ANALIZANDO TU DISPOSITIVO MÓVIL: ¿TE SIENTES SEGURO?

- **Arnau Vives Guasch**
- Technical lead en evaluaciones de seguridad de pago por móvil (HCE)
- **Applus+ Laboratories**
- arnau.vives@applus.com

Índice

1. Introducción y Contexto
2. Entorno móvil: puntos de riesgo
3. Vulnerabilidades y técnicas de ataque de aplicaciones
4. Estrategias de mitigación
5. Certificaciones existentes

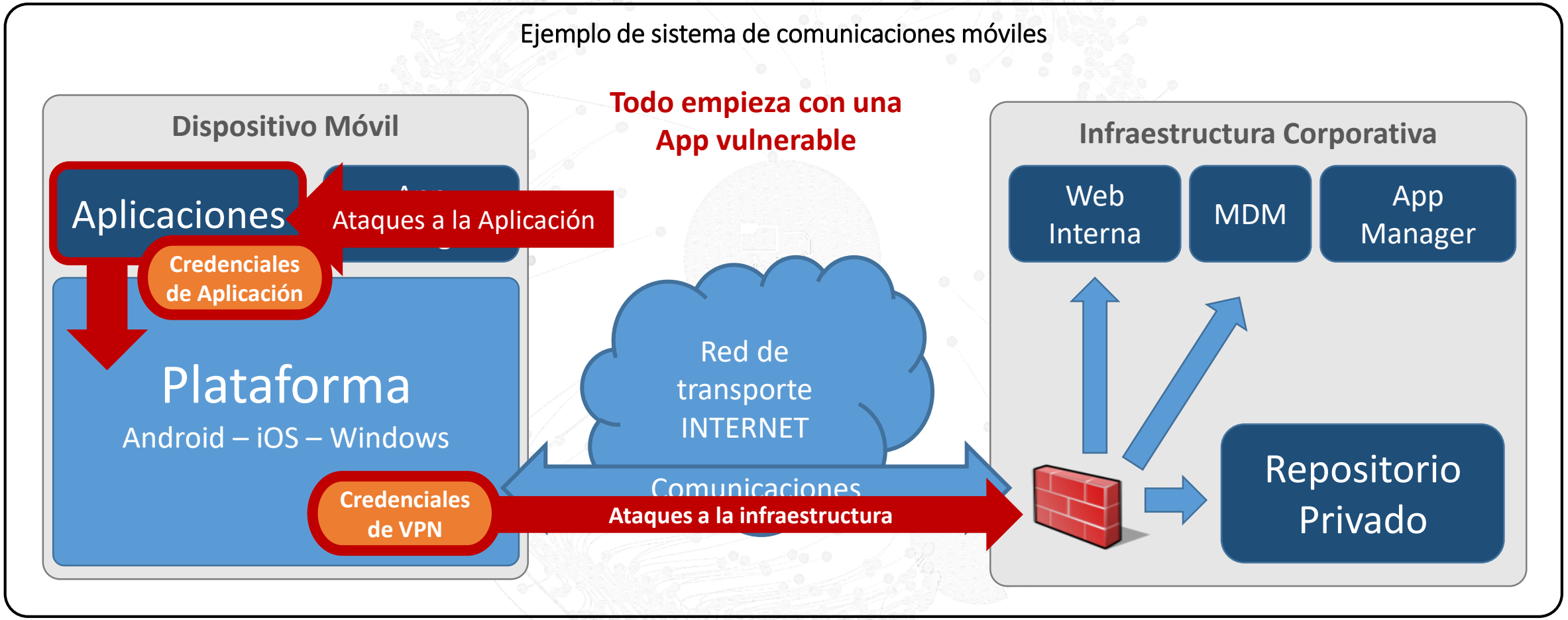
1. Introducción y Contexto

Incremento del **uso** de los **smartphones** para las **tareas diarias** en **organizaciones y empresas**.

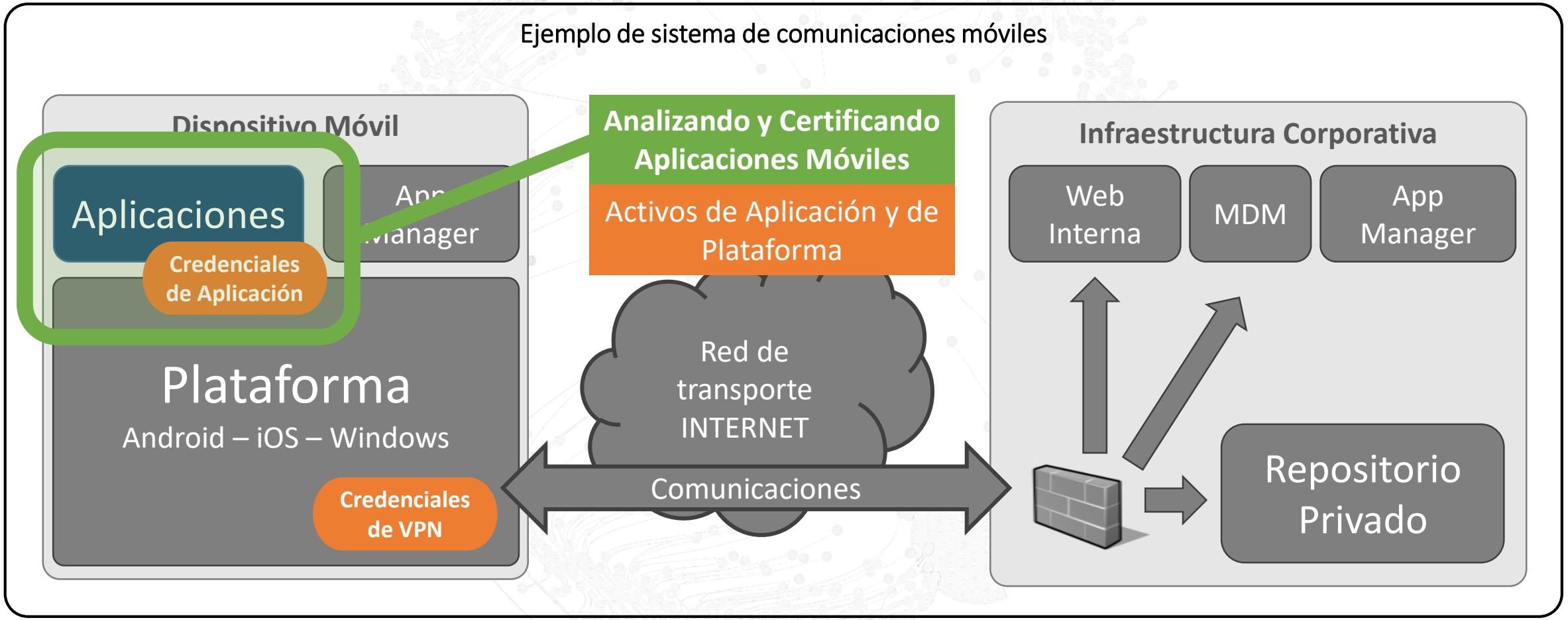
Datos más **sensibles** con el riesgo de estar más **expuestos**, en dispositivos que tienen menos capacidad de protección y con **riesgo de perdida o robo**.

Posibilidad de ataques **escalables** tipo malware.

2. Entorno móvil: puntos de riesgo



2. Entorno móvil: puntos de riesgo



Activos sensibles a proteger

3. Vulnerabilidades y técnicas de ataque: Precondiciones

- **Condiciones iniciales:** Atacante puede tener acceso **físico** o **remoto** al dispositivo (p.ej. malware).

- **Todas las aplicaciones pueden ser vulnerables**, depende del tipo de aplicacion los activos a proteger pueden ser distintos. Algunos ejemplos:
  - Aplicaciones de **pago,** de **identidad, mensajería, transporte, etc…**

Se debe tener en cuenta que cualquier aplicación puede también afectar a activos de la plataforma.

3. Vulnerabilidades y técnicas de ataque de aplicaciones

Técnicas nivel aplicación

ANÁLISIS ESTÁTICO

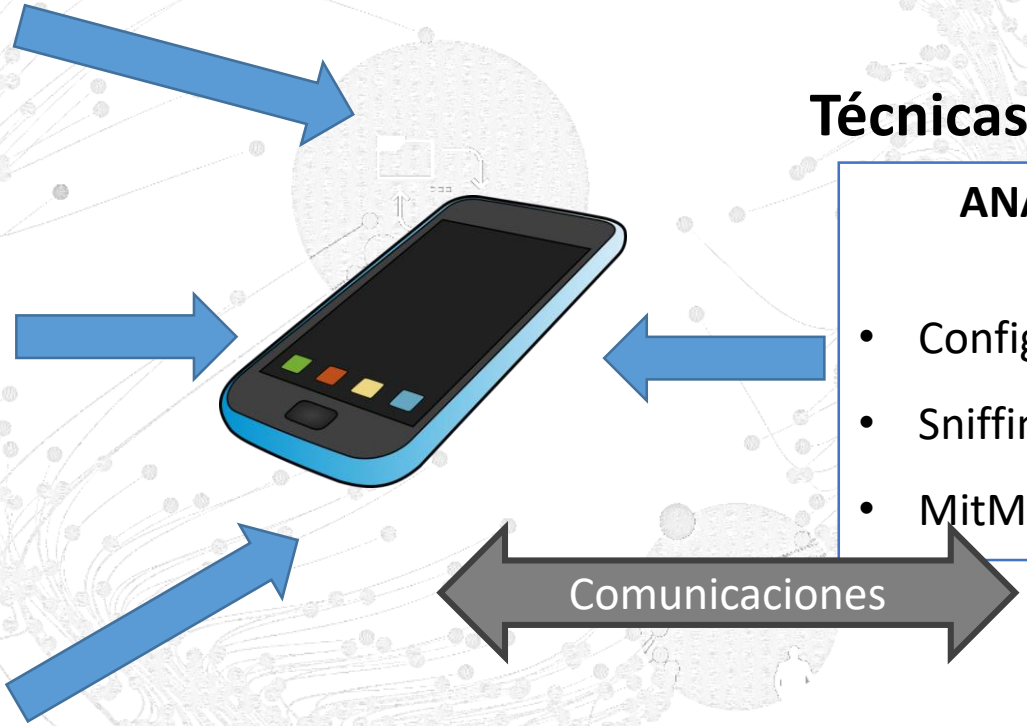
- Ingeniería inversa
- Binary Patching

ANÁLISIS DINÁMICO

- Runtime tampering
- File system /Memory analysis

CRIPTOANÁLISIS

- Architecture & Configuration
- WBC attack



Técnicas nivel comunicaciones

ANÁLISIS DE RED DESDE APLICACION

- Configuración/Implementación
- Sniffing
- MitM attack

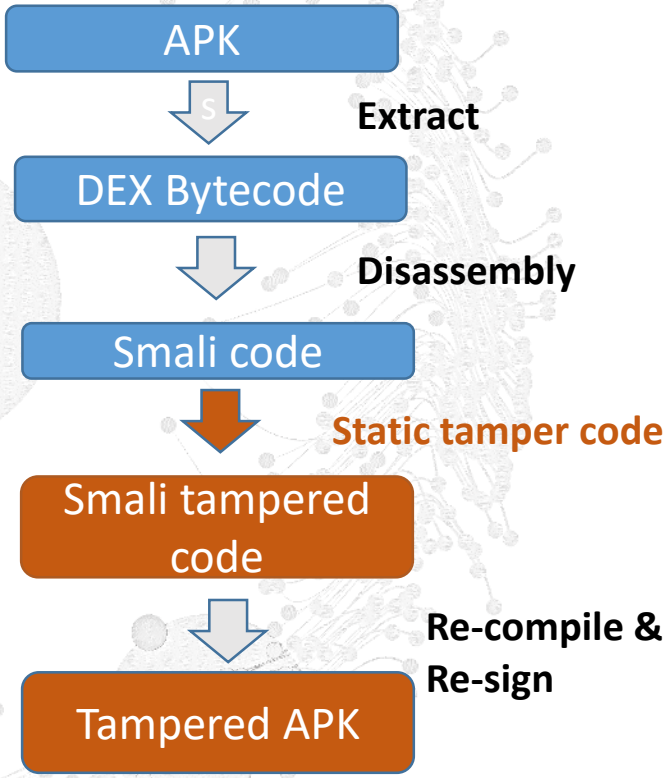
3. Vulnerabilidades y técnicas de ataque: Análisis Estático

INGENIERIA INVERSA

Decompilar, Desensamblar y Deofuscar.

BINARY PATCHING:

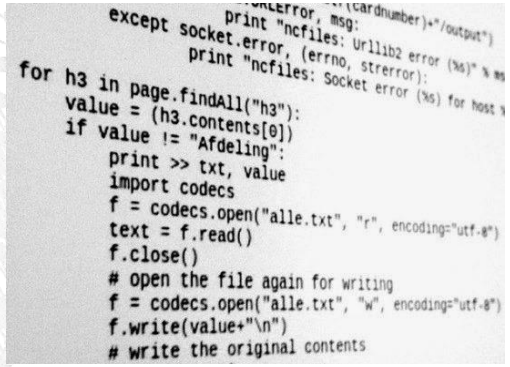
Generar una aplicación fraudulenta desde una aplicación original



3. Vulnerabilidades y técnicas de ataque: Análisis Dinámico

RUNTIME TAMPERING

- **Hooking**
  - Mediante frameworks, inyectar código malicioso en runtime para intervenir p.ej. funciones internas.
  - Saltarse funciones, cambiar su contenido, estimular funciones, etc..
- **Debugging**
  - Controlar el flujo de la aplicación en runtime, pudiendo parar en determinadas instrucciones.

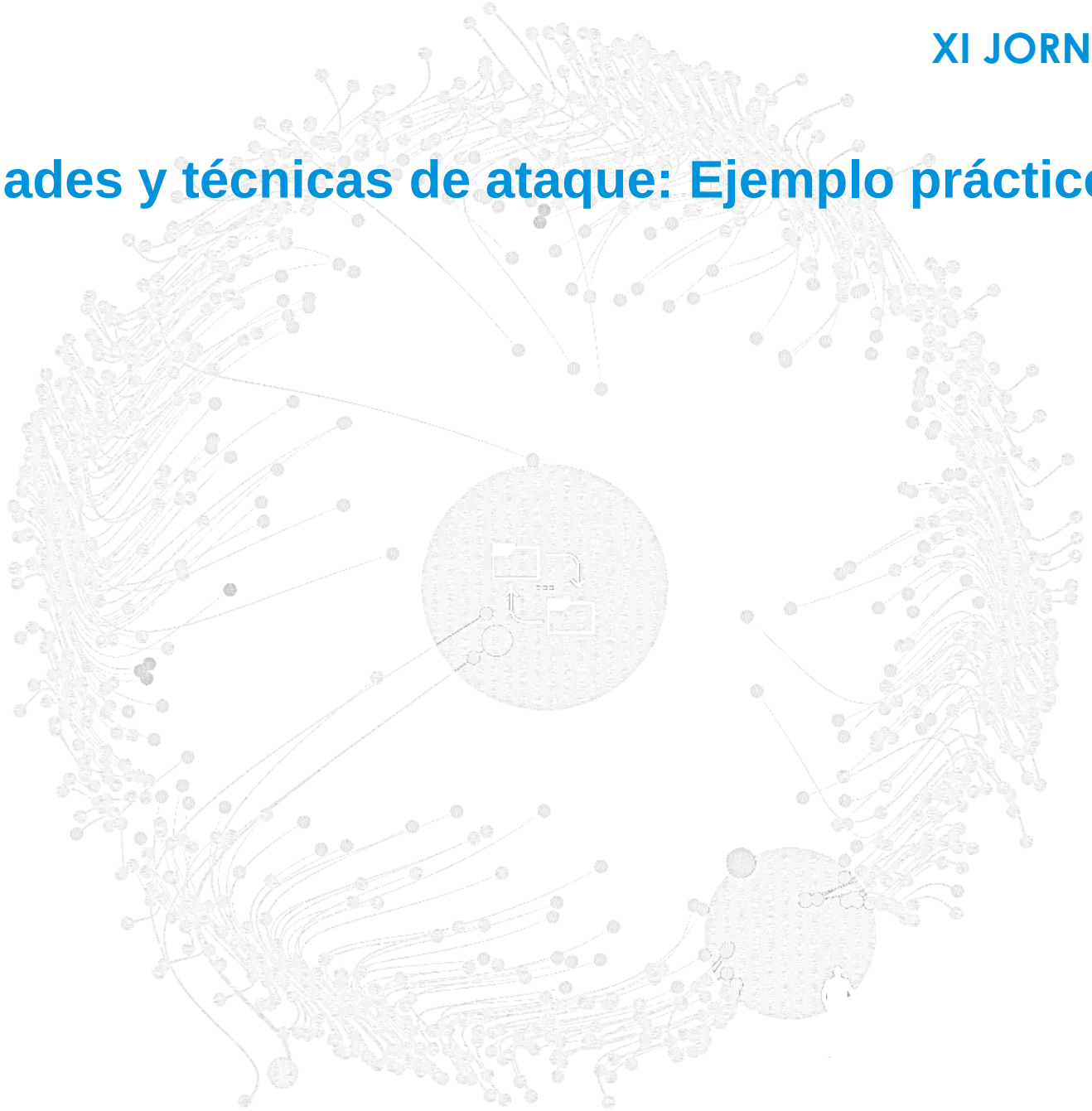


```
except socket.error, (errno, strerror):
        print "ncfiles: Socket error (%s) for host
for h3 in page.findAll("h3"):
    value = (h3.contents[0])
    if value != "Afdeling":
        print >> txt, value
        import codecs
        f = codecs.open("alle.txt", "r", encoding="utf-8")
        text = f.read()
        f.close()
        # open the file again for writing
        f = codecs.open("alle.txt", "w", encoding="utf-8")
        f.write(value+"\n")
        # write the original contents
```

INTERFERENCIAS CON OPERACIONES ENTRADA Y SALIDA (I/O).

- **Sistema de archivos:**
  - Modificar datos que se escriben o se leen para modificar el comportamiento de la aplicación.
- **Análisis de memoria:**
  - Congelar la ejecución y analizar el estado de memoria en cualquier punto del ciclo de vida de la aplicación.

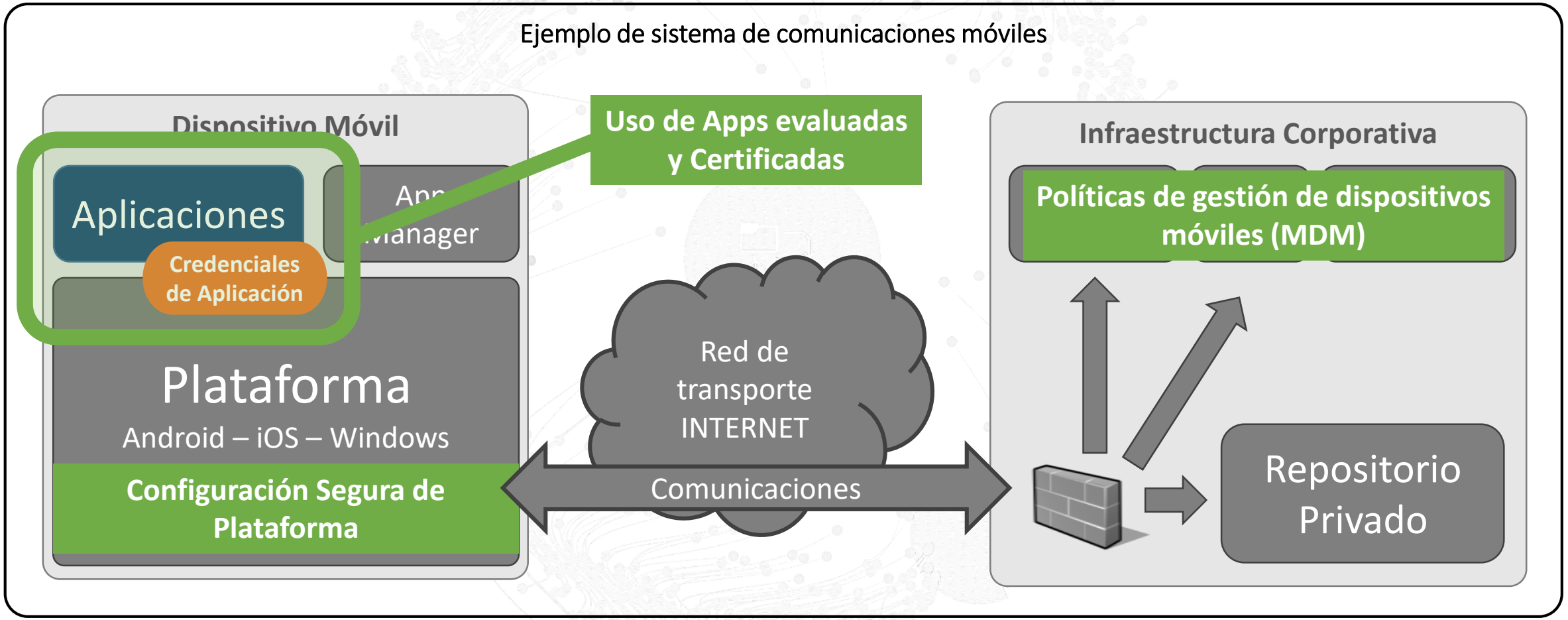
3. Vulnerabilidades y técnicas de ataque: Ejemplo práctico (Hooking)



4. Estrategias de mitigación

- **Implementar Contramedidas** en varios niveles:
  - A **nivel global**: La arquitectura global del sistema, la jerarquía de claves, evitar la escalabilidad de los ataques
  - A **nivel aplicativo**: Las soluciones móviles deben implementar técnicas de programación defensiva.
- **Estar en el estado del arte** de las técnicas de ataque y de defensa.
- **Configuración segura** de los dispositivos siguiendo guías estándares.

4. Certificaciones existentes



E-Mails

- info@ccn-cert.cni.es
- ccn@cni.es
- sat-inet@ccn-cert.cni.es
- sat-sara@ccn-cert.cni.es
- organismo.certificacion@cni.es

Websites

- www.ccn.cni.es
- www.ccn-cert.cni.es
- www.oc.ccn.cni.es

Síguenos en