conocer contraseñas.
IPv6 DNS Takeover (mitm6)	Suplantación de servicios DNS para forzar la autenticación de máquinas Windows hacia el atacante.
Envenenamiento LLMNR/NBT-NS	Intercepción de solicitudes de resolución de nombres fallidas para capturar hashes de usuarios.
Enumeración de Sesiones Nulas	Extracción de información crítica del dominio (usuarios, grupos, políticas) sin autenticación.
Auditoría de SMB Anónimo	Búsqueda de recursos compartidos abiertos al grupo "Everyone" o "Guest" con datos sensibles.
SQL Injection	Búsqueda de posibles puntos inyección de sentencias maliciosas
Cross-Site Scripting XSS	Intentos de inyección Javascript para confirmar o descartar vulnerabilidad de XSS

Tabla 1. Plan de pruebas a realizar

4. DEFINICIONES

- **Activo de Información:** Cualquier elemento que contenga, genere, adquiera, gestione y/o procese información, que tiene valor para uno o más procesos de la organización y debe protegerse (ISO/IEC 27001:2013).

- **Vulnerabilidad:** Es una debilidad o fallo en un sistema de información que pone en riesgo la seguridad de la información, pudiendo permitir que un atacante logre comprometer la integridad, disponibilidad o confidencialidad de la misma.

- **SAST (Static Application Security Testing):** Metodología de análisis de seguridad estático que examina el código fuente de las aplicaciones en busca de debilidades de programación, fallos lógicos o malas prácticas de desarrollo sin necesidad de ejecutar el programa.

- **SCA (Software Composition Analysis):** Proceso de gestión y análisis de componentes de código abierto y dependencias de terceros. Su objetivo es identificar librerías obsoletas, vulnerabilidades conocidas (CVE) y riesgos de licencias en el árbol de dependencias del software.

- **Severidad:** Clasificación del nivel de riesgo asociado a una vulnerabilidad (Baja, Media, Alta o Crítica), basada generalmente en el impacto potencial y la facilidad de explotación.

- **Remediación:** Conjunto de acciones técnicas recomendadas, como la aplicación de parches o la actualización de versiones de paquetes, destinadas a corregir o mitigar una vulnerabilidad identificada.

- **Ethical Hacking:** Pruebas de seguridad que permiten poner en evidencia la presencia de vulnerabilidades sobre uno o varios activos de información mediante el uso de técnicas y herramientas controladas.

- **Retest:** Fase de verificación que implica volver a validar las vulnerabilidades previamente identificadas después de que hayan sido reportadas como corregidas por el equipo de desarrollo, con el fin de confirmar la eficacia de la solución implementada.

5. CONTENIDO

En este numeral se revisan las herramientas a utilizar y la metodología que se aplicará a los activos de información descritos en el alcance, mencionando una a una las fases y entregables que se deben generar como resultado de cada fase:

Nota: La presente metodología consigna las actividades metodológicas para ambas pruebas, tanto las pruebas de vulnerabilidades como las pruebas de hacking ético. Lo anterior dado que las pruebas de hacking ético permiten comprobar la existencia de vulnerabilidades a través de las pruebas de hacking ético se realiza la validación de falsos positivos sobre la información entregada por las pruebas de vulnerabilidad.

Por lo anterior, las pruebas de vulnerabilidad se convierten en un insumo indispensable para planear las pruebas de hacking ético y dichas pruebas de hacking ético corroboran los resultados de las vulnerabilidades con duda que fueron identificadas en las pruebas de vulnerabilidad. Esto mantiene un vínculo entre ambas pruebas y por consiguiente la relación y e intersección en las metodologías.

A continuación, se presenta la metodología que se aplica para cubrir ambos tipos de pruebas:

5.1 Metodología a Utilizar

A continuación, se presenta la metodología OSSTMM (Open Source Security Testing Methodology Manual) y OWASP (Open Web Application Security Project), todas en su última versión, y de las cual se derivan las siguientes actividades:



Figura 1. Metodología a utilizar de pruebas de vulnerabilidad y Ethical hacking

5.1.1. FASE 1. Identificación de vulnerabilidades puertos y servicios.

Objetivo

Establecer la superficie de ataque inicial del objetivo. Esta fase se centra en el descubrimiento y enumeración de todos los activos accesibles (hosts, dominios, aplicaciones), recopilando información clave como puertos abiertos, servicios activos y las versiones de software y tecnologías expuestas. El fin es crear un mapa detallado del entorno para identificar vectores de entrada potenciales y puntos de interés para las siguientes fases.

Alcance

Se centra en el **reconocimiento activo** (escaneo de puertos, sondeo de servicios) sobre los activos definidos en el alcance del ejercicio. Incluye el uso de herramientas automatizadas para una detección inicial de vulnerabilidades conocidas (ej. CVEs en versiones de software expuestas), pero *no* incluye la explotación o validación manual de dichas vulnerabilidades.

Actividades

- **Escaneo y Enumeración de Red (Scanning):** Identificación de hosts activos, escaneo de puertos (TCP/UDP) para descubrir servicios expuestos (ej. SMB, SMTP, SSH, HTTP, RDP).

- **Fingerprinting de Servicios y Versiones:** Determinación precisa de las tecnologías, sistemas operativos, aplicaciones y versiones de software en ejecución en cada puerto abierto.

- **Escaneo Automatizado de Vulnerabilidades:** Ejecución de escáneres de vulnerabilidades (como Nmap scripts, Nessus, Nikto, etc.) para identificar configuraciones incorrectas comunes y vulnerabilidades "de fruta madura" (low-hanging fruit) basadas en las versiones detectadas.

5.1.2. FASE 2. Análisis técnico de vulnerabilidades

Objetivo

Validar, depurar y categorizar los hallazgos de la Fase 1. El objetivo principal es aplicar el criterio y la experiencia de los analistas de Password para confirmar la existencia real de las vulnerabilidades reportadas por las herramientas automatizadas, descartar los

falsos positivos y realizar una evaluación inicial del riesgo de cada hallazgo confirmado. Esta fase filtra el "ruido" y enfoca los esfuerzos de las fases posteriores únicamente en amenazas verificadas.

Alcance

Esta limitado a la revisión y validación de los datos recolectados. Incluye la inspección manual, pero no intrusiva, de servicios y aplicaciones para confirmar o desmentir los hallazgos. No se realizan intentos de explotación, sino de corroboración (ej. verificar banners manualmente, probar configuraciones por defecto conocidas, comparar versiones exactas con bases de datos de exploits).

Actividades

- **Consolidación y Correlación de Hallazgos:** Agrupar y cruzar los resultados de las distintas herramientas de la Fase 1 (escáneres de puertos y de vulnerabilidades) para obtener una vista unificada de los activos y sus debilidades potenciales.

- **Verificación Manual y Descarte de Falsos Positivos:** Análisis crítico de cada vulnerabilidad reportada. Se realizan comprobaciones manuales (ej. revisión de cabeceras HTTP, conexión a servicios para verificar banners, análisis de respuestas de error) para confirmar que la vulnerabilidad es real y no un error del escáner.

- **Investigación y Contextualización de Vulnerabilidades:** Búsqueda de información pública (bases de datos de CVEs, Exploit-DB, reportes de seguridad) sobre las vulnerabilidades *confirmadas* para entender su impacto potencial, su severidad y la existencia de pruebas de concepto (PoCs) o exploits públicos.

- **Priorización Inicial de Hallazgos:** Clasificación de las vulnerabilidades validadas según su severidad (Crítica, Alta, Media, Baja) y su aparente explotabilidad, creando una lista depurada que servirá como entrada para las siguientes fases.

5.1.3. FASE 3. Análisis de vulnerabilidades asistido con Inteligencia Artificial

Objetivo

Enriquecer el análisis tradicional de vulnerabilidades (Fase 2) utilizando modelos de Inteligencia Artificial. El objetivo es identificar patrones complejos, correlacionar hallazgos aparentemente aislados y descubrir vectores de ataque no evidentes que las herramientas convencionales y el análisis manual inicial podrían pasar por alto. Esta fase busca optimizar la priorización de objetivos para la fase de Hacking Ético, enfocando los esfuerzos en las amenazas más probables y de mayor impacto.

Alcance

Aplicación de herramientas basadas en IA (como LLMs entrenados en seguridad, plataformas de análisis predictivo o correlación avanzada) sobre el conjunto de vulnerabilidades *ya validadas* en la Fase 2. Incluye el análisis de código (SAST) asistido por IA (si se dispone del código), la generación de hipótesis de cadenas de ataque (attack chains) y la evaluación contextual de la explotabilidad de un fallo.

Actividades

- **Correlación Avanzada de Hallazgos:** Utilización de IA para cruzar datos de todos los servicios y aplicaciones analizadas, identificando cómo vulnerabilidades de bajo impacto en diferentes sistemas podrían ser encadenadas para lograr un impacto crítico.

- **Análisis Predictivo y de Lógica de Negocio:** Aplicación de modelos de IA para analizar el comportamiento de la aplicación o revisar código fuente en busca de fallos de lógica de negocio (ej. secuencias de pasos inseguras) que no se basan en firmas de CVEs conocidas (este punto aplica solo en casos específicos).

- **Generación de Pruebas de Concepto (PoC):** Asistencia de la IA para generar borradores de *scripts* o *payloads* de prueba de concepto (PoC) adaptados al entorno y tecnología específicos del objetivo, agilizando la preparación para la fase de explotación.

- **Priorización de Vectores de Ataque:** Re-evaluación de la lista de vulnerabilidades priorizadas, esta vez clasificándolas no solo por su severidad (CVSS) sino por su *probabilidad real de explotación* en el contexto del sistema evaluado, según el análisis de la IA.

5.1.4. FASE 4. Pruebas de seguridad (Hacking Ético)

Objetivo

Demostrar el impacto real y tangible de las vulnerabilidades identificadas. El objetivo de esta fase es la explotación controlada de los vectores de ataque priorizados para obtener acceso no autorizado, escalar privilegios o acceder a información sensible. A diferencia de las fases anteriores de *análisis*, esta fase se centra en la *ejecución* activa de ataques simulados para validar el riesgo.

Alcance

Se enfoca estrictamente a la explotación activa y controlada de las vulnerabilidades confirmadas. Se ejecutan pruebas de penetración siguiendo las reglas de enfrentamiento (Rules of Engagement - RoE) acordadas, con el fin de evitar interrupciones de servicio no deseadas. **El alcance incluye tanto la explotación manual como la asistida por IA** para la adaptación de *payloads* o la evasión de defensas.

Actividades

- **Explotación de Vulnerabilidades (Exploitation):** Ejecución de pruebas de concepto (PoCs) y *exploits* (públicos o modificados) contra los servicios, aplicaciones y configuraciones vulnerables para ganar un acceso inicial (ej. Remote Code Execution, SQL Injection, Server-Side Request Forgery).

- **Post-Explotación:** Una vez obtenido un acceso, se intentan técnicas para consolidar la persistencia, escalar privilegios (vertical y horizontalmente) y realizar movimientos laterales (si el alcance lo permite) para identificar el alcance máximo del compromiso.

- **Adaptación de Cargas Útiles (Payloads) con IA:** Uso de herramientas de IA para modificar y ofuscar *payloads* de explotación, con el objetivo de evadir sistemas de detección como Antivirus (AV) o Web Application Firewalls (WAF).

- **Exfiltración de Pruebas (Data Exfiltration):** Búsqueda y extracción controlada de información sensible (como "banderas" o *flags*, archivos de configuración, extractos de bases de datos) que sirva como evidencia irrefutable del impacto del ataque.

- **Documentación Rigurosa de Evidencia:** Registro detallado paso a paso (con capturas de pantalla, comandos y *scripts* utilizados) de cada explotación exitosa, demostrando la cadena de ataque (attack chain) completa.

5.1.5. FASE 5. Generación y socialización de resultados

Objetivo

Consolidar un único documento que contenga los resultados de ambas fases, relacionando todos los hallazgos y priorizando los de mayor riesgo. Adicionalmente, generar una serie de recomendaciones que permita cerrar esos hallazgos o contenerlos.

Alcance

Cubre la generación y entrega del informe y su sesión de socialización que sirva para dar a conocer el detalle de los hallazgos y aclarar posibles dudas.

5.1.6. FASE 6. Verificación final del estado de las vulnerabilidades (Retest)

Objetivo

Validar y verificar la efectividad de las acciones de remediación implementadas por TRANSMILENIO. El objetivo principal es confirmar que las vulnerabilidades reportadas en la Fase 5 han sido corregidas, mitigadas o gestionadas adecuadamente, asegurando que los vectores de ataque (demostrados en la Fase 4) ya no son viables y que el riesgo ha sido reducido a un nivel aceptable.

Alcance

Está estrictamente limitado a la Re-evaluación de los hallazgos documentados en el informe final. No constituye una nueva prueba de penetración completa, sino una auditoría de validación enfocada. Las pruebas se dirigen únicamente a los sistemas, parámetros y vectores previamente identificados como vulnerables para confirmar su estado de remediación.

Actividades

- **Planificación del Retest:** TRANSMILENIO informa a Password que han ejecutado satisfactoriamente todo el proceso de remediación y están listos para

ejecutar nuevamente esta fase de escaneos automatizados con los respectivos análisis manuales y asistidos por IA.

- **Ejecución de Pruebas de Verificación:** Re-ejecución metódica de las Pruebas de Concepto (PoC) y cadenas de ataque (attack chains) que fueron exitosas en la Fase 4, con el fin de confirmar si la explotación sigue siendo posible.

- **Análisis del Estado de Remediación:** Evaluación de los resultados de las pruebas para determinar si cada vulnerabilidad está:

    - **Corregida:** La vulnerabilidad ha sido erradicada desde la raíz.
    - **Mitigada:** La vulnerabilidad de raíz puede persistir, pero un control compensatorio (como un WAF) ahora bloquea efectivamente su explotación.
    - **No Corregida:** La vulnerabilidad sigue siendo explotable.

- **Generación de Informe de Retest:** Creación de un informe de seguimiento (Versión 2 del informe técnico) que actualiza el estado de cada vulnerabilidad reportada (Persistente, Remediada o Nueva en el caso de que se introduzcan nuevas vulnerabilidades).

5.2 Criterios de clasificación de vulnerabilidades

Las vulnerabilidades se clasifican de acuerdo con los siguientes niveles:

Nivel	Descripción
Crítico o Extremo	La explotación de una vulnerabilidad con nivel Extremo podría proporcionar acceso a datos y sistemas no autorizados, a un nivel de administración. El riesgo contempla la exposición de información sensible, tal como identificadores de usuario (nombres), contraseñas, información propietaria, secretos, números de tarjetas de crédito, información de clientes, de colaboradores u otra información sensible de la organización. Así mismo, podría llegar a generar Denegación de Servicio que impacte de manera importante la continuidad de las operaciones.
Alto	La explotación de una vulnerabilidad con nivel Alto podría proporcionar acceso a datos y sistemas no autorizados, en la mayoría de los casos a un nivel administrativo o usuario avanzado.

Nivel	Descripción
Medio	La explotación de vulnerabilidades con nivel Medio podría permitir indirectamente acceso a archivos de configuración, datos, o afectar parcialmente un sistema de información.
Bajo	Estas vulnerabilidades son de tipo informativo, por ejemplo, las configuraciones de puertos que tiene habilitados y servicios corriendo.

Tabla 2. Niveles de severidad vulnerabilidades

5.3 Tiempos de remediación

Los tiempos de remediación se recomiendan con base en la experiencia de otras entidades a partir de la siguiente descripción.

- **Corto Plazo:** La vulnerabilidad debe ser corregida inmediato o antes de 30 días.
- **Mediano Plazo**: La vulnerabilidad debe ser corregida antes 60 días.
- **Largo Plazo:** La vulnerabilidad puede ser corregida antes de 90 días.

Tipo	Tiempo de corrección (Meses)	Nivel de Criticidad
Corto plazo	Inmediato	Extremo
	30 días	Alto
Mediano plazo	60 días	Medio
Largo plazo	90 días	Bajo

Tabla 3. Tiempos sugeridos de remediación.

5.4 Herramientas para análisis de vulnerabilidades

a. **Burp Suite** (portswigger.net): Herramienta para la detección manual y automática de vulnerabilidades de aplicaciones web.

b. **OWASP ZAP** (www.owasp.org): Herramienta utilizada para el escaneo de vulnerabilidades de portales web.

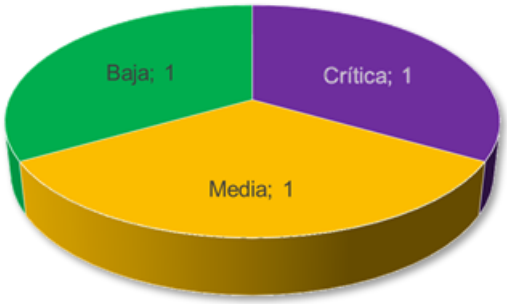
c. **Nmap** (nmap.org): Herramienta para el descubrimiento de redes y la auditoría de seguridad.

d. **MobSF** (github.com): Herramienta para análisis de vulnerabilidades dinámico y estático de aplicaciones móviles.
e. **Drozer** (github.com): Herramienta para exploración de superficie de ataque y explotación de brechas de seguridad móvil a bajo nivel.

6. RESULTADOS

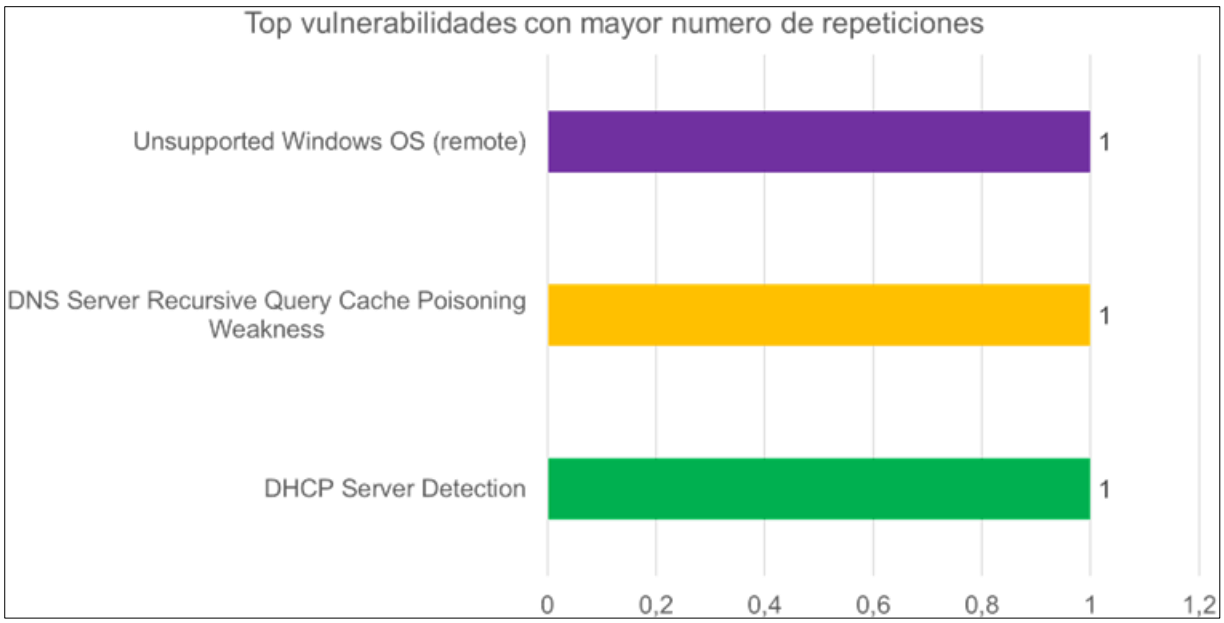
En este numeral se agrupan los resultados por los dos componentes evaluados, vulnerabilidades a nivel de código y a nivel de dependencias invocadas en las funcionalidades de la aplicación:

6.1 Resultados del análisis de vulnerabilidades de infraestructura local



Gráfica 1. Resultado general del análisis de vulnerabilidades de la infraestructura local.

Resumen: A nivel de infraestructura local no hubo detecciones de alto riesgo, excepto por la detección de severidad crítica que corresponde a un sistema Windows Server 2012 (de acuerdo con la detección) que no cuenta actualmente con soporte por parte del fabricante, por lo demás no se observaron debilidades o superficie de ataque de alto riesgo en cuanto a este primer alcance.



Gráfica 2. Detalle de vulnerabilidades detectadas por cantidad.

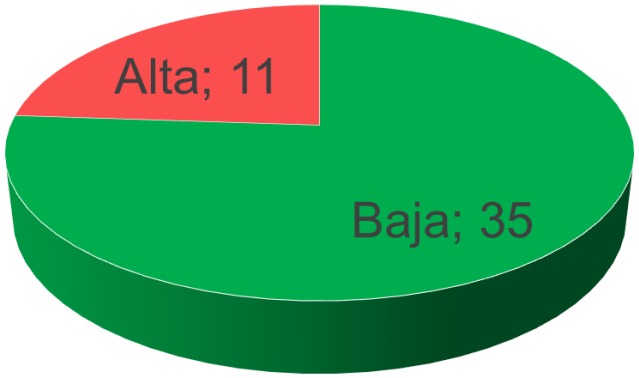
A continuación, se describen las tres principales debilidades identificadas en el gráfico:

- **Unsupported Windows OS (remote) (1 hallazgo):** Esta vulnerabilidad indica la presencia de sistemas operativos Windows que han alcanzado su fin de vida útil y ya no reciben actualizaciones de seguridad por parte del fabricante. Al carecer de parches para nuevas amenazas, estos activos son blancos críticos para ataques de ejecución remota de código o escalamiento de privilegios, representando un riesgo significativo para la integridad de la red.
- **DNS Server Recursive Query Cache Poisoning Weakness (1 hallazgo):** Esta debilidad se presenta cuando un servidor DNS permite consultas recursivas de fuentes no autorizadas, lo que facilita ataques de envenenamiento de caché. Un atacante podría falsificar registros DNS para redirigir el tráfico legítimo hacia servidores maliciosos, permitiendo el robo de credenciales o la interceptación de comunicaciones sensibles mediante técnicas de *Man-in-the-Middle*.
- **DHCP Server Detection (1 hallazgo):** La detección de un servidor DHCP en el segmento de red analizado resalta la disponibilidad de este servicio para asignar direcciones IP de forma automática. Aunque es una función operativa, si no está debidamente asegurado o si se trata de un servidor "rogue" (no autorizado), puede ser aprovechado por un atacante para realizar ataques de denegación de

servicio (DoS) o para actuar como puerta de enlace predeterminada y capturar el tráfico de la red.

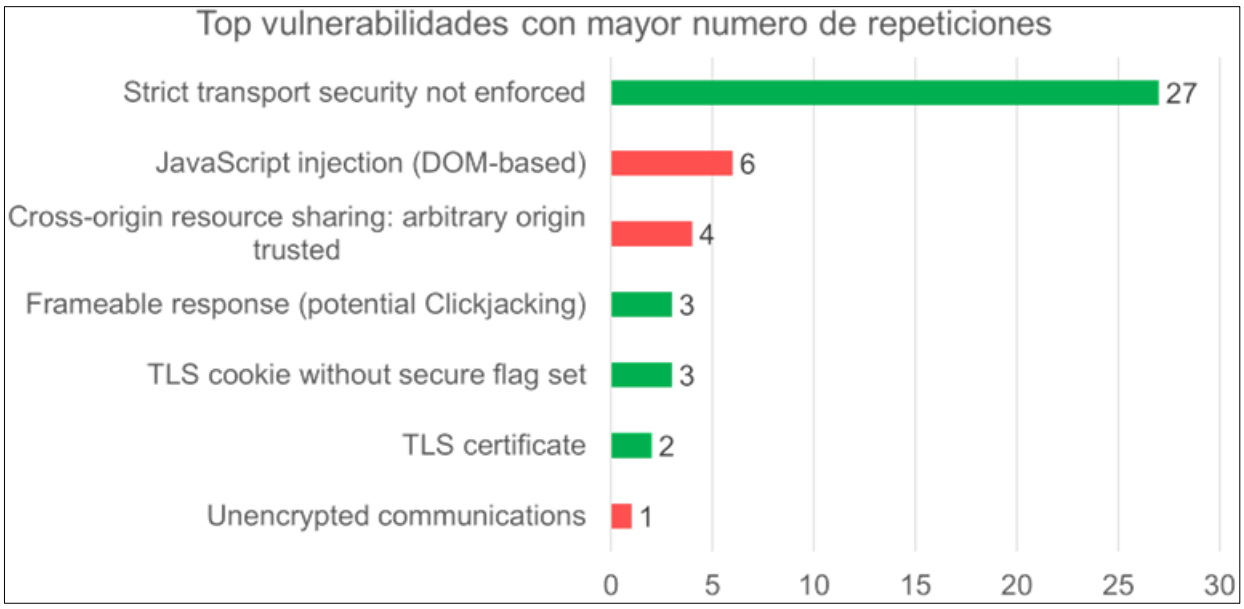
6.2 Resultados del análisis de vulnerabilidades de infraestructura web

A nivel de portales web, se evaluaron un total de 45 direcciones URL, solo 36 de ellas respondieron, las otras 9 no mostraron ningún tipo de respuesta en el momento de las verificaciones:



Gráfica 3. Detección general de las vulnerabilidades a nivel de portales web

Resumen: El análisis de seguridad en los activos web reveló una postura de riesgo centrada en la exposición de datos y la debilidad en los controles de transporte, donde la falta de políticas estrictas de cifrado (HSTS) y el uso de comunicaciones no cifradas podrían permitir la interceptación de tráfico sensible. Adicionalmente, se identificaron fallos críticos como la inyección de JavaScript (DOM-based) y configuraciones permisivas en CORS que habilitarían a un atacante para ejecutar instrucciones maliciosas, suplantar identidades o realizar ataques de Clickjacking, comprometiendo directamente la integridad de las sesiones de los usuarios y la confidencialidad de la información corporativa.

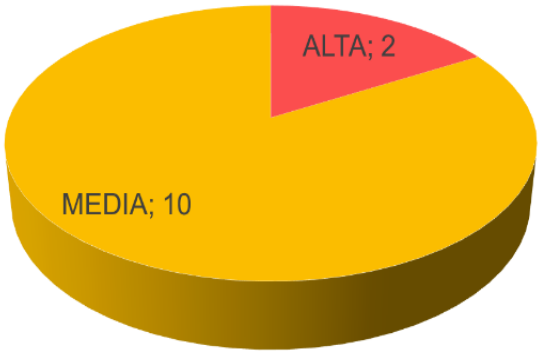


Gráfica 4. Vulnerabilidades detectadas en los portales evaluados

Algunas de las vulnerabilidades detectadas están presentes en el mismo portal por lo que, aunque el número pueda parecer alto, aplicando la solución de raíz se van a poder cerrar varias detecciones con un mismo movimiento, el numero mas alto corresponde a una cabecera de seguridad que se puede corregir fácilmente sin mayor dificultad.

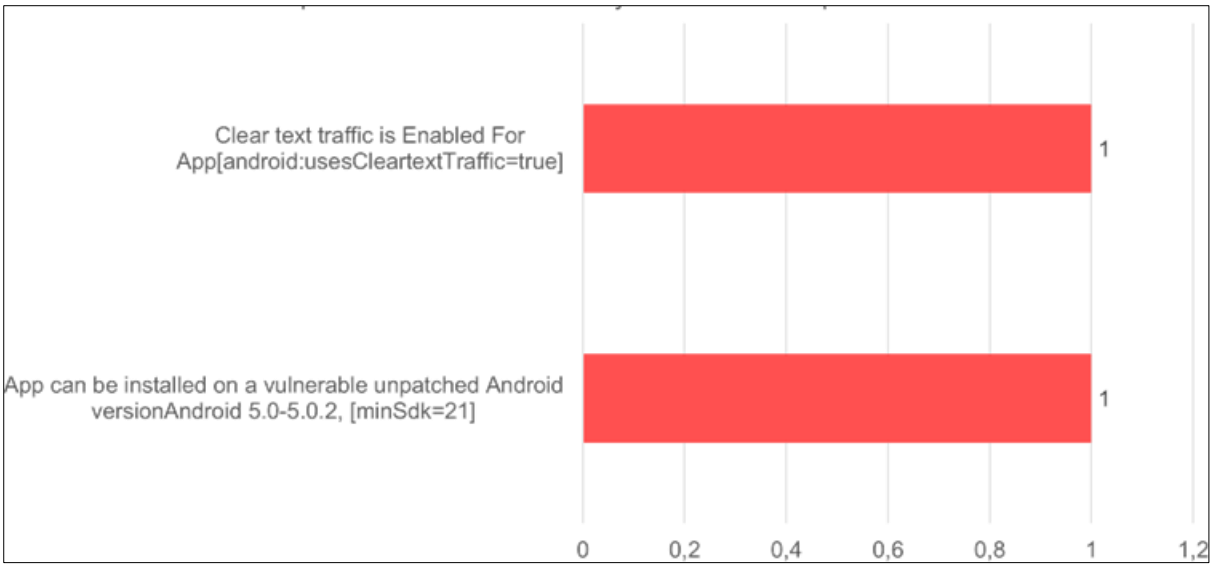
6.3 Resultados del análisis de vulnerabilidades de la aplicación móvil

En la aplicación móvil se observaron algunas debilidades en la configuración del archivo “Manifest” que podrían permitir comportamientos inusuales en la aplicación



Gráfica 5. Cantidad de vulnerabilidades en TransmiApp distribuidas por severidad

Resumen: El análisis de seguridad reveló una superficie de ataque crítica debido a configuraciones permisivas en el “Manifest” y controles de transporte deficientes. La habilitación de tráfico en texto claro y el soporte para versiones de Android obsoletas facilitan la interceptación de datos y la explotación de vulnerabilidades de sistema. Asimismo, la presencia de múltiples componentes exportados sin restricciones de acceso y la falta de políticas de cifrado estricto comprometen la confidencialidad de la información y permiten la manipulación de flujos de autenticación por parte de aplicaciones maliciosas.



Gráfica 6. Configuraciones clasificadas en severidad alta

- **Cleartext Traffic is Enabled:** Esta debilidad en la configuración permite la transmisión de datos sin cifrado (HTTP), lo que facultaría a un atacante posicionado en la misma red para realizar ataques de interceptación (Sniffing) y **Man-in-the-Middle (MITM)**. Esto compromete la confidencialidad de la información del usuario y permite la manipulación de las respuestas del servidor en tiempo real.

- **App can be installed on a vulnerable unpatched Android version:** Al permitir la ejecución en versiones obsoletas del sistema operativo (API 21), la aplicación queda expuesta a un entorno sin parches de seguridad modernos. Un atacante podría aprovechar **vulnerabilidades conocidas del kernel o del sistema** (como escalada de privilegios) para romper el aislamiento de la app (*sandboxing*) y extraer datos sensibles que, en versiones modernas, estarían protegidos por hardware.

7.Pruebas de seguridad

Las pruebas de hacking ético han revelado importantes oportunidades de mejora que no se observaron con claridad durante la fase de análisis, los resultados de agrupan en 3 numerales para separar los hallazgos a nivel de infraestructura local, de sitios web y de aplicación TransmiApp.

7.1 Resultados de las pruebas de seguridad en infraestructura local

En este frente, se confirmó el resultado del análisis de vulnerabilidades que indicó detecciones mínimas, se realizaron algunas pruebas adicionales a vulnerabilidades clásicas de estos sistemas y estos fueron los resultados.

7.1.1 SMB Signing not required y envenenamiento LLMNR/NBT-NS – No exitosa

En esta Prueba, a pesar de que se encontraron algunos activos con esta configuración incorrecta, no se logro ejecutar el ataque tipo relay.

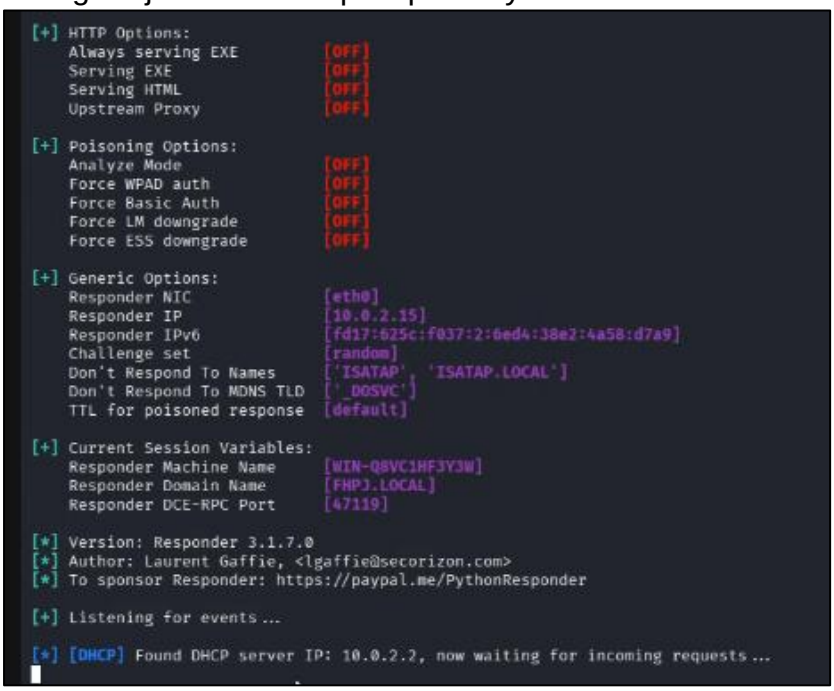


Figura 2. Ejecución del responder para envenenamiento de tráfico – No exitoso, no se observo envío de paquetes envenenados

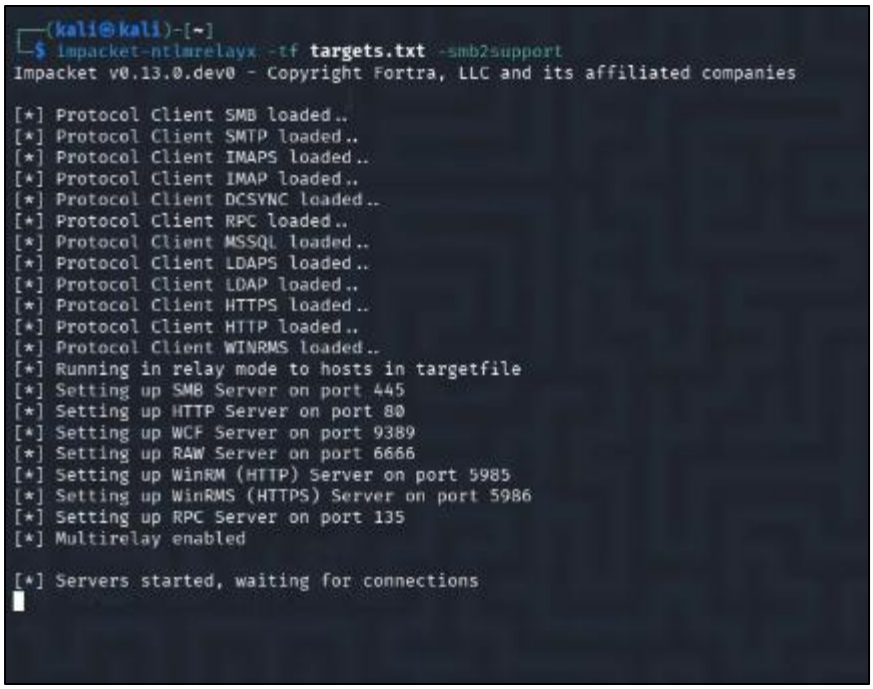


Figura 3. Ataque relay a los activos detectados con esta debilidad – No exitoso, no se logró interceptar tráfico SMB

En la fase de análisis no se observaron equipos con esta debilidad, sin embargo, se ejecuto un escaneo en varias de las redes locales completas y como resultado, algunos tenían el "signing" con valor "False" además de "SMBv1 True" lo que hacia viable la prueba, en este caso, como lo muestran las figuras anteriores, no se observo ninguna acción, es bastante probable que el firewall tenga protección ante ataques de "relay" dentro de las redes locales; con este resultado no se consideró necesario ampliar la prueba.

7.1.2 Enumeración de Sesiones Nulas (Null Session) y Auditoría de SMB Anónimo – No exitosa

En cuanto a la enumeracion de usuarios o recursos mediante la técnica de sesiones nulas, tampoco se obtuvo resultados favorables en términos de ataque, es decir, la configuración actual que se probó, no permitió ejecutar este tipo de ataque.



Figura 4. Enumeración de usuarios mediante cliente rpc – No exitosa, el resultado fue acceso denegado

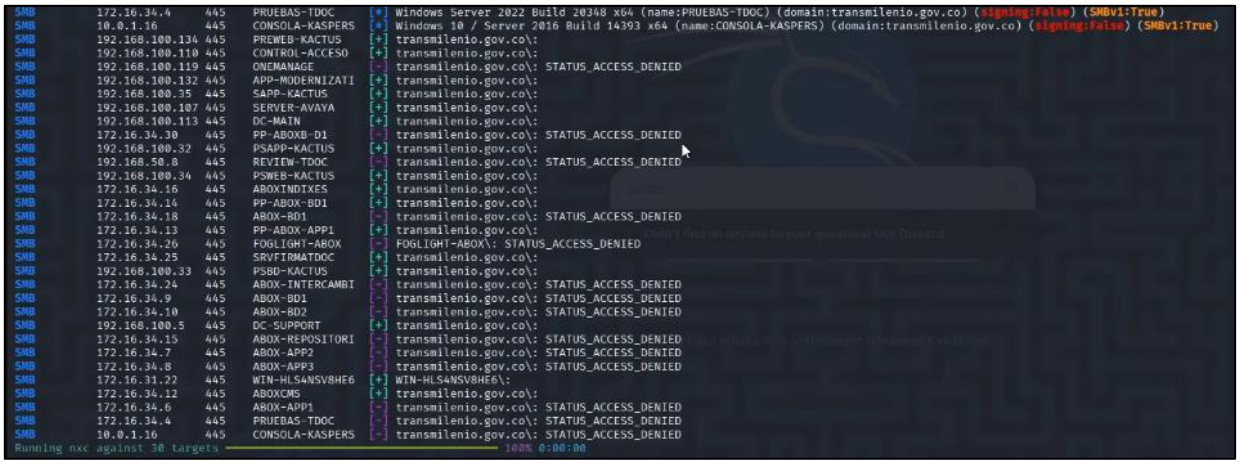


Figura 5. Enumeración de usuarios con la herramienta nxc – No exitoso, a pesar de que algunos parecen dar positivo, ninguno permitió hacer una enumeración efectiva

Como se observa en las figuras anteriores, no se logró hacer ningún tipo de enumeración ni de usuarios ni de recursos compartidos, la respuesta a todos estos intentos fue "Acceso denegado" lo que indica una configuración correcta en ese sentido.

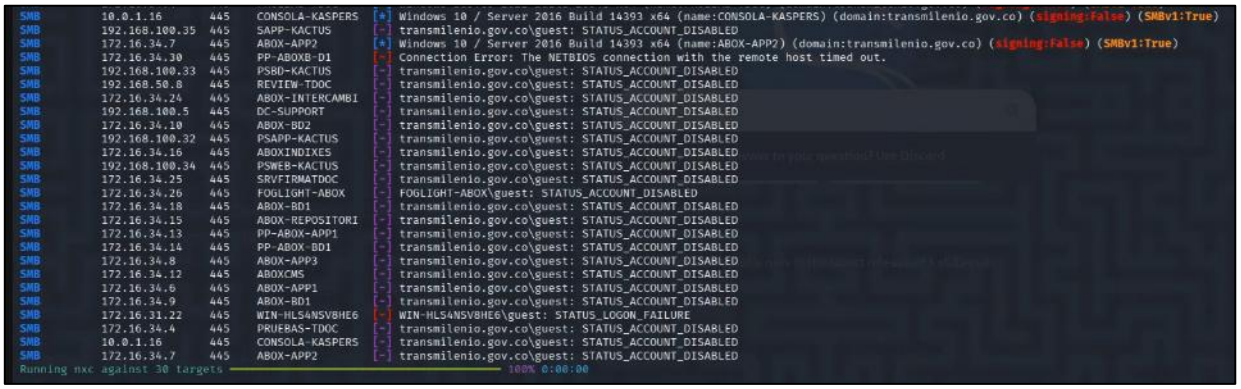


Figura 6. Inicio de sesión automatizado con el usuario "guest" o invitado con nxc – No exitoso, el usuario fue deshabilitado

También se intento hacer enumeraciones con el usuario invitado (guest), sin embargo, esta cuenta de usuario se encuentra deshabilitada como se observa en la figura 6, por lo que con estas pruebas se descarta la posibilidad de acceder a alguno de estos activos usando está reconocida técnica.

7.1.3 IPv6 DNS Takeover (mitm6) – No exitosa

Se intenta utilizar mitm6 para tomar control del tráfico mediante la suplantación de servicios DNS por IPv6. Sin embargo, al estar el atacante en una **VLAN distinta**, el tráfico de difusión (broadcast) no alcanza al objetivo, lo que obligó a pivotar hacia un ataque unicast dirigido.

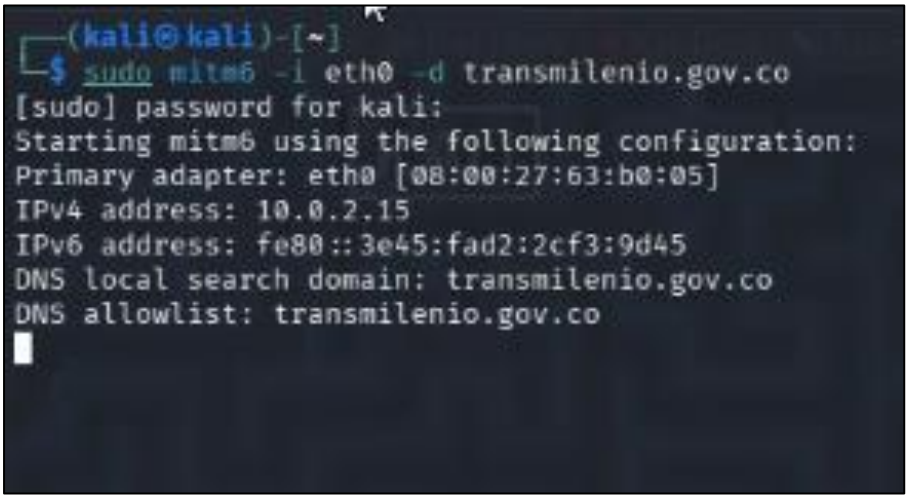


Figura 7. Intento de toma de tráfico con mimt6 – No exitoso, no hay alcance a la VLAN

Se ejecuta script personalizado dnd.py mediante Python 3. Este script pretende realizar una **inyección dirigida** de paquetes UDP al puerto 53 del Cisco. La salida en pantalla (representada por los puntos continuos) indica el envío masivo y constante de respuestas DNS falsas, buscando que el dispositivo acepte la dirección IP del atacante (10.0.2.15) como válida para el dominio transmilenio.gov.co antes de recibir la respuesta legítima.



Figura 8. Ejecución de script personalizado para direccionar paquetes



Figura 9. Consulta al dominio transmilenio.gov.co – No exitoso, se siguen resolviendo a las IP reales

Esta consulta confirma que el ataque fue **No Exitoso**. A pesar de la inundación de paquetes enviada por el script dnd.py, esta nueva consulta nslookup demuestra que el activo sigue resolviendo las direcciones IP reales. Esto evidencia que el dispositivo cuenta con protecciones que impidieron el envenenamiento, como la validación de IDs de transacción o la aleatorización de puertos de origen.

7.2 Resultados de las pruebas de seguridad en aplicaciones web

7.2.1 Inyecciones SQL – No exitosa

Se ejecuta un ataque automatizado con SQLMAP para validar una detección tentativa reportada en la fase de escaneos y análisis sobre uno de los portales evaluados.

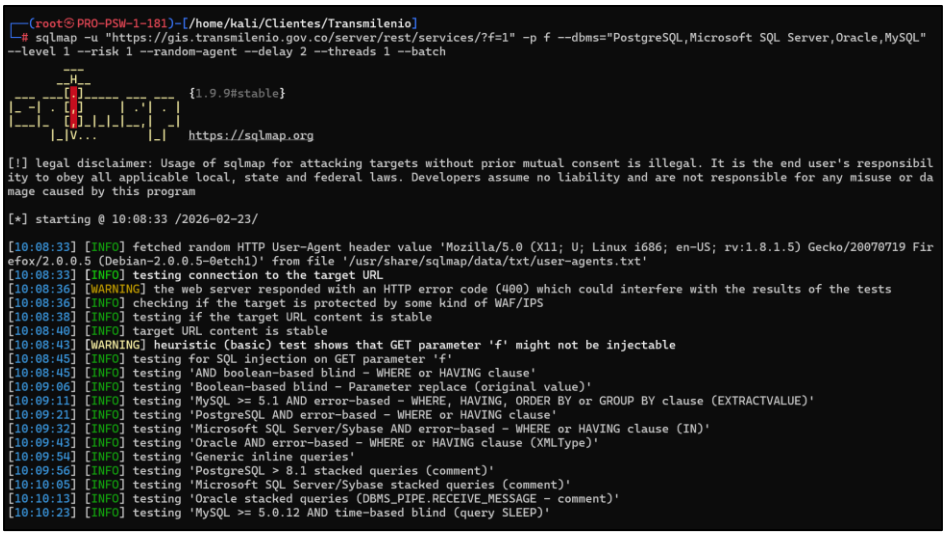


Figura 10. Inyecciones SQL con SQLMAP al portal gis.transmilenio.gov.co

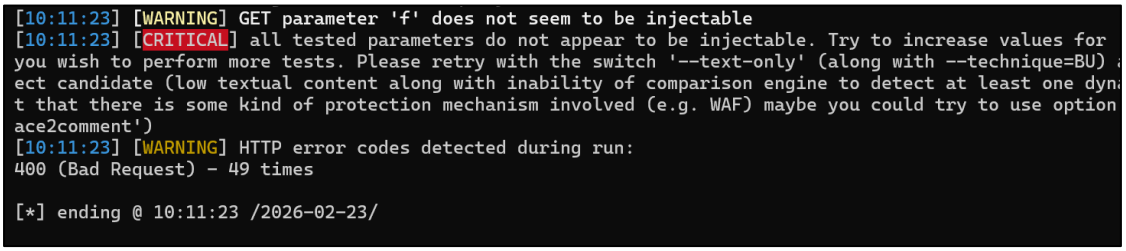


Figura 11. Resultado de la prueba de inyecciones SQL – No exitoso, los parámetros probados no son vulnerables.

La prueba no tuvo éxito, se probaron posibles parámetros vulnerables, pero ninguno dio una respuesta positiva en términos de ataque, esto indica que esta aplicación no es vulnerable a este tipo de ataques.

7.2.2 Cross-Site Scripting – No exitosa

En este caso no se encontraron debilidades graves, pero si alguna recomendación para controlar las entradas de usuario.



Figura 12. Inyección básica de script XSS – No exitoso, no se observó ningún tipo de ejecución

A pesar de no ejecutar ninguna acción, es evidente que no hay sanitización adecuada de entradas de usuario, para esta prueba es la única recomendación que se emite debido a que el campo de usuario permite ingresar y combinar diferentes tipos de caracteres que no deberían ser permitidos.

7.2.23 Hallazgos Proactivos / Fuera de Alcance (Out-of-Scope Insights)

Dado que no hubo muchos hallazgos a nivel web, se hizo una revisión de vulnerabilidades reportadas en el ciclo pasado (2025) para verificar que se hayan cerrado satisfactoriamente esas brechas, el único hallazgo a mencionar en este caso es el de un “drive temporal” que al parecer continúa almacenando información interna de la organización y que al momento de las pruebas fue posible obtener evidencia solida de esta exposición.



Figura 13. Exposición de carpetas en drivetemporal-tdoc.transmilenio.gov.co

Se encontró que esta debilidad aun persiste, aunque este activo no hace parte del alcance, se hizo la revisión de manera proactiva para cubrir posibles riesgos, en algunos archivos se observó exposición de credenciales que, a pesar de que no se tiene conocimiento de si fue alguna prueba en su momento.

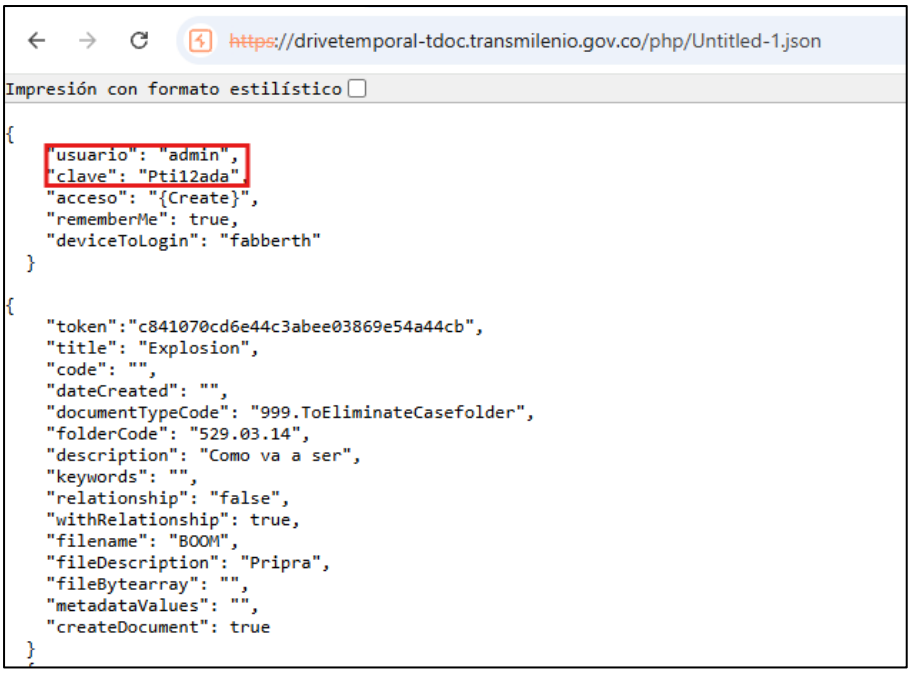


Figura 14. Credenciales expuestas en archivo accesible desde internet

Estas credenciales se probaron en algunos portales, pero no son válidas, de todas maneras, se registra el hallazgo para que sea revisado por el equipo de TRANSMILENIO y se aplique el ajuste que sea necesario.



Figura 15. Video de cámara de seguridad de vehículo público

Se encontraron archivos con fecha reciente, esto evidencia que el repositorio sigue siendo usado por el personal de TRANSMILENIO poniendo en riesgo la confidencialidad de esta información.



Figura 16. Copias de One Drive al corte de enero 30 de 2026

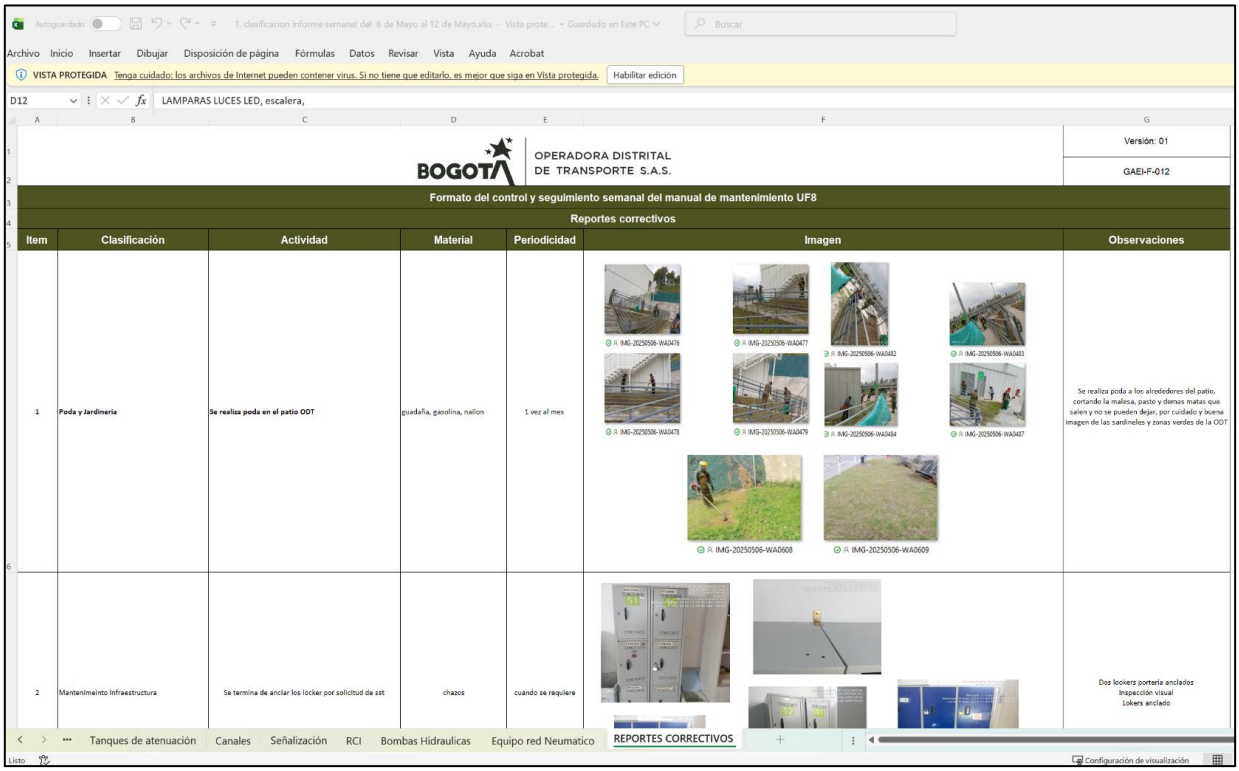


Figura 17. Archivo Excel con registros de mantenimiento locativo de junio 2025

En lo que respecta a hallazgos web no se tienen mas detecciones, este descubrimiento es de riesgo considerable a nivel de seguridad por eso se recomienda al equipo que tome las medidas necesarias para controlar el acceso a este “drive temporal”.

7.3 Resultados de las pruebas de seguridad en TransmiApp

Las pruebas a nivel de aplicación móvil mostraron una superficie de ataque moderada pero algunas configuraciones con importantes oportunidades de mejora.

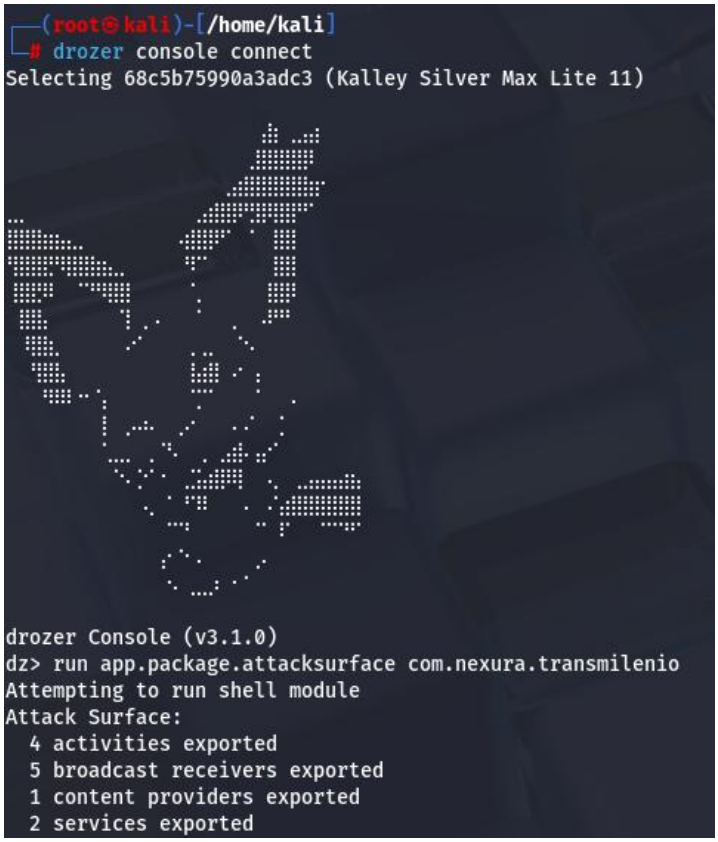


Figura 18. Reconocimiento de superficie de ataque con Drozer – 12 componentes exportados

El uso de herramientas de explotación (Drozer) permitió identificar y enumerar múltiples URIs de proveedores de contenido expuestas. Esta debilidad facilita que un atacante o una aplicación maliciosa en el mismo dispositivo intente realizar consultas directas a las bases de datos internas o archivos de la aplicación, saltándose los controles de acceso de la interfaz de usuario y comprometiendo la integridad de la información almacenada.

```
dz> run app.provider.finduri com.nexura.transmilenio
Attempting to run shell module
Scanning com.nexura.transmilenio...
content://com.nexura.transmilenio.huawei.push.provider/
content://com.nexura.transmilenio.AGCInitializeProvider/
content://com.nexura.transmilenio.androidx-startup/
content://com.nexura.transmilenio.AGCInitializeProvider
content://com.nexura.transmilenio.mobileadsinitprovider
content://com.nexura.transmilenio.firebaseinitprovider
content://com.nexura.transmilenio.firebaseinitprovider/
content://com.nexura.transmilenio.aaidinitprovider/
content://com.nexura.transmilenio.mobileadsinitprovider/
content://com.nexura.transmilenio.androidx-startup
content://com.nexura.transmilenio.huawei.push.provider
content://com.nexura.transmilenio
content://com.nexura.transmilenio.aaidinitprovider
content://com.nexura.transmilenio/
```

Figura 19. Exploración de URIs de proveedores

```
dz> run app.provider.query content://com.nexura.transmilenio/ --vertical
Attempting to run shell module
Exception occured: Permission Denial: opening provider androidx.core.content.FileProvider from ProcessRecord{51c85d5 16533:com.withsecure.dz:remote/u0a192} (pid=16533, uid=10192) that is not exported from UID 101
97
```

Figura 20. Consulta a content://com.nexura.transmilenio/ - No exitoso, acceso denegado

Aunque el finduri detectó la dirección, el sistema operativo bloqueó el acceso por dos razones:

1. **Falta de exported="true":** El proveedor específico (androidx.core.content.FileProvider) está marcado como privado en el Manifest.
2. **Diferencia de UID:** Drozer (UID 10192) intentó leer datos de TransMi (UID 10197) y Android detectó que no tienen permisos compartidos.

```
dz> run app.activity.start --component com.nexura.transmilenio com.nexura.transmilenio.Activity.MainActivity
Attempting to run shell module
```

Figura 21. Intento de ejecutar la actividad principal de TransmiApp

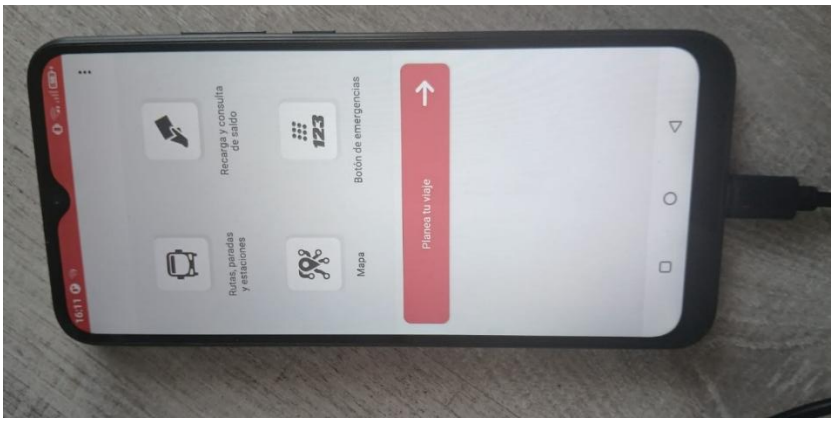


Figura 22. Actividad principal lanzada en el dispositivo evadiendo el "splash" de carga inicial

Se logró ingresar a la interfaz principal saltándose la imagen de carga inicial de TransmiApp, esta vulnerabilidad demuestra que la superficie de ataque está expuesta debido a que la actividad principal no está protegida por permisos ni configurada correctamente en el Manifiesto. Un atacante o una aplicación maliciosa en el mismo dispositivo podría forzar el inicio de la app en estados inesperados, facilitando ataques de denegación de servicio local o la exploración de funcionalidades internas sin pasar por los controles de seguridad perimetrales del software.

```
dz> run scanner.provider.injection -a com.nexura.transmilenio
Attempting to run shell module
Scanning com.nexura.transmilenio...
Not Vulnerable:
  content://com.nexura.transmilenio.AGCInitializeProvider/
  content://com.nexura.transmilenio.huawei.push.provider/
  content://com.nexura.transmilenio
  content://com.nexura.transmilenio.huawei.push.provider
  content://com.nexura.transmilenio.mobileadsinitprovider
  content://com.nexura.transmilenio.mobileadsinitprovider/
  content://com.nexura.transmilenio.aaidinitprovider/
  content://com.nexura.transmilenio.aaidinitprovider
  content://com.nexura.transmilenio.androidx-startup/
  content://com.nexura.transmilenio.AGCInitializeProvider
  content://com.nexura.transmilenio.firebaseinitprovider
  content://com.nexura.transmilenio/
  content://com.nexura.transmilenio.androidx-startup
  content://com.nexura.transmilenio.firebaseinitprovider/

Injection in Projection:
  No vulnerabilities found.

Injection in Selection:
  No vulnerabilities found.
```

Figura 23. Escaneo de inyecciones SQL a proveedores de contenido – No exitoso, no se reportan vulnerabilidades

El análisis del comando scanner.provider.injection indica que los **Content Providers** de la aplicación no son vulnerables a ataques de **SQL Injection** comunes a través de proyecciones o selecciones. Aunque las URIs están expuestas, los desarrolladores parecen haber implementado correctamente consultas parametrizadas o validaciones que impiden la manipulación de la base de datos local mediante este vector.

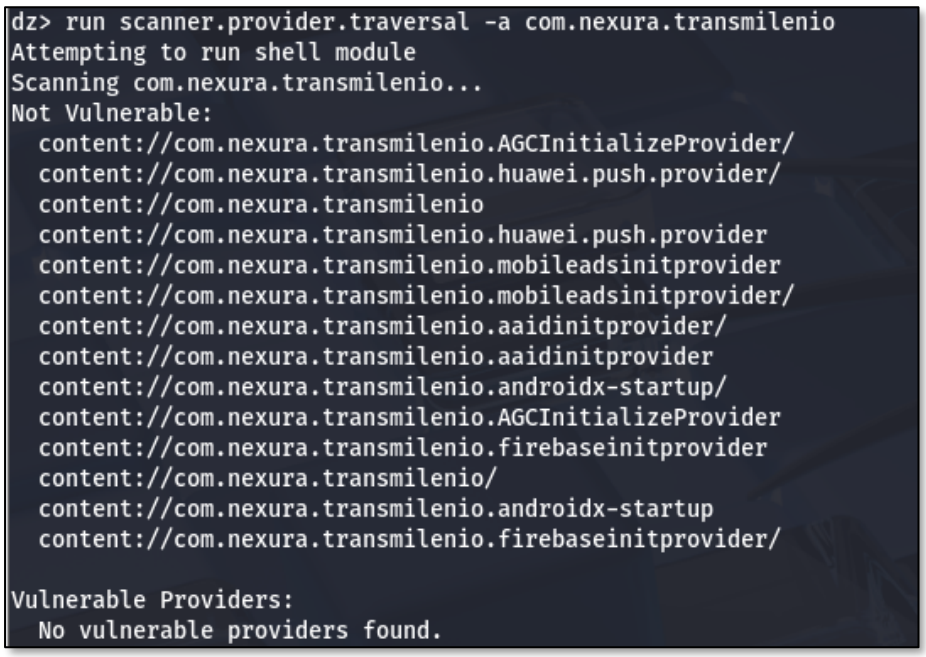


Figura 24. Escaneo de acceso no autorizado a rutas del sistema de proveedores de contenido – No exitoso, no se detectaron vulnerabilidades

El resultado del comando scanner.provider.traversal es **negativo**; Drozer no encontró proveedores que permitan el acceso no autorizado a archivos del sistema a través de rutas relativas (como ../../etc/password). Esto indica que los **Content Providers** de la app de TransMi están bien configurados contra este vector específico de ataque.

8. CONCLUSIONES

Todo este ejercicio nos ha permitido evidenciar una superficie de ataque baja-moderada, es decir, no hay un riesgo crítico que ponga a TRANSMILENIO en una desventaja de seguridad de llegar a perder el control de su infraestructura, pero si existe un riesgo principalmente de información interna expuesta públicamente como el caso de los videos de cámaras de seguridad de los buses que esta accesible en el drive a parte de los documentos que se observaron en algunas de las copias de One Drive descargadas de ese mismo repositorio (drive temporal), a nivel de infraestructura local, desde Password reconocemos el esfuerzo y efectividad que ha demostrado el equipo de TRANSMILENIO por el aseguramiento de sus activos y configuraciones de red que ha mejorado a lo largo del tiempo en los diferentes ciclos en los que hemos estado haciendo estas evaluaciones de seguridad, sin embargo, el camino de la seguridad siempre tiene nuevos desafíos y en este caso, nos presenta estas observaciones aquí documentadas, se anima al equipo para generar un plan de acción y cerrar estos hallazgos.

9. RECOMENDACIONES

Este numeral abarca la remediación integral de los tres frentes evaluados.

1. Fortalecimiento de la Infraestructura de Red y Servidores

- **Actualización de Sistemas Críticos:** Reemplazar o actualizar de forma inmediata los servidores con **Windows OS sin soporte**, ya que carecen de parches para vulnerabilidades de ejecución remota de código (RCE).
- **Hardening de Servicios de Red:** Restringir las consultas recursivas en el servidor **DNS** a solo IPs internas autorizadas para mitigar el envenenamiento de caché.
    - Implementar **DHCP Snooping** en los switches para evitar la detección y el abuso del servidor DHCP por parte de atacantes internos.

2. Seguridad en el Ciclo de Vida de la Aplicación Móvil

- **Aislamiento de Componentes (Hardening del Manifest):**
    - Configurar android:exported="false" en todas las Activities (incluyendo la MainActivity), Services y Providers que no requieran interacción externa.

- Establecer android:allowBackup="false" para prevenir la extracción de datos locales mediante ADB.

- **Seguridad en el Transporte y Compatibilidad:**
  - Elevar el minSdkVersion a **26 (Android 8.0)** para aprovechar las mejoras nativas de cifrado.
  - Desactivar usesCleartextTraffic y configurar un archivo de **Network Security Configuration** que fuerce HTTPS y aplique **Certificate Pinning**.

3. Mitigación de Vulnerabilidades en Activos Web

- **Implementación de Encabezados de Seguridad:**
  - Habilitar **HSTS (Strict-Transport-Security)** en todos los dominios para forzar conexiones cifradas permanentemente.
  - Configurar encabezados X-Frame-Options: DENY y una **Content Security Policy (CSP)** robusta para mitigar el Clickjacking y las inyecciones de JavaScript (DOM-based).
- **Control de Acceso y Sesiones:**
  - Restringir las políticas de **CORS** a una lista blanca (*allowlist*) de dominios corporativos específicos, eliminando el uso de comodines (*).
  - Asegurar que todas las cookies de sesión incluyan los flags Secure, HttpOnly y SameSite=Strict.

4. Estrategia de Monitoreo y Pruebas Continuas

- Integrar herramientas de **análisis estático (SAST)** como MobSF y de **análisis dinámico (DAST)** en el pipeline de CI/CD para detectar estas configuraciones permisivas antes de que lleguen a producción.
- Realizar ejercicios de **Adversary Emulation** semestrales para validar que los controles implementados resistan ataques reales de bypass como los realizados con Drozer.

5. Exposición de Datos en Portal Temporal (drivetemporal-tdoc)

- **Acción:** Eliminar la visibilidad pública del índice de archivos y restringir el acceso solo a personal autenticado.

- **Solución Técnica:** Desactivar el listado de directorios (*Directory Browsing*) en la configuración del servidor web (IIS o Apache) para evitar el estilo "Index of". Implementar obligatoriamente un control de acceso (Login) con MFA y ejecutar una política de purga automática que elimine archivos con más de 24 horas de antigüedad, garantizando que el almacenamiento sea realmente temporal.