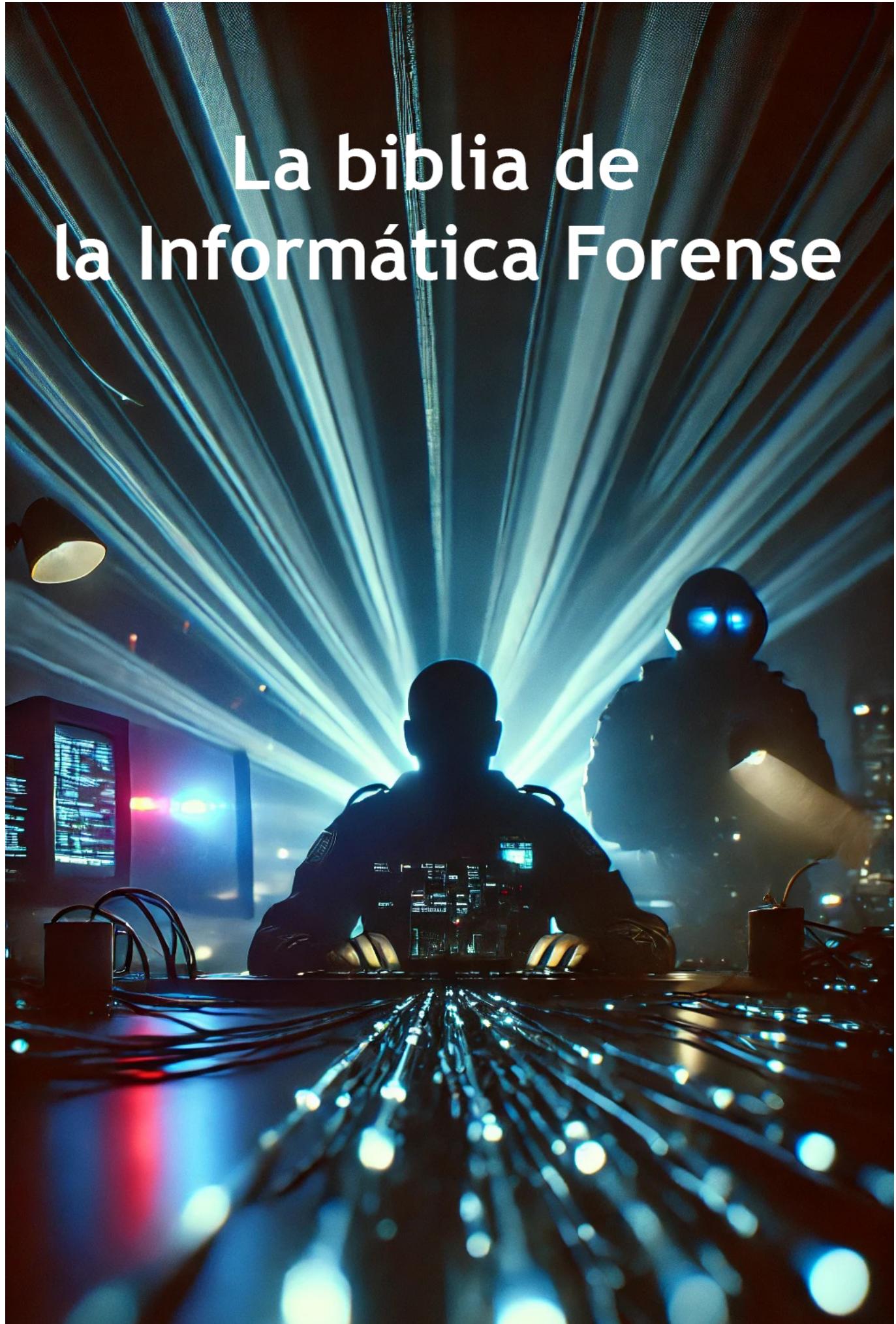


La biblia de la Informática Forense



Alejandro G Vera

Índice actualizado (con prólogo)

- **Prólogo** — Breve presentación del autor, alcance del libro y a quién va dirigido.
- **Capítulo 1 – Introducción a la Informática Forense**
- **Capítulo 2 – Principios y Normativas Internacionales en Informática Forense**
- **Capítulo 3 – Marco Legal en Informática Forense**
- **Capítulo 4 – Cadena de Custodia**
- **Capítulo 5 – Adquisición de Evidencia Digital**
- **Capítulo 6 – Análisis y Preservación de Datos**
- **Capítulo 7 – Recuperación de Archivos Eliminados**
- **Capítulo 8 – Análisis de Sistemas Operativos**
- **Capítulo 9 – Forense en Dispositivos Móviles**
- **Capítulo 10 – Análisis de Redes y Tráfico**
- **Capítulo 11 – Esteganografía y Ocultación de Información**
- **Capítulo 12 – Criptografía y Forense**
- **Capítulo 13 – Herramientas Profesionales de Informática Forense**
- **Capítulo 14 – Procedimientos Paso a Paso**
- **Capítulo 15 – Estudio de Casos Reales**
- **Capítulo 16 – Forense en la Nube**
- **Capítulo 17 – Windows Forensics Avanzado**
- **Capítulo 18 – Respuesta a Incidentes (DFIR)**
- **Capítulo 19 – Arquitectura de Red y DNS Forense**
- **Capítulo 20 – Análisis de Tráfico y Detección de Intrusiones**
- **Capítulo 21 – PCAP Avanzado y Zeek**
- **Capítulo 22 – Técnicas de Esteganografía y Contraesteganografía**
- **Capítulo 23 – Criptografía aplicada a comunicaciones seguras**
- **Capítulo 24 – Forense de Malware y Sandboxing**
- **Capítulo 25 – OSINT y atribución técnica**
- **Capítulo 26 – Forense de Correo Electrónico y Phishing**
- **Capítulo 27 – Logs y Telemetría en Endpoints (EDR)**
- **Capítulo 28 – Análisis Forense de Redes – Captura y PCAP**
- **Capítulo 29 – Esteganografía y su Detección Forense**
- **Capítulo 30 – Gestión de Logs y SIEM**
- **Capítulo 31 – Análisis de Memoria RAM**
- **Capítulo 32 – Forense en Linux y Unix (FS, artefactos y contenedores)**
- **Capítulo 33 – Forense de IoT y dispositivos embebidos**

- **Capítulo 34 – Blockchain y Criptomonedas en Investigación Forense**
 - **Capítulo 35 – Legislación Vigente en Argentina**
 - **Capítulo 36 – Legislación Internacional y Convenios**
 - **Capítulo 37 – Gestión de Evidencia Digital y Cadena de Custodia**
 - **Capítulo 38 – Redacción de Informes Periciales**
 - **Capítulo 39 – Presentación de Evidencias en Juicio**
 - **Capítulo 40 – Buenas Prácticas, Retos y Futuro de la Informática Forense**
-

Prólogo

La informática forense es mucho más que una disciplina técnica: es el puente entre el mundo digital y el sistema de justicia. Cada archivo recuperado, cada rastro en la red y cada línea de código analizada puede convertirse en la pieza clave que determine el resultado de una investigación. Este libro nace con la intención de reunir, en un solo volumen, los fundamentos, procedimientos y técnicas más relevantes para abordar este trabajo con rigor, ética y eficacia.

Soy **Alejandro G. Vera**, profesional con experiencia en ciberseguridad, desarrollo de software y formación tecnológica. A lo largo de mi trayectoria he visto cómo la criminalidad digital evoluciona a gran velocidad y cómo la capacidad de adaptarse y aprender continuamente se convierte en el activo más importante de cualquier perito. Este libro es el resultado de años de estudio, práctica y observación directa de casos y herramientas, con un enfoque eminentemente práctico.

"*La Biblia de la Informática Forense*" está pensada tanto para **profesionales del área** —peritos, analistas de ciberseguridad, investigadores policiales y judiciales— como para **estudiantes y entusiastas** que quieran adentrarse en el mundo del análisis forense digital. Su contenido abarca desde los conceptos básicos y la legislación vigente, hasta técnicas avanzadas de análisis de memoria, redes, esteganografía, criptografía, blockchain y entornos en la nube.

El objetivo es claro: que el lector no solo comprenda la teoría, sino que pueda aplicarla en casos reales. Por ello, cada capítulo está escrito con un equilibrio entre **precisión técnica** y **claridad expositiva**, incorporando ejemplos, herramientas concretas y procedimientos paso a paso. Aquí no encontrarás simples definiciones, sino un compendio de conocimientos listos para ser usados en laboratorio o en campo.

Este libro es, en esencia, una guía y un aliado. Una obra que acompaña al lector desde sus primeros pasos hasta la especialización, y que le prepara para enfrentar los retos actuales y futuros de la informática forense con solvencia, ética y profesionalismo.

Capítulo 1 – Introducción a la Informática Forense

1.1 Definición y Objetivos

La **informática forense** es la disciplina encargada de identificar, preservar, analizar y presentar datos informáticos con valor probatorio en un proceso legal o en una investigación técnica. A diferencia de otras ramas de la informática, su finalidad no es únicamente técnica, sino **jurídico-probatoria**, asegurando que la información obtenida pueda ser utilizada como evidencia legítima ante un tribunal, una auditoría interna o un proceso administrativo.

Objetivos principales:

- **Identificar** posibles fuentes de evidencia digital.
- **Preservar** la información evitando su alteración.
- **Analizar** los datos para reconstruir eventos.
- **Presentar** conclusiones de forma clara y verificable.

Ejemplo: Si una empresa sospecha que un empleado ha filtrado información, el análisis forense puede determinar desde qué equipo se envió la información, a qué hora y mediante qué medio, manteniendo la cadena de custodia para que esos datos puedan ser aceptados en un juicio.

1.2 Diferencia entre Informática Forense y Ciberseguridad

Aunque están relacionadas, no son lo mismo:

Ciberseguridad	Informática Forense
Previene ataques y protege sistemas.	Investiga incidentes ya ocurridos.
Se enfoca en defensa y monitoreo.	Se enfoca en recolección y análisis de evidencias.
Trabajo continuo y preventivo.	Trabajo puntual tras un incidente.
Herramientas: firewalls, antivirus, SIEM.	Herramientas: FTK, Autopsy, Volatility.

Un analista de ciberseguridad puede detener un ataque de ransomware; un perito forense investigará **cómo** se produjo, qué archivos fueron afectados y quién fue responsable.

1.3 Historia y Evolución de la Informática Forense

- **Década de 1980:** Los primeros casos judiciales con evidencia digital surgen en EE.UU., principalmente relacionados con fraude bancario y espionaje industrial.
- **Década de 1990:** Aparición de herramientas específicas como EnCase y FTK. Comienzan a definirse metodologías estandarizadas.
- **2000 en adelante:** La explosión de Internet, el cibercrimen organizado y el uso masivo de smartphones obliga a desarrollar técnicas forenses para redes, dispositivos móviles y entornos en la nube.
- **Actualidad:** La informática forense abarca desde análisis de malware hasta investigación en entornos IoT, inteligencia artificial y blockchains.

Ejemplo real: El caso **BTK Killer** en EE.UU. (2005) se resolvió gracias al análisis forense de un disquete enviado por el criminal a la policía, que reveló su nombre y ubicación.

1.4 Campos de Aplicación

La informática forense se aplica en múltiples contextos:

1. Ámbito Judicial:

- Investigación de delitos informáticos (phishing, fraude, ransomware).

- Casos de acoso digital, pornografía infantil, robo de identidad.

2. Ámbito Corporativo:

- Auditorías internas ante fuga de información.
- Investigación de abuso de recursos corporativos.

3. Ámbito Gubernamental:

- Inteligencia digital para seguridad nacional.
- Rastreo de ciberataques a infraestructuras críticas.

4. Ámbito Académico:

- Capacitación y entrenamiento de peritos y fuerzas de seguridad.
-

1.5 Perfil del Perito Informático

Un **perito informático forense** combina conocimientos técnicos, legales y habilidades de comunicación.

Debe ser capaz de:

- Trabajar bajo presión y con plazos judiciales estrictos.
- Documentar cada acción realizada.
- Explicar hallazgos técnicos en un lenguaje comprensible para jueces y abogados.

Habilidades clave:

- **Técnicas:** sistemas operativos, redes, hardware, software especializado.
- **Legales:** leyes de evidencia, cadena de custodia.
- **Analíticas:** pensamiento crítico, capacidad de reconstrucción de hechos.
- **Blandas:** ética, confidencialidad, comunicación clara.

Ejemplo: Un perito puede encontrar un archivo borrado, pero si no documenta correctamente **cómo** lo recuperó y **desde dónde**, ese archivo podría no ser aceptado como prueba.

1.6 Retos y Desafíos Actuales

- Creciente volumen y diversidad de datos (big data).
 - Uso de cifrado fuerte y anonimización.
 - Computación en la nube y jurisdicciones internacionales.
 - Privacidad y regulaciones como el GDPR en Europa.
 - IoT y nuevos dispositivos con datos relevantes.
-

Resumen del capítulo: La informática forense es el puente entre la tecnología y la justicia, donde cada paso debe ser técnico, legal y éticamente correcto. Este capítulo sentó las bases para comprender la disciplina, sus diferencias con la ciberseguridad, su historia, áreas de aplicación y el perfil del especialista que la ejerce.

Capítulo 2 – Principios y Normativas Internacionales en Informática Forense

2.1 La importancia de las normas en la informática forense

En informática forense, **no basta con encontrar la evidencia**: es imprescindible demostrar que fue obtenida de manera **válida, íntegra y conforme a la ley**. Las normativas y principios internacionales proporcionan un marco común que garantiza:

- **Integridad:** La evidencia no ha sido alterada.
- **Trazabilidad:** Se puede demostrar cómo y por quién fue manejada en cada etapa.
- **Reproducibilidad:** Otro perito puede repetir el proceso y llegar a las mismas conclusiones.
- **Admisibilidad:** El juez o autoridad acepta la evidencia como legítima.

La falta de cumplimiento con estas normas puede provocar que la evidencia se declare **inadmisible** en juicio, incluso si prueba la culpabilidad.

2.2 Principios fundamentales de la evidencia digital

Existen principios aceptados internacionalmente para la manipulación de datos digitales en investigaciones forenses. Entre los más importantes:

1. **Principio de Integridad:** La evidencia debe permanecer inalterada desde el momento de su adquisición. Esto se logra mediante técnicas como la creación de una imagen forense y el cálculo de hashes (MD5, SHA-256) antes y después del análisis.
 2. **Principio de Autenticidad:** Se debe demostrar que la evidencia es lo que se afirma que es. Ejemplo: un archivo de registro (log) debe provenir realmente del sistema investigado y no haber sido manipulado.
 3. **Principio de Reproducibilidad:** Otro profesional, con las mismas herramientas y procedimientos, debe poder obtener los mismos resultados.
 4. **Principio de Cadena de Custodia:** Debe registrarse todo el recorrido de la evidencia, desde su obtención hasta su presentación en juicio, incluyendo quién la tuvo y en qué condiciones.
 5. **Principio de Competencia Profesional:** El análisis debe ser realizado por personal capacitado y con experiencia comprobable.
-

2.3 Normas ISO relevantes para la informática forense

Las **normas ISO/IEC** proporcionan guías internacionales para el manejo de evidencia digital. Las más importantes son:

ISO/IEC 27037 – Directrices para la identificación, recolección y preservación de evidencia digital

- Explica cómo manejar dispositivos y datos para que sean admisibles en juicio.
- Establece procesos claros de adquisición.
- Define roles y responsabilidades en la recolección de evidencia.

ISO/IEC 27041 – Garantía de la idoneidad de los métodos de investigación

- Se centra en verificar que las herramientas y métodos usados sean apropiados para el caso.
- Recomienda validaciones previas a su aplicación.

ISO/IEC 27042 – Análisis de evidencia digital

- Describe cómo examinar, interpretar y documentar la evidencia.
- Establece criterios para los informes técnicos.

ISO/IEC 27043 – Principios de investigación de incidentes

- Guía para investigaciones más amplias, como ciberataques complejos.
- Integra aspectos técnicos y organizativos.

ISO/IEC 27050 – eDiscovery

- Orientada a la recuperación y análisis de datos electrónicos en contextos legales y corporativos.
- Fundamental en investigaciones de gran escala.

2.4 Buenas prácticas y guías internacionales

Además de las ISO, existen guías reconocidas:

- **NIST (National Institute of Standards and Technology)** – Publica documentos técnicos sobre procedimientos forenses, como el **NIST SP 800-86** para integrar técnicas forenses en respuesta a incidentes.
- **ENFSI (European Network of Forensic Science Institutes)** – Red europea que estandariza metodologías forenses, incluyendo la informática forense.
- **SWGDE (Scientific Working Group on Digital Evidence)** – Grupo de trabajo que desarrolla directrices sobre evidencia digital.

2.5 Ética profesional en la informática forense

El perito informático no solo debe cumplir con normas técnicas, sino también con principios éticos, ya que:

- Maneja información sensible y privada.
- Sus conclusiones pueden determinar la inocencia o culpabilidad de una persona.
- Cualquier sesgo, manipulación o negligencia puede invalidar una investigación.

Principios éticos clave:

- **Confidencialidad:** Proteger toda la información obtenida.
- **Imparcialidad:** No manipular resultados a favor de ninguna de las partes.

- **Transparencia:** Documentar todo el proceso para que pueda ser auditado.
 - **Actualización continua:** Estar al día con nuevas tecnologías y amenazas.
-

2.6 Ejemplo práctico: aplicación de normas en un caso real

Un caso judicial en Argentina involucró a un empleado acusado de robar bases de datos de clientes.

- La defensa intentó invalidar la evidencia alegando manipulación.
- El perito demostró que aplicó **ISO/IEC 27037**, calculó hashes antes y después del análisis, y documentó la cadena de custodia paso a paso.
- El juez aceptó la evidencia, y el acusado fue condenado.

Este ejemplo muestra que **seguir las normas no es burocracia, es blindar el trabajo para que resista cualquier objeción legal.**

Resumen del capítulo: Este capítulo estableció el marco normativo y los principios fundamentales que guían la informática forense a nivel internacional. El cumplimiento de estas reglas asegura que la evidencia digital sea íntegra, legítima y aceptada legalmente.

Capítulo 3 – Marco Legal en Informática Forense

3.1 Introducción

La informática forense opera en la intersección entre **la tecnología y el derecho**. Un perito no solo debe saber obtener y analizar evidencias, sino también entender **en qué marco jurídico se utilizarán**. Esto es clave, porque una evidencia técnicamente impecable pero obtenida fuera de la ley **puede ser descartada** en un juicio.

En este capítulo revisaremos:

- Legislación vigente en Argentina.
 - Comparativa con otras jurisdicciones.
 - Tratados internacionales que afectan el trabajo pericial.
 - Casos reales y jurisprudencia relevante.
-

3.2 Legislación Argentina aplicable

En Argentina, la informática forense se apoya en distintas leyes y reformas que contemplan el delito informático y la validez de la evidencia digital.

3.2.1 Código Penal Argentino

- **Ley 26.388 (2008):** Introdujo los delitos informáticos al Código Penal. Principales artículos:
 - **Art. 153 bis:** Acceso indebido a sistemas o datos.

- **Art. 183 y 184:** Daño a datos, programas o sistemas.
- **Art. 197:** Intercepción indebida de comunicaciones electrónicas.
- **Art. 292 y 296:** Falsificación de documentos digitales.

3.2.2 Ley 25.326 – Protección de Datos Personales

- Regula la recolección, tratamiento y almacenamiento de datos personales.
- Alineada con principios internacionales de privacidad.

3.2.3 Ley 25.506 – Firma Digital

- Reconoce la firma digital como equivalente legal de la firma manuscrita.
- Regula certificados y autoridades certificantes.

3.2.4 Ley 27.446 – Responsabilidad del Estado

- Incluye la protección de datos y sistemas críticos en el ámbito estatal.

3.2.5 Normas Procesales

- Código Procesal Penal de la Nación y códigos provinciales: regulan la incorporación de pruebas digitales y las facultades del perito.

3.3 Admisibilidad de la evidencia digital en Argentina

Para que una evidencia digital sea admitida en un juicio, debe cumplir:

1. **Relevancia:** Debe estar vinculada al caso.
2. **Autenticidad:** Demostración de que no ha sido alterada.
3. **Integridad:** Conservación sin cambios desde su obtención.
4. **Legalidad:** Obtención conforme a la ley y con orden judicial cuando sea necesario.

Ejemplo: Si un perito extrae datos de un teléfono sin autorización judicial y sin consentimiento del dueño, la defensa puede pedir la nulidad de la prueba.

3.4 Comparativa internacional

Estados Unidos

- Leyes como el **CFAA (Computer Fraud and Abuse Act)** castigan accesos no autorizados.
- Normas federales de evidencia (Federal Rules of Evidence) establecen estándares de admisibilidad.
- Procedimientos de **eDiscovery** en litigios corporativos.

Unión Europea

- Directiva 2013/40/UE sobre ataques contra sistemas de información.
- Reglamento General de Protección de Datos (**GDPR**) con fuertes sanciones por violar la privacidad.
- Estricto control sobre transferencia internacional de datos.

Latinoamérica

- Chile: Ley 19.223 (delitos informáticos).
 - México: Código Penal Federal y Ley Federal de Protección de Datos Personales.
 - Brasil: Marco Civil da Internet y Ley General de Protección de Datos (LGPD).
-

3.5 Tratados y acuerdos internacionales

Convenio de Budapest sobre Ciberdelincuencia

- Principal tratado internacional en delitos informáticos.
- Establece cooperación entre países para la investigación y extradición.
- Argentina aún no es signataria, pero varios principios son aplicados como referencia.

Acuerdos bilaterales

- Cooperación judicial y policial con países como España, Brasil y EE.UU.
-

3.6 Jurisprudencia relevante en Argentina

Caso: "Banco Nación vs. Ataque Cibernético" (2018)

- Un ataque a la banca online generó pérdidas millonarias.
- La investigación forense identificó a los responsables mediante análisis de logs y rastreo de transferencias.
- El tribunal destacó la validez de la cadena de custodia y el uso de hashes para garantizar la integridad de la evidencia.

Caso: "Ministerio Público Fiscal vs. X" (2021)

- Investigación por acoso digital y difusión no autorizada de imágenes íntimas.
 - La evidencia de chats de WhatsApp fue aceptada gracias a la pericia que demostró extracción certificada y preservación íntegra.
-

3.7 Desafíos legales actuales

- Falta de un marco unificado en todas las provincias argentinas.
 - Dificultades para investigar delitos transnacionales sin tratados bilaterales.
 - Colisión entre derecho a la privacidad y facultades de investigación.
 - Pruebas obtenidas en la nube: ¿jurisdicción local o extranjera?
-

Resumen del capítulo: El marco legal es tan importante como la técnica en informática forense. En Argentina, la legislación actual cubre muchos aspectos de la evidencia digital, pero la globalidad de Internet exige conocer también leyes extranjeras y tratados internacionales. Un perito debe trabajar siempre alineado con este marco para garantizar la validez de su labor.

Capítulo 4 – Cadena de Custodia

4.1 Introducción

En informática forense, la **cadena de custodia** es el procedimiento documentado que garantiza la **trazabilidad, integridad y legitimidad** de la evidencia digital desde el momento de su obtención hasta su presentación en juicio.

Si la cadena de custodia se rompe o presenta irregularidades, **la evidencia puede ser considerada inválida**, incluso si técnicamente demuestra un delito.

En pocas palabras:

“No es suficiente con encontrar la verdad, hay que poder probarla de forma legal”.

4.2 Objetivos de la cadena de custodia

1. **Garantizar la integridad:** Que la evidencia se mantenga exactamente igual desde que se obtiene hasta que se presenta.
 2. **Registrar la trazabilidad:** Documentar cada persona que tuvo acceso y en qué condiciones.
 3. **Asegurar la admisibilidad:** Cumplir con requisitos legales y procesales.
 4. **Prevenir manipulaciones:** Minimizar el riesgo de alteraciones, pérdidas o destrucción.
-

4.3 Etapas de la cadena de custodia

4.3.1 Identificación de la evidencia

- Localizar y reconocer elementos que puedan contener información relevante.
- Ejemplos: discos duros, SSD, pendrives, teléfonos, servidores, registros de red, imágenes forenses.

4.3.2 Preservación

- Proteger la evidencia para evitar daños o alteraciones.
- Uso de bolsas antiestáticas, sellos de seguridad, y almacenamiento en entornos controlados.

4.3.3 Recolección

- Adquirir la evidencia siguiendo procedimientos forenses.
- Ejemplo: crear una imagen bit a bit de un disco con herramientas como **FTK Imager** o **dd**.
- Calcular **hashes** (MD5, SHA-256) en el momento de la adquisición.

4.3.4 Transporte

- Trasladar la evidencia física o digital bajo condiciones seguras.
- Documentar fecha, hora, responsable y medio de transporte.

4.3.5 Almacenamiento

- Guardar la evidencia en lugares seguros, con control de acceso y registro de entradas/salidas.
 - En evidencias digitales: copias de seguridad y almacenamiento en medios de solo lectura.
-

4.4 Elementos esenciales de la documentación

Un registro de cadena de custodia debe contener como mínimo:

- Número único de identificación de la evidencia.
 - Descripción detallada (tipo de dispositivo, marca, modelo, número de serie).
 - Fecha y hora de cada movimiento.
 - Nombres y firmas de las personas responsables en cada etapa.
 - Valores hash antes y después de cada manipulación.
 - Observaciones (condición física, sellos, embalaje).
-

4.5 Ejemplo de formulario de cadena de custodia

ID Evidencia	Descripción	Fecha/Hora	Acción	Responsable	Firma	Hash MD5/SHA- 256
001-2025	HDD 1TB Seagate SN:AB12345	08/08/2025 – 14:32	Adquisición de imagen forense	Alejandro G. Vera	_____	MD5: 3b1d... / SHA-256: 7f2c...
001-2025	HDD 1TB Seagate SN:AB12345	08/08/2025 – 16:10	Transferencia a custodia judicial	Juan Pérez (Perito Oficial)	_____	MD5: 3b1d... / SHA-256: 7f2c...

4.6 Buenas prácticas en la cadena de custodia

- Usar **identificadores únicos** para cada evidencia.
 - Sellar físicamente los dispositivos cuando sea posible.
 - Limitar el número de personas que manipulan la evidencia.
 - Evitar trabajar directamente sobre la evidencia original; siempre usar copias forenses.
 - Mantener copias de la documentación en formato físico y digital.
-

4.7 Errores comunes que rompen la cadena de custodia

- No calcular ni registrar hashes.
- Dejar evidencia sin embalaje adecuado.
- No documentar cambios de manos.
- Alterar la evidencia durante el análisis por no trabajar sobre copias.

Ejemplo real: En un caso de fraude bancario, un disco duro fue manipulado sin generar un registro de cadena de custodia. El abogado defensor argumentó que pudo haber sido modificado, y la evidencia fue excluida del juicio.

4.8 Herramientas de soporte

- **Software de gestión de cadena de custodia:** CaseGuard, Tracker Products.
- **Generadores de hash:** HashCalc, md5sum, sha256sum.
- **Equipos de adquisición forense:** Tableau Forensic Duplicator, Logicube Falcon.

Resumen del capítulo: La cadena de custodia es el eje central de la validez de la evidencia digital. Sin una trazabilidad completa y documentada, cualquier hallazgo técnico pierde valor legal. Un buen perito protege tanto la evidencia como el procedimiento que la respalda.

Capítulo 5 – Adquisición de Evidencia Digital

5.1 Introducción

La **adquisición de evidencia digital** es el proceso de obtener una copia exacta y verificable de los datos contenidos en un dispositivo, sin alterar el original. En informática forense, la máxima es:

"Nunca se trabaja sobre la evidencia original, siempre sobre una copia forense".

Este procedimiento es crítico porque cualquier error en la adquisición **puede comprometer la integridad y la validez legal de la prueba**.

5.2 Tipos de adquisición

5.2.1 Adquisición física (bit a bit)

- Crea una copia exacta del dispositivo, incluyendo sectores vacíos y espacio no asignado.
- Ideal para recuperar archivos borrados o datos ocultos.
- Ejemplo: Clonación de un disco de 1TB en formato **.E01** (EnCase Evidence File).

5.2.2 Adquisición lógica

- Copia solo los archivos visibles y accesibles del sistema de archivos.
- Más rápida, pero no incluye sectores no asignados ni datos eliminados.
- Útil para dispositivos con cifrado activo donde no es posible copia física.

5.2.3 Adquisición en vivo (Live Forensics)

- Se realiza mientras el sistema está encendido.
- Permite capturar información volátil como procesos activos, conexiones de red y memoria RAM.
- Requiere extremo cuidado para no alterar el sistema.

5.3 Consideraciones previas

Antes de iniciar la adquisición:

1. **Evaluar el estado del dispositivo** (encendido, apagado, dañado).
2. **Determinar el tipo de adquisición** más adecuado.
3. **Verificar autorización legal** (orden judicial o consentimiento).
4. **Preparar las herramientas y medios de almacenamiento.**
5. **Registrar todo** en la documentación de cadena de custodia.

5.4 Procedimiento general de adquisición física

1. Identificación del dispositivo

- Marca, modelo, número de serie.
- Tipo de conexión (SATA, NVMe, USB).

2. Uso de bloqueadores de escritura (write blockers)

- Dispositivos que impiden modificar el contenido del medio.
- Ejemplos: Tableau T35u, WiebeTech Forensic UltraDock.

3. Selección de la herramienta forense

- **FTK Imager** (GUI y CLI).
- **dd o dc3dd** (Linux).
- **Guymager** (Linux, interfaz gráfica).

4. Creación de la imagen forense

- Comando ejemplo en Linux:

```
sudo dc3dd if=/dev/sda of=/mnt/caso/evidencia.dd hash=sha256
log=registro.log
```

5. Cálculo de hashes

- Comparar hash de origen y copia para garantizar integridad.

```
sha256sum evidencia.dd
```

6. Documentación

- Anotar hora de inicio y fin, herramientas usadas, responsable y hash.

5.5 Adquisición de diferentes medios

5.5.1 Discos HDD

- Generalmente no presentan dificultades, siempre que se use bloqueador de escritura.
- Recuperación de datos borrados posible en la mayoría de casos.

5.5.2 Discos SSD

- El comando TRIM puede haber borrado de forma irrecuperable ciertos datos.
- Se recomienda adquisición inmediata para maximizar recuperación.

5.5.3 Pendrives y tarjetas de memoria

- Alta probabilidad de datos fragmentados.
- Útiles herramientas como **Photorec** y **R-Studio**.

5.5.4 Memoria RAM

- Contiene información volátil: contraseñas, claves de cifrado, historial de comandos.
- Herramientas: **Belkasoft RAM Capturer**, **DumpIt**, **Volatility** para análisis posterior.

5.6 Ejemplo práctico: adquisición de un pendrive de 32GB

1. **Identificación:** Pendrive Kingston 32GB, ID: 005-2025.
2. **Conexión a bloqueador de escritura USB.**
3. **Herramienta:** Guymager en Linux.
4. **Imagen forense:** Guardada como **pendrive005.E01**.
5. **Hashes:**
 - MD5: **9f86d081884c7d659a2feaa0c55ad015**
 - SHA-256: **b94d27b9934d3e08a52e52d7da7dabfa**
6. **Registro:** Documentado en cadena de custodia con fecha, hora y responsable.

5.7 Herramientas recomendadas

- **FTK Imager** – Adquisición física y lógica, verificación de hashes.
- **Autopsy/Sleuth Kit** – Análisis posterior de imágenes.
- **dd / dc3dd** – Copia bit a bit en entornos Unix/Linux.
- **Guymager** – Interfaz gráfica para adquisición en Linux.
- **Belkasoft RAM Capturer** – Captura de memoria volátil en Windows.

5.8 Errores comunes en la adquisición

- No usar bloqueador de escritura.
- Olvidar calcular o registrar hashes.

- Guardar la imagen en el mismo dispositivo fuente.
 - No verificar la integridad de la copia antes de continuar.
-

Resumen del capítulo: La adquisición es uno de los pasos más críticos en informática forense. Un procedimiento mal ejecutado puede destruir la validez de toda la investigación. Seguir las mejores prácticas, documentar cada paso y proteger la integridad son la base de un trabajo forense sólido.

Capítulo 6 – Análisis y Preservación de Datos

6.1 Introducción

Una vez adquirida la evidencia digital, el siguiente paso es **analizarla** para extraer información relevante, y **preservarla** para mantener su integridad durante todo el proceso judicial o corporativo.

En informática forense, analizar no es simplemente “abrir y mirar”:

- Debe hacerse **sobre copias forenses**, nunca sobre el original.
 - Cada acción debe ser **documentada y reproducible**.
 - La evidencia debe mantenerse **intacta y protegida** para que sea admisible legalmente.
-

6.2 Principios fundamentales de la preservación

1. **Integridad** – La información no debe ser alterada.
 2. **Confidencialidad** – Solo personas autorizadas deben acceder a la evidencia.
 3. **Disponibilidad** – La evidencia debe estar accesible para análisis autorizado, pero siempre bajo control.
 4. **Trazabilidad** – Todo acceso o manipulación debe estar registrado.
-

6.3 Uso de copias forenses

El análisis debe realizarse siempre sobre una **imagen forense** (copia bit a bit) obtenida en el proceso de adquisición.

Ventajas:

- Permite repetir análisis si es necesario.
 - Protege el original de alteraciones accidentales.
 - Facilita la preservación a largo plazo.
-

6.4 Verificación de integridad con funciones hash

Antes y después de cualquier análisis, se deben calcular y registrar **valores hash** para garantizar que los datos no han cambiado.

Algoritmos más comunes:

- **MD5** – Rápido, pero no infalible ante colisiones.
- **SHA-1** – Más seguro que MD5, pero en desuso.
- **SHA-256** – Actualmente recomendado para pericias.

Ejemplo en Linux:

```
sha256sum imagen_forense.dd
```

6.5 Almacenamiento seguro de la evidencia

6.5.1 Evidencia física

- Guardar en bolsas antiestáticas y selladas.
- Etiquetado claro con ID único.
- Almacenar en cajas fuertes o gabinetes con control de acceso.

6.5.2 Evidencia digital

- Guardar copias forenses en medios de solo lectura (Blu-ray, WORM).
- Usar cifrado para proteger la confidencialidad.
- Mantener copias redundantes en ubicaciones distintas.

6.6 Procedimiento de análisis

1. Preparar el entorno de trabajo

- Usar estación forense aislada de internet.
- Cargar la copia forense en herramientas de análisis.

2. Identificar el sistema de archivos

- NTFS, FAT, EXT4, HFS+, APFS, etc.

3. Explorar artefactos relevantes

- Archivos de registro (logs).
- Historial de navegación.
- Archivos recientes.
- Correo electrónico.
- Archivos borrados (recuperación).

4. Extraer y preservar datos clave

- Guardar en carpetas separadas.
- Calcular hashes de cada archivo extraído.

5. Documentar hallazgos

- Descripción clara.
 - Capturas de pantalla.
 - Hash de cada evidencia.
-

6.7 Herramientas de análisis forense

- **Autopsy / Sleuth Kit** – Análisis general de discos.
 - **X-Ways Forensics** – Análisis profesional de imágenes forenses.
 - **Volatility** – Análisis de memoria RAM.
 - **Wireshark** – Análisis de tráfico de red.
 - **FTK** – Análisis de datos estructurados y correo electrónico.
-

6.8 Preservación a largo plazo

En investigaciones judiciales, la evidencia puede requerir almacenamiento durante **años**. Buenas prácticas:

- Revisar periódicamente el estado de los medios de almacenamiento.
 - Migrar datos a nuevos formatos o medios cuando sea necesario.
 - Mantener registros actualizados de la ubicación y custodia.
-

6.9 Ejemplo práctico: preservación de una imagen forense

Caso: Imagen **.E01** de un disco de 500GB obtenida en un caso de fraude corporativo.

1. Guardar la copia forense en dos discos duros externos cifrados con VeraCrypt.
 2. Depositar uno en el laboratorio forense y otro en una caja de seguridad bancaria.
 3. Calcular y registrar hash SHA-256 de la imagen.
 4. Documentar ubicación, responsables y fechas de revisión.
-

6.10 Errores comunes en la preservación

- Guardar la evidencia solo en un medio y sin copias.
 - No protegerla contra accesos no autorizados.
 - No recalcular hashes periódicamente para verificar integridad.
 - Almacenar evidencias digitales junto con datos operativos del laboratorio.
-

Resumen del capítulo: El análisis y la preservación de datos son pilares fundamentales de la informática forense. La integridad se mantiene trabajando siempre sobre copias forenses, usando funciones hash para verificar datos y almacenando la evidencia de forma segura y controlada.

Capítulo 7 – Recuperación de Archivos Eliminados

7.1 Introducción

En la informática forense, la **recuperación de archivos eliminados** es una de las tareas más frecuentes y valiosas. Cuando un usuario borra un archivo, normalmente **no se elimina físicamente de inmediato**, sino que el sistema de archivos marca ese espacio como disponible para ser sobrescrito.

El trabajo del perito consiste en:

- Localizar esos datos.
 - Recuperarlos en el estado más íntegro posible.
 - Documentar el proceso para que la recuperación sea admisible como prueba.
-

7.2 Conceptos clave

1. Borrado lógico vs. borrado físico

- *Lógico*: El archivo desaparece del índice del sistema de archivos, pero su contenido sigue en el disco hasta ser sobrescrito.
- *Físico*: El archivo se sobreescribe o se destruye con métodos como "wipe" seguro.

2. Espacio no asignado (unallocated space)

- Sectores del disco que el sistema considera libres, pero que pueden contener datos recuperables.

3. Slack space

- Espacio sobrante en un bloque de almacenamiento que no está siendo usado por el archivo actual, pero que puede contener fragmentos de archivos antiguos.
-

7.3 Factores que influyen en la recuperación

- **Tipo de sistema de archivos**: NTFS, FAT32, EXT4, APFS, etc.
 - **Tiempo transcurrido desde el borrado**: Cuanto más tiempo, más riesgo de sobrescritura.
 - **Uso del dispositivo**: Alta actividad = alta probabilidad de sobrescritura.
 - **Tipo de medio**: HDD, SSD, memoria flash, etc.
 - **Cifrado**: Si el medio está cifrado, la recuperación puede requerir claves.
-

7.4 Recuperación en HDD (Discos Duros Mecánicos)

Ventajas:

- Alta probabilidad de recuperación si los datos no han sido sobrescritos.
- La velocidad de recuperación no se ve afectada por borrados previos.

Herramientas recomendadas:

- **TestDisk** – Recupera particiones y archivos borrados.
- **R-Studio** – Herramienta comercial muy completa.
- **Autopsy/Sleuth Kit** – Integración de recuperación y análisis.

Ejemplo de comando con TestDisk:

```
testdisk /log
```

Permite analizar particiones y listar archivos borrados para recuperarlos.

7.5 Recuperación en SSD (Unidades de Estado Sólido)

Desafíos:

- El comando **TRIM** borra de forma inmediata los bloques no usados, reduciendo las posibilidades de recuperación.
- La recuperación depende de si TRIM está habilitado y del uso posterior al borrado.

Buenas prácticas:

- Inmovilizar el dispositivo inmediatamente después de detectar el borrado.
- Desactivar TRIM en el sistema si es posible.
- Usar adquisición forense directa para capturar lo que quede en memoria NAND.

Herramientas:

- **R-Studio** y **UFS Explorer** tienen módulos optimizados para SSD.

7.6 Recuperación en pendrives y tarjetas de memoria

Características:

- Utilizan memoria flash similar a los SSD, con posibles algoritmos de desgaste y borrado.
- Alta probabilidad de fragmentación de archivos grandes.

Herramientas:

- **PhotoRec** – Especializada en recuperación por tipo de archivo (firma).
- **Recuva** – Sencilla y gratuita, útil para casos rápidos.
- **Disk Drill** – Soporta múltiples formatos y sistemas de archivos.

7.7 Métodos de recuperación

1. Recuperación por metadatos

- Busca información en la tabla de archivos del sistema (FAT, MFT).
- Rápida y conserva nombres y rutas originales.

2. Recuperación por firmas (carving)

- Busca patrones binarios característicos de cada tipo de archivo (ej. cabecera JPEG **FFD8FFE0**).
- Útil cuando la tabla de archivos está dañada o no existe.

3. Recuperación manual

- Usada en casos complejos, donde el perito analiza sectores hexadecimales.
 - Herramientas como **WinHex** o **HxD** son imprescindibles.
-

7.8 Ejemplo práctico: recuperación en un caso real

Caso: Un pendrive de 16GB fue formateado accidentalmente. Procedimiento:

1. Adquisición forense del pendrive con **Guymager**.
 2. Análisis de la imagen con **PhotoRec** para buscar firmas de archivos JPEG y DOCX.
 3. Recuperación del 80% de los documentos, aunque algunos sin nombre original.
 4. Generación de hashes para cada archivo recuperado y documentación del proceso.
-

7.9 Documentación del proceso

Todo proceso de recuperación debe incluir:

- Identificación del dispositivo.
 - Herramientas y versiones usadas.
 - Procedimiento paso a paso.
 - Hashes de los archivos recuperados.
 - Limitaciones y observaciones (archivos corruptos, datos parciales).
-

7.10 Errores comunes

- Trabajar sobre el dispositivo original.
 - No inmovilizar el medio antes de la recuperación.
 - No calcular hashes de los archivos recuperados.
 - Sobrescribir espacio libre antes de realizar la recuperación.
-

Resumen del capítulo: La recuperación de archivos eliminados es una técnica esencial en informática forense, pero su éxito depende de la rapidez de actuación, el tipo de medio y la metodología aplicada. Documentar cada paso es tan importante como la recuperación misma.

Capítulo 8 – Análisis de Sistemas Operativos

8.1 Introducción

El **análisis de sistemas operativos** es una de las tareas centrales en la informática forense. Cada sistema guarda información en ubicaciones específicas que, correctamente interpretadas, pueden revelar:

- Actividad reciente del usuario.
- Archivos abiertos o borrados.
- Programas instalados y utilizados.

- Conexiones de red y dispositivos conectados.

Este capítulo cubre los artefactos más importantes de **Windows**, **Linux** y **macOS**, así como las técnicas y herramientas recomendadas para su análisis.

8.2 Análisis forense en Windows

Windows es el sistema más utilizado en entornos domésticos y corporativos, y por ello el más frecuente en investigaciones.

8.2.1 Artefactos clave

1. Registro de Windows (Windows Registry)

- Base de datos jerárquica con configuración del sistema, software y usuarios.
- Ubicación de archivos:
 - **C:\Windows\System32\config** (Hives del sistema).
 - **%USERPROFILE%\NTUSER.DAT** (Configuración del usuario).
- Información útil:
 - Dispositivos USB conectados.
 - Últimas aplicaciones abiertas.
 - Cuentas de usuario.

2. Archivos Prefetch

- Ubicación: **C:\Windows\Prefetch**
- Guardan datos de aplicaciones ejecutadas para mejorar su carga.
- Pueden indicar cuándo y cuántas veces se ejecutó un programa.

3. Visor de eventos (Event Logs)

- Ubicación: **C:\Windows\System32\winevt\Logs**
- Contienen registros de seguridad, sistema y aplicaciones.
- Útiles para identificar inicios de sesión, apagados forzados y errores.

4. Shadow Copies y Restauración del Sistema

- Permiten acceder a versiones previas de archivos.
- Herramienta recomendada: **Shadow Explorer**.

5. Archivos temporales y cachés

- Incluyen historial de navegación, cookies, caché DNS.
- Ubicaciones dependen del navegador y versión.

8.3 Análisis forense en Linux

Linux se utiliza en servidores, entornos de desarrollo y sistemas embebidos. Su estructura abierta facilita el acceso directo a los artefactos.

8.3.1 Artefactos clave

1. Archivos de log del sistema

- Ubicación: `/var/log/`
- Ejemplos:
 - `auth.log` → Inicios de sesión y autenticaciones.
 - `syslog` → Actividad general del sistema.
 - `dmesg` → Mensajes del kernel.

2. Historial de comandos

- Ubicación: `~/ .bash_history` (o equivalente según shell).
- Puede revelar comandos críticos ejecutados por el usuario.

3. Cron y tareas programadas

- Ubicación: `/etc/crontab` y `crontab -l` por usuario.
- Indican scripts y tareas automáticas ejecutadas.

4. Configuraciones de red

- Archivos en `/etc/network/interfaces` o `/etc/sysconfig/network-scripts/`.
- Pueden indicar cambios sospechosos en la configuración.

5. Metadatos de archivos

- Comando `stat` para obtener fecha de creación, modificación y acceso.

8.4 Análisis forense en macOS

macOS combina una base Unix con elementos propios de Apple, lo que requiere un enfoque mixto.

8.4.1 Artefactos clave

1. Time Machine (Copias de seguridad)

- Permite recuperar archivos y versiones anteriores.
- Ubicación por defecto: `/Volumes/Time Machine Backups/`.

2. Logs del sistema

- Ubicación: `/var/log/` y visualización con **Console.app**.
- Incluyen registros de kernel, aplicaciones y red.

3. Plist Files (Property List)

- Ubicación: `~/Library/Preferences/` y `/Library/Preferences/`.

- Contienen configuraciones de usuario y aplicación.

4. Spotlight Index

- Base de datos de búsqueda del sistema.
- Puede contener referencias a archivos eliminados.

5. Keychain (Llavero)

- Almacena contraseñas y certificados.
- Accesible con herramientas forenses y autorización adecuada.

8.5 Procedimiento general de análisis de un sistema operativo

1. Montar la imagen forense en modo solo lectura.
 2. Identificar el sistema operativo y versión.
 3. Ubicar y extraer artefactos relevantes.
 4. Analizar metadatos y contenido.
 5. Correlacionar con otros artefactos (por ejemplo, uso de USB + ejecución de un programa).
 6. Documentar cada hallazgo con ubicación, hash y relevancia.
-

8.6 Herramientas recomendadas

- **Windows:** FTK Imager, Registry Explorer, Magnet AXIOM.
 - **Linux:** Autopsy, Sleuth Kit, Log2Timeline.
 - **macOS:** BlackLight, mac_apt, fsevents_parser.
-

8.7 Ejemplo práctico: caso de robo de información en Windows

1. Imagen forense de un disco duro de un equipo corporativo.
 2. Prefetch revela ejecución repetida de **filezilla.exe**.
 3. Logs del firewall muestran conexiones salientes al puerto 21 (FTP).
 4. Registro de Windows indica conexión de un pendrive poco antes de la transferencia.
 5. Conclusión: Se usó un USB para cargar datos y luego transferirlos vía FTP.
-

Resumen del capítulo: Cada sistema operativo guarda rastros únicos de actividad que, correctamente analizados, pueden reconstruir eventos clave. El perito debe conocer las rutas y artefactos esenciales de Windows, Linux y macOS para maximizar la extracción de información útil.

Capítulo 9 – Forense en Dispositivos Móviles

9.1 Introducción

Los **dispositivos móviles** se han convertido en una fuente crítica de evidencia digital, ya que almacenan:

- Comunicaciones (mensajes, llamadas, chats).
- Archivos multimedia.
- Datos de geolocalización.
- Credenciales y aplicaciones bancarias.

El análisis forense de un teléfono o tablet puede ser determinante en casos de:

- Ciberacoso.
- Fraude financiero.
- Tráfico de información confidencial.
- Delitos contra la integridad sexual.

La complejidad de estos dispositivos radica en que:

- Se actualizan constantemente.
- Utilizan cifrado avanzado.
- Manejan almacenamiento interno y en la nube.

9.2 Métodos de extracción forense

9.2.1 Extracción lógica

- Copia de datos accesibles a través de APIs o herramientas autorizadas por el sistema operativo.
- Información recuperada: contactos, SMS, registros de llamadas, archivos visibles.
- Limitación: no accede a datos eliminados o cifrados.

9.2.2 Extracción física

- Copia bit a bit de la memoria del dispositivo.
- Incluye datos borrados y áreas protegidas.
- Requiere técnicas más invasivas y hardware especializado.

9.2.3 Extracción en vivo

- Se realiza con el dispositivo encendido para obtener datos volátiles.
- Útil para capturar sesiones abiertas, claves en memoria o chats activos.

9.2.4 Métodos especiales

- **Chip-off:** Retiro físico del chip de memoria NAND para su lectura directa.
- **JTAG:** Acceso a bajo nivel a través de interfaces de depuración.

9.3 Forense en Android

Android es el sistema operativo móvil más utilizado, lo que lo convierte en un objetivo frecuente.

9.3.1 Artefactos clave en Android

1. Base de datos de llamadas y SMS

- Ubicación típica: `/data/data/com.android.providers.telephony/databases/mmssms.db`

2. Chats de mensajería

- WhatsApp: `/data/data/com.whatsapp/databases/msgstore.db`
- Telegram: almacenamiento cifrado en `/data/data/org.telegram.messenger/files/`

3. Historial de ubicación

- Servicios de Google Location.
- Archivos `cache.cell` y `cache.wifi`.

4. Archivos multimedia

- Carpeta `/sdcard/DCIM/` y subcarpetas de apps.

5. Metadatos EXIF

- Incluyen fecha, hora, modelo del dispositivo y, a veces, coordenadas GPS.

9.4 Forense en iOS

iOS utiliza un ecosistema más cerrado, pero con un alto valor probatorio en sus datos.

9.4.1 Artefactos clave en iOS

1. Copia de seguridad iTunes

- Contiene la mayoría de datos del dispositivo, incluyendo mensajes, historial de llamadas y configuraciones.

2. Mensajes de iMessage y SMS

- Ubicación: `/private/var/mobile/Library/SMS/sms.db`

3. Fotos y videos

- Carpeta `/private/var/mobile/Media/DCIM/`

4. Datos de salud y actividad

- Ubicación: `/private/var/mobile/Library/Health/`

5. Historial de ubicación y Wi-Fi

- Archivos `consolidated.db` y `cache_encryptedA.db`.

9.5 Herramientas forenses para dispositivos móviles

- **Cellebrite UFED** – Estándar industrial, soporta extracción física y lógica.
- **Oxygen Forensic Detective** – Análisis profundo de aplicaciones y metadatos.
- **MOBILedit Forensic** – Amplio soporte de dispositivos.

- **Autopsy con módulos móviles** – Opción de código abierto.
 - **ADB (Android Debug Bridge)** – Acceso a datos en Android con depuración activada.
-

9.6 Procedimiento general de análisis

1. Asegurar el dispositivo

- Activar modo avión para evitar borrado remoto.
- Proteger batería y evitar que se apague (especialmente en cifrados).

2. Documentar estado inicial

- Fotografiar pantalla, conexiones y accesorios.
- Registrar versión del sistema operativo y número de serie.

3. Elegir método de extracción

- Segundo el modelo, versión y estado de bloqueo.

4. Realizar copia forense

- Guardar imagen con hashes para integridad.

5. Analizar datos

- Extraer información relevante de comunicaciones, ubicación y multimedia.

6. Documentar hallazgos

- Con capturas y referencias precisas de origen.
-

9.7 Ejemplo práctico: caso de acoso vía WhatsApp

- **Situación:** Denuncia por acoso digital a través de mensajes.
 - **Procedimiento:**
 1. Extracción física con Cellebrite UFED.
 2. Recuperación de mensajes eliminados en `msgstore.db`.
 3. Correlación con fechas y horas de capturas de pantalla provistas por la víctima.
 4. Documentación de metadatos que vinculan el número de origen con el acusado.
-

9.8 Desafíos en el análisis móvil

- Cifrado fuerte y bloqueo biométrico.
 - Funciones de borrado remoto.
 - Actualizaciones frecuentes del sistema.
 - Sincronización con la nube (requiere órdenes judiciales adicionales).
-

Resumen del capítulo: El análisis forense de dispositivos móviles requiere métodos adaptados a cada sistema, herramientas especializadas y una respuesta rápida para preservar datos volátiles. La información obtenida puede ser decisiva en investigaciones penales y corporativas.

Capítulo 10 – Análisis de Redes y Tráfico

10.1 Introducción

El **análisis forense de redes** consiste en la captura, preservación y examen del tráfico de datos que circula por una red con el fin de identificar:

- Intrusiones o accesos no autorizados.
- Transferencias de datos sensibles.
- Actividades maliciosas, como malware o ataques DDoS.
- Comunicación entre actores de un incidente.

A diferencia de otros tipos de evidencia, el tráfico de red es **altamente volátil** y requiere captura inmediata para no perder información.

10.2 Objetivos del análisis forense de redes

1. **Reconstruir eventos:** Determinar qué ocurrió, cuándo y desde dónde.
 2. **Identificar responsables:** Asociar direcciones IP, dispositivos y usuarios.
 3. **Preservar evidencia:** Guardar capturas en formato seguro y verificable.
 4. **Generar inteligencia:** Prevenir futuros incidentes.
-

10.3 Tipos de evidencia en redes

- **Capturas de paquetes (PCAP):** Contienen datos completos de comunicaciones.
 - **Registros de firewall:** Conexiones permitidas y bloqueadas.
 - **Logs de routers y switches:** Información de tráfico y cambios de configuración.
 - **Registros de servidores:** Actividad de servicios como web, correo o FTP.
 - **Alertas de sistemas IDS/IPS:** Señales de ataques o anomalías.
-

10.4 Captura de tráfico de red

10.4.1 Métodos de captura

- **En el punto final:** Captura desde el equipo investigado.
- **En el perímetro:** Captura en firewalls o routers.
- **SPAN/Port Mirroring:** Copia de todo el tráfico que pasa por un puerto de switch.
- **Taps de red:** Dispositivos que duplican tráfico para análisis.

10.4.2 Herramientas de captura

- **Wireshark** – Captura y análisis gráfico.
- **tcpdump** – Captura por línea de comandos en Unix/Linux.
- **NetworkMiner** – Extracción de archivos y metadatos desde PCAP.
- **Zeek (Bro)** – Monitoreo avanzado y scripting de detección.

Ejemplo de captura con tcpdump:

```
tcpdump -i eth0 -w captura.pcap
```

10.5 Análisis de capturas

10.5.1 Identificación de conexiones sospechosas

- Revisar direcciones IP de origen/destino.
- Analizar puertos inusuales o conexiones frecuentes.
- Verificar protocolos utilizados.

10.5.2 Reconstrucción de sesiones

- Reensamblar conversaciones TCP.
- Identificar contenido transferido (archivos, credenciales).
- Herramientas: Wireshark, NetworkMiner.

10.5.3 Detección de malware en tráfico

- Firmas conocidas en payloads.
- Conexiones a dominios de comando y control (C2).
- Consultar bases como VirusTotal o AbuseIPDB.

10.6 Análisis forense de ataques comunes

1. Escaneo de puertos

- Patrón: múltiples conexiones a distintos puertos en corto tiempo.
- Herramienta de detección: Zeek.

2. Ataques de fuerza bruta

- Múltiples intentos fallidos de autenticación.
- Revisar logs de SSH, RDP, FTP.

3. Exfiltración de datos

- Transferencias grandes hacia direcciones externas.
- Protocolos como FTP, HTTP o HTTPS.

4. Ataques Man-in-the-Middle

- Análisis de ARP spoofing.
 - Comprobación de certificados HTTPS falsos.
-

10.7 Preservación de capturas y logs

- Guardar archivos PCAP en medios de solo lectura.
 - Calcular hashes (MD5, SHA-256) para garantizar integridad.
 - Mantener copias en ubicaciones seguras.
 - Documentar fecha, hora y condiciones de captura.
-

10.8 Ejemplo práctico: investigación de exfiltración de datos

1. **Contexto:** Una empresa detecta pérdida de archivos confidenciales.
 2. **Captura:** PCAP de 48 horas en el firewall perimetral.
 3. **Análisis:**
 - Filtrado de tráfico por puerto 21 (FTP).
 - Reconstrucción de sesiones muestra transferencia de archivos [.zip](#).
 - Dirección IP destino registrada en un país extranjero.
 4. **Conclusión:** Transferencia no autorizada ejecutada desde una estación interna fuera de horario laboral.
-

10.9 Herramientas recomendadas para análisis forense de redes

- **Wireshark** – Captura y análisis interactivo.
 - **NetworkMiner** – Extracción de archivos, imágenes y certificados.
 - **Zeek (Bro)** – Monitoreo y registro de eventos.
 - **Security Onion** – Distribución Linux para monitoreo y análisis forense.
 - **Suricata** – IDS/IPS de alto rendimiento.
-

10.10 Errores comunes

- No sincronizar relojes de captura y sistemas (problemas de correlación).
 - Capturar tráfico sin permisos legales.
 - Analizar solo tráfico claro, ignorando conexiones cifradas (TLS, VPN).
 - No filtrar el tráfico, generando archivos PCAP demasiado grandes e inmanejables.
-

Resumen del capítulo: El análisis forense de redes es clave para detectar y documentar incidentes. La captura rápida, el filtrado eficiente y la preservación adecuada son esenciales para que la evidencia sea válida y útil.

**Capítulo 11 – Esteganografía y Ocultación de Información

11.1 Introducción

La **esteganografía** es el arte y la ciencia de ocultar información dentro de otros archivos o medios de forma que su presencia pase desapercibida. A diferencia de la **criptografía**, que oculta el contenido pero no su existencia, la esteganografía busca **ocultar la existencia misma de la comunicación**.

En el ámbito de la informática forense, es fundamental:

- Saber identificar cuándo se ha utilizado.
 - Poder extraer la información oculta.
 - Documentar el proceso para su uso como evidencia.
-

11.2 Usos legítimos y maliciosos

- **Usos legítimos:** Protección de marcas de agua, firma digital de imágenes, control de derechos de autor.
- **Usos maliciosos:** Ocultar datos robados, comunicación encubierta entre criminales, transmisión de malware.

Ejemplo real: En 2010, un grupo de espionaje internacional ocultó mensajes en imágenes publicadas en foros de internet para coordinar operaciones.

11.3 Tipos de esteganografía

11.3.1 En imágenes

- Método más común: modificación de los bits menos significativos (**LSB – Least Significant Bit**).
- Puede ocultar texto o incluso otros archivos completos.
- Formatos habituales: BMP, PNG, JPEG.

11.3.2 En audio

- Modificación de frecuencias o amplitudes imperceptibles para el oído humano.
- Métodos: codificación de eco, modulación de fase, LSB en audio digital.

11.3.3 En vídeo

- Combina técnicas de imagen y audio.
- Mayor capacidad de ocultación.

11.3.4 En texto

- Uso de espacios, caracteres invisibles o variaciones tipográficas para ocultar datos.

- Ejemplo: añadir espacios dobles en posiciones específicas que representan bits.

11.3.5 En tráfico de red

- Encapsular datos en cabeceras de protocolos o en campos no utilizados.
- Método frecuente en malware para exfiltrar información sin ser detectado.

11.4 Detección de esteganografía

Detectar esteganografía es un reto porque no siempre hay alteraciones visibles o audibles. Los métodos de detección incluyen:

1. Análisis visual

- Comparación de imágenes sospechosas con sus originales.
- Uso de filtros y análisis de histograma.

2. Análisis estadístico

- Estudio de patrones en la distribución de bits.
- Herramientas como Stegdetect para JPEG.

3. Extracción forense

- Uso de software especializado para intentar recuperar datos ocultos.

11.5 Herramientas para esteganografía y su detección

Para ocultar datos

- **OpenStego** – Interfaz sencilla, soporta imágenes.
- **Steghide** – Trabaja con imágenes y audio.
- **SilentEye** – Interfaz gráfica multiplataforma.

Para detectar y extraer

- **Stegdetect** – Detecta esteganografía en imágenes JPEG.
- **Stegsolve** – Análisis visual por capas de color.
- **zsteg** – Detección de LSB en PNG y BMP.
- **binwalk** – Extracción de archivos embebidos en binarios.

Ejemplo de extracción con Steghide:

```
steghide extract -sf imagen.jpg
```

11.6 Procedimiento forense para casos de esteganografía

1. Identificación

- Analizar si el archivo tiene un tamaño inusual o metadatos extraños.

2. Preservación

- Guardar copia forense y calcular hashes.

3. Análisis preliminar

- Inspección visual o auditiva.
- Comparación con versión original si está disponible.

4. Análisis técnico

- Uso de herramientas de detección.
- Extracción de posibles datos.

5. Documentación

- Descripción del proceso, herramientas usadas, resultados y hashes.

11.7 Ejemplo práctico: extracción de mensaje oculto en una imagen PNG

Caso: Una imagen enviada por correo electrónico se sospecha que contiene datos ilegales.

Procedimiento:

1. **Hash inicial:** SHA-256 calculado y registrado.

2. **Análisis de tamaño:** Imagen de 500 KB, inusualmente grande para su resolución.

3. **Uso de zsteg:**

```
zsteg imagen.png
```

4. Resultado: Cadena de texto codificada en Base64 recuperada.

5. Decodificación revela lista de direcciones IP y contraseñas.

11.8 Errores comunes en la detección

- Suponer que solo las imágenes pueden contener esteganografía.
- Usar solo una herramienta de detección.
- No preservar el archivo original antes de intentar extraer datos.
- Alterar el archivo sospechoso durante el análisis.

Resumen del capítulo: La esteganografía es una técnica poderosa para ocultar información, tanto para fines legítimos como maliciosos. El perito debe conocer los métodos más comunes, las herramientas de detección y los pasos forenses para preservar y extraer datos sin comprometer la evidencia.

Capítulo 12 – Criptografía y Forense

12.1 Introducción

La **criptografía** es el conjunto de técnicas destinadas a proteger la información mediante su transformación en un formato ilegible para quien no posea la clave adecuada. En informática forense, la criptografía aparece en dos contextos:

1. **Protección legítima:** datos cifrados para resguardar la privacidad o cumplir regulaciones.
2. **Uso malicioso:** cifrado empleado por criminales para ocultar evidencias, comunicaciones o implantar ransomware.

El desafío del perito forense es **acceder legalmente a datos cifrados**, garantizando la integridad de la evidencia y respetando el marco legal.

12.2 Tipos de criptografía relevantes para la informática forense

12.2.1 Criptografía simétrica

- Usa la **misma clave** para cifrar y descifrar.
- Ejemplo: AES (Advanced Encryption Standard), DES, 3DES.
- Ventajas: rápida y eficiente.
- Desventaja: el intercambio seguro de la clave es crítico.

12.2.2 Criptografía asimétrica

- Usa un **par de claves**: pública para cifrar, privada para descifrar.
- Ejemplo: RSA, ECC.
- Ventajas: intercambio seguro de información sin compartir clave privada.
- Desventaja: más lenta que la simétrica.

12.2.3 Hashing

- Función unidireccional que genera un valor único a partir de datos.
- Ejemplo: MD5, SHA-256.
- Usos forenses: verificar integridad de evidencia.

12.2.4 Firmas digitales

- Validan la autenticidad y la integridad de un mensaje o archivo.
- Utilizan criptografía asimétrica.

12.3 Casos forenses comunes con criptografía

1. **Discos duros cifrados** (BitLocker, VeraCrypt, FileVault).
 2. **Archivos cifrados** con contraseñas (ZIP, RAR, 7z).
 3. **Mensajería cifrada** (WhatsApp, Signal, Telegram).
 4. **Cifrado en malware** (ransomware).
 5. **Cifrado en bases de datos** (MySQL, SQL Server con TDE).
-

12.4 Técnicas de acceso a datos cifrados

12.4.1 Obtención de claves

- Buscar claves almacenadas en el sistema (archivos de configuración, memoria RAM).
- Herramientas: **Volatility, Mimikatz**.

12.4.2 Ataques de fuerza bruta

- Probar todas las combinaciones posibles hasta encontrar la clave.
- Herramientas: **John the Ripper, Hashcat**.

12.4.3 Ataques de diccionario

- Usar listas de contraseñas comunes.
- Mucho más rápidos que fuerza bruta pura.

12.4.4 Ingeniería social

- Obtener la clave a través de engaño o interacción directa (en un contexto legal autorizado).

12.4.5 Análisis de memoria volátil

- Extraer claves temporales almacenadas en RAM mientras el dispositivo está encendido.
-

12.5 Manejo forense de discos cifrados

1. Identificación del cifrado

- Reconocer si se usa BitLocker, VeraCrypt, FileVault, LUKS, etc.
- Herramientas: **Dislocker** para BitLocker, **cryptsetup** para LUKS.

2. Preservación

- Crear imagen bit a bit si es posible (antes de bloqueo).
- Guardar metadatos y sectores de arranque.

3. Obtención de claves

- Solicitar clave a propietario o autoridad competente.
- Buscar en memoria si el equipo está encendido.

4. Montaje y análisis

- Montar volumen cifrado en modo solo lectura.
 - Trabajar sobre copia forense.
-

12.6 Ejemplo práctico: recuperación de datos de un disco BitLocker

1. El disco se obtuvo encendido, por lo que la clave estaba en memoria.
 2. Con **Belkasoft RAM Capturer** se extrajo la memoria RAM.
 3. **Volatility** localizó la clave de BitLocker.
 4. El volumen fue montado en modo solo lectura con **Dislocker**.
 5. Se realizó la copia forense completa para análisis posterior.
-

12.7 Herramientas recomendadas

- **Hashcat** – Ataques de fuerza bruta y diccionario sobre hashes.
 - **John the Ripper** – Cracking de contraseñas y hashes.
 - **Volatility** – Análisis de memoria RAM.
 - **Mimikatz** – Extracción de credenciales en Windows.
 - **Dislocker** – Montaje de volúmenes BitLocker.
 - **Cryptsetup** – Manejo de LUKS en Linux.
-

12.8 Aspectos legales y éticos

- El acceso a datos cifrados sin autorización judicial puede ser ilegal.
 - Toda acción debe ser documentada, incluyendo la obtención de claves.
 - La intervención de cifrados debe justificarse en informes técnicos claros.
 - Evitar romper cifrados que no estén directamente relacionados con el caso.
-

Resumen del capítulo: La criptografía es una barrera importante para el análisis forense, pero con las herramientas y procedimientos adecuados es posible acceder a datos cifrados de forma legal y segura. La clave está en actuar rápido, preservar la evidencia y documentar cada paso.

Capítulo 13 – Herramientas Profesionales de Informática Forense

13.1 Introducción

El trabajo del perito forense se apoya fuertemente en **herramientas especializadas** que permiten adquirir, analizar y presentar evidencia digital de forma eficiente y conforme a estándares internacionales. La elección de la herramienta depende de:

- Tipo de dispositivo a analizar.

- Presupuesto disponible.
- Necesidad de certificaciones o validaciones judiciales.
- Nivel de profundidad requerido.

En este capítulo se describen las **principales soluciones** del sector, tanto **comerciales** como **de código abierto**, y se indican sus usos más comunes.

13.2 Herramientas comerciales

13.2.1 EnCase Forensic

- **Descripción:** Herramienta líder en el sector, desarrollada por OpenText.
- **Funciones:**
 - Adquisición forense de discos y dispositivos.
 - Análisis de archivos, correo electrónico y sistemas de archivos.
 - Soporte para múltiples formatos de imagen forense (.E01, .AFF, .RAW).
- **Ventajas:** Altamente aceptada en tribunales, amplia documentación.
- **Desventajas:** Licencia costosa.
- **Caso de uso:** Investigación corporativa con múltiples discos y correos electrónicos.

13.2.2 FTK (Forensic Toolkit)

- **Descripción:** Suite de AccessData para análisis de evidencia digital.
- **Funciones:**
 - Indexación rápida para búsqueda avanzada.
 - Recuperación de archivos eliminados.
 - Análisis de contraseñas y datos cifrados.
- **Ventajas:** Muy rápido en búsquedas.
- **Desventajas:** Requiere hardware potente.
- **Caso de uso:** Análisis de gran volumen de correos electrónicos.

13.2.3 X-Ways Forensics

- **Descripción:** Herramienta forense alemana, muy ligera y potente.
- **Funciones:**
 - Adquisición y análisis forense.
 - Soporte para sistemas de archivos menos comunes.
 - Análisis de metadatos y recuperación de datos.
- **Ventajas:** Bajo consumo de recursos, portátil.

- **Desventajas:** Interfaz menos amigable.
- **Caso de uso:** Pericias en campo con recursos limitados.

13.2.4 Magnet AXIOM

- **Descripción:** Herramienta muy usada para análisis de móviles, nube y PC.
- **Funciones:**
 - Recuperación de mensajes, historial web, redes sociales.
 - Integración con bases de datos OSINT.
- **Ventajas:** Excelente para datos de redes sociales y mensajería.
- **Desventajas:** Licencia elevada.
- **Caso de uso:** Casos de ciberacoso y delitos sexuales.

13.2.5 Cellebrite UFED

- **Descripción:** Referencia en análisis forense de dispositivos móviles.
- **Funciones:**
 - Extracción física, lógica y en vivo.
 - Recuperación de datos borrados en Android e iOS.
- **Ventajas:** Altísima tasa de recuperación en móviles.
- **Desventajas:** Licencia muy costosa y acceso restringido a organismos autorizados.
- **Caso de uso:** Investigación penal con dispositivos incautados.

13.3 Herramientas de código abierto y gratuitas

13.3.1 Autopsy / Sleuth Kit

- **Descripción:** Plataforma open source para análisis forense de discos.
- **Funciones:**
 - Análisis de imágenes forenses.
 - Extracción de artefactos de sistemas operativos.
- **Ventajas:** Gratuito, comunidad activa.
- **Desventajas:** Menos soporte oficial que herramientas comerciales.
- **Caso de uso:** Laboratorios académicos y casos con bajo presupuesto.

13.3.2 Volatility

- **Descripción:** Framework de análisis de memoria RAM.
- **Funciones:**
 - Recuperación de procesos activos.
 - Extracción de claves y datos volátiles.
- **Ventajas:** Muy potente y flexible.
- **Desventajas:** Requiere experiencia en línea de comandos.
- **Caso de uso:** Análisis de malware y ataques en vivo.

13.3.3 Wireshark

- **Descripción:** Herramienta de captura y análisis de tráfico de red.
- **Funciones:**
 - Filtrado de protocolos.
 - Reconstrucción de sesiones.
- **Ventajas:** Gratuito, muy documentado.
- **Desventajas:** No cifra ni preserva automáticamente.
- **Caso de uso:** Investigación de intrusión por red.

13.3.4 TestDisk / PhotoRec

- **Descripción:** Herramientas para recuperación de particiones y archivos borrados.
- **Funciones:**
 - TestDisk: Recuperación de particiones.
 - PhotoRec: Recuperación por firmas de archivos.
- **Ventajas:** Gratuito, multiplataforma.
- **Desventajas:** Interfaz poco intuitiva.
- **Caso de uso:** Recuperación rápida de datos en campo.

13.3.5 OSForensics (versión gratuita)

- **Descripción:** Suite de análisis de archivos, correo electrónico y metadatos.
- **Funciones:**
 - Indexación rápida de datos.
 - Extracción de datos de navegadores.
- **Ventajas:** Interfaz amigable.
- **Desventajas:** Versión gratuita con funciones limitadas.

- **Caso de uso:** Casos pequeños o formación.
-

13.4 Criterios para elegir una herramienta

1. **Compatibilidad con el tipo de evidencia** (discos, móviles, nube, RAM).
 2. **Admisibilidad en tribunales** (herramientas reconocidas en el sector).
 3. **Presupuesto disponible** (licencia vs. open source).
 4. **Facilidad de uso** y curva de aprendizaje.
 5. **Disponibilidad de soporte técnico** y actualizaciones.
-

13.5 Ejemplo práctico: combinación de herramientas en un caso real

Caso: Robo de información en empresa.

1. **FTK Imager** para adquisición de discos.
 2. **Autopsy** para análisis de artefactos del sistema operativo.
 3. **Wireshark** para examinar tráfico de exfiltración.
 4. **Volatility** para extraer credenciales desde memoria RAM.
 5. **Magnet AXIOM** para revisar comunicaciones en aplicaciones.
-

Resumen del capítulo: Existen herramientas comerciales muy potentes y ampliamente reconocidas judicialmente, así como opciones de código abierto capaces de realizar análisis completos. La clave es elegir la combinación adecuada para cada caso, maximizando la eficacia y minimizando costos.

Capítulo 14 – Procedimientos Paso a Paso

14.1 Introducción

La teoría forense es imprescindible, pero lo que realmente define el éxito de una investigación es **la correcta ejecución de los procedimientos prácticos**. Este capítulo presenta **cuatro guías paso a paso** para casos reales que un perito puede enfrentar:

1. Análisis de un **pendrive sospechoso**.
2. Recuperación de datos de un **SSD**.
3. Extracción forense de un **teléfono Android**.
4. Investigación de un **caso de ransomware** en una red corporativa.

Cada procedimiento se acompaña de:

- **Herramientas recomendadas.**
 - **Ejemplos de comandos.**
 - **Buenas prácticas.**
 - **Errores comunes a evitar.**
-

14.2 Procedimiento 1 – Análisis forense de un pendrive sospechoso

Escenario: Un pendrive de 32 GB es incautado a un empleado sospechado de filtrar información.

Objetivo: Determinar si el dispositivo contiene datos relevantes para el caso, incluyendo archivos borrados.

14.2.1 Preparación

- **Herramientas:**

- Bloqueador de escritura USB (hardware).
- Guymager o FTK Imager para adquisición.
- Autopsy/Sleuth Kit para análisis.

- **Documentación inicial:**

- Marca, modelo, número de serie, capacidad.
- Estado físico y observaciones.

14.2.2 Adquisición forense

1. Conectar el pendrive al bloqueador de escritura.

2. Crear imagen forense en formato **.E01**:

```
guymager
```

3. Calcular y registrar hash SHA-256 de la imagen.

14.2.3 Análisis

- Montar la imagen en modo solo lectura.

- Buscar:

- Archivos activos.
- Archivos borrados (con TestDisk/PhotoRec).
- Metadatos sospechosos (fechas, autoría).

- Revisar la presencia de malware.

14.2.4 Informe

- Listado de archivos relevantes.
- Hashes de cada evidencia.
- Capturas de pantalla de hallazgos clave.

Error común: Conectar el pendrive directamente al PC sin write-blocker, alterando fechas de acceso.

14.3 Procedimiento 2 – Recuperación de datos de un SSD

Escenario: SSD de 512 GB perteneciente a una computadora corporativa formateada recientemente.

Objetivo: Recuperar la mayor cantidad posible de información, pese a las limitaciones del comando TRIM.

14.3.1 Preparación

- **Herramientas:**
 - Bloqueador de escritura SATA/NVMe.
 - FTK Imager o dd para copia bit a bit.
 - R-Studio para recuperación avanzada.
- **Consideración:** TRIM puede haber eliminado datos irrecuperables.

14.3.2 Adquisición

1. Conectar el SSD al bloqueador de escritura.
2. Adquirir imagen:

```
sudo dd if=/dev/nvme0n1 of=ssd_copia.dd bs=4M status=progress
```

3. Calcular hash SHA-256 y registrarlo.

14.3.3 Recuperación

- Usar R-Studio o PhotoRec en la imagen.
- Recuperar por firmas de archivo (carving).
- Documentar nombres originales si es posible.

14.3.4 Informe

- Archivos recuperados clasificados por tipo.
- Porcentaje de recuperación.
- Limitaciones encontradas.

Error común: No aislar el SSD inmediatamente después de la incautación, permitiendo que TRIM borre más bloques.

14.4 Procedimiento 3 – Extracción forense de un teléfono Android

Escenario: Teléfono Android involucrado en un caso de acoso digital.

Objetivo: Extraer mensajes de WhatsApp y ubicación GPS de forma forense.

14.4.1 Preparación

- **Herramientas:**
 - Cellebrite UFED o MOBILedit Forensic.

- ADB (Android Debug Bridge) para extracción básica.
- **Precauciones:**
 - Activar modo avión.
 - Evitar desbloqueos que alteren datos.

14.4.2 Adquisición

- Si está desbloqueado:

```
adb pull /sdcard/ /evidencia/
```

- Si está bloqueado: usar extracción física con Cellebrite.

14.4.3 Análisis

- Localizar base de datos `msgstore.db` de WhatsApp.
- Usar herramientas para descifrar mensajes si es necesario.
- Extraer historial GPS de Google Location Services.

14.4.4 Informe

- Mensajes relevantes con fecha y hora.
- Mapas de ubicación.
- Capturas de pantallas clave.

Error común: No registrar versión del sistema operativo y modelo del dispositivo antes de iniciar.

14.5 Procedimiento 4 – Investigación de ransomware en red corporativa

Escenario: Empresa reporta cifrado masivo de archivos en sus servidores.

Objetivo: Identificar origen del ataque, vector de entrada y posibles archivos sin cifrar.

14.5.1 Preparación

- **Herramientas:**
 - Wireshark para tráfico de red.
 - Volatility para memoria RAM.
 - Autopsy para análisis de discos.
- **Acciones inmediatas:**
 - Aislar equipos afectados.
 - Capturar memoria RAM antes de apagar sistemas.

14.5.2 Análisis de red

- Revisar PCAP para detectar conexiones sospechosas.
- Identificar direcciones IP de C2 (Command and Control).

14.5.3 Análisis de discos

- Localizar notas de rescate y ejecutables maliciosos.
- Extraer claves de cifrado si están en memoria.

14.5.4 Recuperación

- Verificar si existen copias sombra (Shadow Copies).
- Intentar restaurar archivos no cifrados.

14.5.5 Informe

- Línea de tiempo del ataque.
- IP de origen y posible actor.
- Estado de recuperación de datos.

Error común: Reiniciar servidores antes de capturar memoria, perdiendo claves de cifrado.

Resumen del capítulo: Estos procedimientos ofrecen guías prácticas y reproducibles para distintos escenarios forenses. Documentar cada paso, mantener la integridad de la evidencia y actuar con rapidez son claves para el éxito de la investigación.

Capítulo 15 – Estudio de Casos Reales

15.1 Introducción

El estudio de casos reales permite ver **cómo la teoría se convierte en resultados** en investigaciones concretas. En este capítulo revisaremos **cuatro casos relevantes** que abarcan distintos tipos de incidentes:

1. Fraude bancario digital.
2. Filtración de datos en una empresa multinacional.
3. Ciberacoso y delito contra la intimidad.
4. Investigación de un ransomware en una institución pública.

Cada caso incluye:

- Contexto y descripción del incidente.
- Procedimientos forenses aplicados.
- Herramientas utilizadas.
- Resultados obtenidos.
- Lecciones aprendidas.

15.2 Caso 1 – Fraude bancario digital

15.2.1 Contexto

Un cliente denuncia movimientos bancarios no autorizados en su cuenta, por un total de USD 18.000. El banco sospecha de **phishing** y solicita una investigación forense.

15.2.2 Investigación

1. Análisis del equipo del cliente

- Se realizó una adquisición forense del disco duro y memoria RAM.
- Herramientas: FTK Imager, Volatility.
- Se halló un archivo malicioso en la carpeta de descargas.

2. Examen de tráfico de red

- Captura de red mostró conexiones a un dominio registrado en Europa del Este.
- Herramientas: Wireshark, Zeek.

3. Análisis de correo electrónico

- El cliente recibió un email falso del banco solicitando "verificación de datos".
- El enlace redirigía a un sitio web idéntico al original pero con URL diferente.

15.2.3 Resultados

- Se identificó el malware como un **troyano bancario** que capturaba credenciales.
- Se rastreó el dominio al proveedor de hosting, que colaboró para su baja.
- Se preservaron logs y PCAPs como evidencia para la causa judicial.

15.2.4 Lecciones aprendidas

- La ingeniería social sigue siendo el vector más común.
- El análisis de memoria RAM puede revelar procesos activos invisibles al antivirus.

15.3 Caso 2 – Filtración de datos en empresa multinacional

15.3.1 Contexto

Una empresa de tecnología detecta que **bases de datos con información de clientes** aparecieron a la venta en foros de la dark web.

15.3.2 Investigación

1. Auditoría interna

- Revisión de logs de acceso a servidores.
- Herramientas: Splunk, ELK Stack.

2. Análisis de endpoints sospechosos

- Se detectó que un ingeniero descargó grandes volúmenes de datos fuera de horario laboral.
- Los artefactos Prefetch y logs confirmaron la ejecución de un cliente FTP.

3. Correlación con tráfico de red

- Wireshark mostró transferencias a un servidor en Singapur.
- El servidor de destino fue identificado y se solicitó cooperación internacional.

15.3.3 Resultados

- El ingeniero admitió haber vendido la base de datos.
- La evidencia fue presentada en formato aceptado judicialmente (.E01 y PCAP).

15.3.4 Lecciones aprendidas

- La correlación de artefactos de sistema y tráfico de red es clave.
- La monitorización continua podría haber detectado la exfiltración antes.

15.4 Caso 3 – Ciberacoso y delito contra la intimidad

15.4.1 Contexto

Una víctima denuncia que fotos íntimas suyas circulan en redes sociales sin su consentimiento.

15.4.2 Investigación

1. Extracción forense de dispositivos de la víctima

- Herramientas: Cellebrite UFED, MOBILedit Forensic.
- Se recuperaron mensajes de WhatsApp y Telegram eliminados.

2. Análisis de metadatos de imágenes

- EXIF mostró que las fotos fueron tomadas con un iPhone 13.
- Coincidía con el dispositivo del sospechoso.

3. Rastreo en redes sociales

- Análisis OSINT identificó perfiles falsos que compartían las imágenes.
- Los registros IP de Facebook y Twitter apuntaban a la misma conexión doméstica.

15.4.3 Resultados

- Se vinculó directamente al sospechoso mediante su IP y metadatos.
- Las imágenes fueron eliminadas de la red mediante gestión judicial.

15.4.4 Lecciones aprendidas

- OSINT es una herramienta poderosa para complementar análisis técnicos.
- El metadato EXIF sigue siendo oro puro en forense digital.

15.5 Caso 4 – Ransomware en una institución pública

15.5.1 Contexto

Un organismo gubernamental sufre un cifrado masivo de archivos que paraliza sus operaciones.

15.5.2 Investigación

1. Aislamiento y preservación

- Servidores desconectados de la red.
- Captura de memoria RAM para intentar obtener claves.

2. Análisis de red

- PCAP mostró conexiones cifradas hacia un servidor Tor.
- Identificación de patrones de tráfico de ransomware **LockBit**.

3. Análisis de discos

- Localización de binarios maliciosos y nota de rescate.
- Se descubrió que la intrusión inicial fue vía RDP con credenciales robadas.

4. Recuperación

- Gracias a Shadow Copies, se restauró el 70% de los archivos.
- Se emitió alerta a nivel nacional para prevenir ataques similares.

15.5.3 Resultados

- Vector de entrada identificado y bloqueado.
- Implementación de MFA (autenticación multifactor) en toda la institución.

15.5.4 Lecciones aprendidas

- Capturar la RAM antes de reiniciar puede salvar claves de cifrado.
- La segmentación de red es crítica para contener ataques.

15.6 Síntesis de patrones comunes

En todos los casos:

- La **preservación de evidencia** fue clave.
- Las herramientas **combinadas** dieron mejores resultados que una sola.
- La **documentación detallada** permitió que la evidencia fuera aceptada en juicio.

Resumen del capítulo: Los casos reales muestran que la informática forense no es solo técnica, sino también estrategia, legalidad y rapidez de acción. El éxito de una investigación depende de la correcta combinación de procedimientos, herramientas y colaboración entre especialistas.

Capítulo 16 – Forense en la Nube

16.1 Introducción

La computación en la nube cambió el terreno del DFIR (Digital Forensics & Incident Response): **recursos elásticos, efímeros y multi-inquilino**, registros distribuidos y **controles de seguridad administrados por terceros**. Investigar en cloud no es “montar un disco y leerlo”: es **reconstruir eventos** a partir de **logs, metadatos, instantáneas y artefactos API** bajo un régimen de **responsabilidad compartida y jurisdicciones cruzadas**.

Claves mentales del perito en cloud:

- *Todo es API.* Lo que no puedas pedir por consola/CLI/SDK, no existe para ti en ese momento.
 - *Los datos son volátiles.* Instancias que desaparecen, contenedores efímeros, funciones serverless.
 - *La evidencia son logs + snapshots + objetos inmutables.*
 - *Legal primero.* Ten clara la **competencia, el país de residencia de los datos** y las **condiciones del proveedor**.
-

16.2 Modelos de servicio y qué evidencia puedes obtener

- **IaaS (Infraestructura):** Tú gestionas SO y aplicaciones. Evidencia típica: **snapshots de discos (EBS/Disk/Persistent Disk), imágenes, volumetría de red (VPC Flow Logs), CloudTrail/Audit/Activity Logs, claves KMS, metadatos de instancias**, dumps de RAM (si los preparaste).
- **PaaS (Plataforma):** El proveedor gestiona SO. Evidencia: **logs de la plataforma** (funciones, App Services, Cloud Run), **config e IAM**, artefactos del servicio (bases de datos administradas, colas, topics).
- **SaaS (Aplicación):** Tú consumes. Evidencia: **registros de auditoría, exportaciones nativas, eDiscovery/Legal Hold**, metadatos de archivos/correos/chats.

Regla de oro: cuanto más “alto” el modelo (SaaS), **menos forense tradicional** y más **eDiscovery y auditoría**.

16.3 Responsabilidad compartida (resumen operativo)

- **Proveedor:** seguridad del *hardware, data centers, hipervisor*, ciertos logs de control, disponibilidad.
 - **Cliente:** *configuración segura, identidades, claves, red, logging, retención, respuesta a incidentes.*
 - Para DFIR: el **cliente debe haber habilitado los logs** correctos **antes** del incidente (si no, el vacío probatorio es real).
-

16.4 Marco legal y jurisdicción

- **Residencia de datos:** regiones y zonas determinan qué leyes aplican.
- **Multi-inquilino:** no podrás “entrar al host” del proveedor. Evidencia a través de **APIs y exportaciones certificadas**.
- **Órdenes y preservación:** usa **Legal Hold/retenciones inmutables** (p. ej., M365/Google Vault, S3 Object Lock).

- **Transferencias internacionales:** coordina con asesoría legal por **acuerdos de cooperación** y políticas del proveedor.
 - **Trazabilidad:** documenta *quién solicitó qué* a la API, con *tokens, scopes, tiempos y hashes* de artefactos exportados.
-

16.5 Preservación de evidencia en cloud

Estrategias por capa:

- **Discos de VM (IaaS):** crea **snapshots** inmutables, clona a un proyecto/cuenta forense.
- **Buckets/objetos:** habilita **versionado** y **Object Lock/retention** (modo gobernanza/compliance).
- **Logs:** exporta a almacenamiento **WORM** (Write Once Read Many) o a **SIEM** con **retención extendida**.
- **SaaS:** aplica **Legal Hold** y exporta **con verificadores de integridad**.

Checklist rápido de preservación:

1. Congela la escena: revoca claves comprometidas, **sin apagar** todo de golpe si pierdes telemetría.
 2. **Snapshot/export** inmediata de recursos clave.
 3. Eleva **retención de logs** temporalmente.
 4. **Copia forense** a un *tenant/proyecto aislado*.
 5. Calcula **hashes** y guarda **manifiestos** (quién, cuándo, cómo).
-

16.6 Telemetría esencial por proveedor

AWS

- **CloudTrail:** auditoría de llamadas API (quién, qué, desde dónde).
- **CloudWatch Logs & Metrics:** logs de apps y servicios.
- **VPC Flow Logs:** flujo de red por interfaz/subred/VPC.
- **S3 Server/Access Logs & Object-level logging:** accesos a objetos.
- **GuardDuty/Inspector/Security Hub:** hallazgos de seguridad.
- **ELB/ALB/NLB Logs, WAF Logs, EKS audit logs:** capa perimetral y Kubernetes.
- **KMS logs:** uso de claves.

Azure

- **Azure Activity Log:** acciones sobre recursos del tenant.
- **Azure AD (Entra) Sign-in/Audit Logs:** autenticaciones y cambios de directorio.
- **Microsoft Defender for Cloud / Sentinel:** detección y correlación.
- **Log Analytics / Azure Monitor:** lago de logs y consultas Kusto.
- **NSG Flow Logs, App Gateway/Firewall Logs:** red/perímetro.

GCP

- **Cloud Audit Logs (Admin/Access/Data):** uso de APIs y acceso a datos.
- **VPC Flow Logs:** flujo de red.
- **Cloud Logging/Monitoring:** centralización y métricas.
- **Event Threat Detection / Security Command Center:** hallazgos.

- **KMS (Cloud KMS) logs:** uso de llaves.

SaaS frecuentes

- **Microsoft 365 (Exchange/SharePoint/Teams):** Unified Audit Log, eDiscovery (Purview), holds y export.
- **Google Workspace (Gmail/Drive/Chat):** Admin & Drive audit, Google Vault (holds/export).
- **Dropbox Business / Box / Slack / GitHub / Okta:** Audit Logs y APIs de export.
- **Correo y colaboración:** Message trace, sharing logs, DLP events, OAuth app grants.

16.7 Playbooks paso a paso (operativos)

Playbook A – Compromiso de cuenta e IAM en AWS

Escenario: claves de un usuario IAM filtradas; se sospecha acceso indebido a S3/EC2.

1. Contención mínima viable

- Desactiva/rota **Access Keys** comprometidas.
- Aplica **MFA** y revisa **políticas** del usuario/rol.

2. Preservación

- Congela **CloudTrail** (asegura que esté *multi-region* y *data events* para S3).
- Exporta últimos **90 días** a un bucket con **Object Lock**.
- **Snapshots** de EBS de instancias sospechosas.

3. Recolección

- Lista **acciones** del usuario/rol (Create/Delete/List/Get).
- Revisa **VPC Flow Logs** de las subredes afectadas.
- S3: consulta **accesos por objeto** y cambios de políticas/bucket ACL.

4. Análisis

- Línea de tiempo: *primera autenticación anómala* → *enumeración* → *exfiltración*.
- Correlaciona IPs con **GuardDuty** y *threat intel*.

5. Erradicación

- Menor privilegio (principio de **least privilege**), rotación de secretos, revisión de **trust policies**.

6. Informe

- Evidencias: *exports de CloudTrail/Flow Logs con hashes*, snapshots referenciados, diagrama de ataque.

Error común: deshabilitar CloudTrail por “ruido” antes de exportarlo.

Playbook B – Exfiltración de datos desde S3

Indicadores: aumentos de **GetObject**, **ListBucket**, accesos desde IPs inusuales, bucket *público por error*.

1. Bloquea **Public Access** y verifica **Bucket Policy**.
 2. Exporta logs **S3 Access** y **CloudTrail data events** del bucket.
 3. Identifica **prefijos/objetos** accedidos y tamaño de transferencia.
 4. Si hay **versioning**, revisa **objetos previos** y habilita **retention** temporal.
 5. Documenta remitentes (UserAgent, ARN, IP) y correlación con IAM.
-

Playbook C – O365 (M365) eDiscovery por filtración de correos

1. **Habilita/valida Unified Audit Log** y aplica **Legal Hold** en los buzones.
 2. Ejecuta búsquedas por **palabras clave, remitentes, adjuntos** (Purview eDiscovery).
 3. Exporta resultados con **integridad verificable**; preserva *Search Reports*.
 4. Revisa **Sign-in logs** de Entra ID, reglas de **Inbox** maliciosas (auto-forward).
 5. Cambia contraseñas, activa **MFA**, bloquea *legacy auth*.
-

Playbook D – Google Workspace (Drive) exfiltración

1. En **Admin/Audit**: lista **compartidos externos** y **descargas masivas**.
 2. En **Google Vault**: aplica **hold** a usuarios y **export** de Drive.
 3. Correlaciona por archivo: **propietario** → **historial de uso** → **rutas de compartido**.
 4. Suspende cuentas o revoca **OAuth tokens** de apps de tercero.
-

Playbook E – Azure: cuenta comprometida y movimiento lateral

1. **Activity Log** y **Sign-in/Audit** (Entra ID) para origen del compromiso.
 2. **NSG Flow Logs** y **Firewall logs** para movimientos en subredes.
 3. **Snapshots** de discos de VMs impactadas; *just-in-time access* temporal para DFIR.
 4. **Defender for Cloud** hallazgos → revisa *recommendations* vinculadas.
-

Playbook F – PaaS/Serverless (funciones, contenedores)

1. Exporta **logs de ejecución** (CloudWatch/Azure Monitor/Cloud Logging).
 2. Obtén **imagen o digest** del contenedor (si aplica) y mételo en un **registro forense**.
 3. Conserva **variables de entorno/secretos** (ojo con exposición).
 4. Revisa **eventos IAM** que pudieron disparar despliegues maliciosos.
-

16.8 Herramientas y utilidades (campo)

- **CLIs oficiales:** **aws**, **az**, **gcloud**, **gsutil**, **PowerShell** para M365.
- **Auditoría/seguridad de configuración:** **Prowler** (AWS), **ScoutSuite** (multi-cloud), **DefectDojo** (tracking).
- **DFIR & flujo:** **Timesketch** (líneas de tiempo), **dfTimewolf** (orquestación GCP), pipelines a **SIEM** (Sentinel, Splunk, Elastic).

- **K8s/EKS/AKS/GKE:** recolectores de **audit logs**, descripciones de **Pods/Secrets**, *admission controller logs*.
- **SaaS:** conectores nativos (Purview, Google Vault), y APIs de **Dropbox/Slack/GitHub/Okta** para auditoría.

Consejo: prepara **playbooks automatizados** (scripts CLI) con **nombres de caso y salidas normalizadas** (JSON/CSV + hashes).

16.9 Buenas prácticas proactivas (antes del incidente)

- **Encender todos los logs relevantes y elevar retención** (mín. 90 días; ideal 180–365 según riesgo).
- **Centralizar logs** en un **proyecto/tenant de seguridad** con **inmutabilidad**.
- **Etiquetado y cuentas forenses**: proyecto “cold” con permisos de solo-lectura para DFIR.
- **Segmentación de red y privilegios mínimos** en IAM; **MFA obligatorio**.
- **Gestión de claves (KMS/Key Vault/Cloud KMS)** con **CMK** y rotación.
- **Runbooks y ejercicios de mesa** (tabletop) con legales y compliance.
- **Pruebas de export eDiscovery/Logs** para no improvisar en producción.

16.10 Errores comunes que arruinan una pericia en la nube

- Confiar en que “el proveedor tiene todo”: **logs no habilitados** o con **retención de 7 días**.
- **Apagar** instancias críticas sin **exportar logs** ni **snapshot**.
- Hacer cambios de seguridad **sin preservar** el estado (pierdes la “foto del crimen”).
- Exportar evidencia sin **hashes ni manifiestos** (pierde fuerza probatoria).
- No coordinar con **legales**: problemas de **jurisdicción y transferencia internacional**.
- No aislar **tokens OAuth y apps de terceros** (persistencia del atacante).

16.11 Ejemplos de comandos (referenciales)

Ajusta a tu entorno y privilegios. No ejecutes en caliente sin pasar por legales/IR.

AWS – Buscar eventos de un usuario en CloudTrail

```
aws cloudtrail lookup-events --lookup-attributes
AttributeKey=Username,AttributeValue=usuario-comprometido \
--max-results 50 --output json > trail_usuario.json
```

AWS – Crear snapshot EBS

```
aws ec2 create-snapshot --volume-id vol-xxxxxxxx --description "DFIR snapshot
caso-123"
```

Azure – Activity Log últimos 24h (Kusto via az)

```
az monitor activity-log list --status Succeeded --offset 24h > activity_24h.json
```

GCP – Leer auditoría de un proyecto (Cloud Logging)

```
gcloud logging read 'logName:"cloudaudit.googleapis.com"' --limit=1000 --format=json > gcp_audit.json
```

M365 – PowerShell (ejemplo de búsqueda de auditoría unificada)

```
Search-UnifiedAuditLog -StartDate "2025-08-07" -EndDate "2025-08-09" -Operations FileAccessed,FileDownloaded -UserIds usuario@empresa.com | Export-Csv UAL_export.csv -NoTypeInformation
```

16.12 Caso práctico integrado (resumen narrado)

Contexto: fuga de documentos desde un repositorio S3 y envío por correo M365.

1. **Detección:** SOC ve picos de **GetObject** desde IP residencial.
2. **AWS:** export de **CloudTrail data events** y **Access Logs** del bucket; bloqueo *Public Access*; revisión de **IAM** y **MFA**.
3. **M365: Legal Hold** al buzón del sospechoso, búsqueda en **eDiscovery** por términos y adjuntos; auditoría de **auto-forward**.
4. **Correlación:** línea de tiempo con **IPs comunes**, hora de descarga S3 = hora de adjunto en Exchange.
5. **Preservación:** snapshots, exports firmados, hashes y cadena de custodia.
6. **Cierre:** revocación de tokens OAuth, rotación de claves, endurecimiento de políticas S3 y DLP.

Resumen del capítulo La forense en la nube es **telemetría + preservación + API** bajo reglas de **responsabilidad compartida y jurisdicción**. Sin logs ni retenciones, no hay caso. Con playbooks, cuentas forenses, inmutabilidad y eDiscovery, puedes **reconstruir hechos** con solidez probatoria.

Capítulo 17 – Forense en IoT (Internet de las Cosas)

17.1 Introducción

El **Internet de las Cosas (IoT)** ha multiplicado el número de dispositivos conectados que generan, procesan y transmiten datos:

- Cámaras de seguridad y timbres inteligentes.
- Dispositivos de salud (wearables, monitores de ritmo cardíaco).
- Asistentes virtuales (Alexa, Google Home, Siri).

- Sensores industriales.
- Vehículos conectados y sistemas telemáticos.

En informática forense, el IoT representa un **doble desafío**:

1. **Heterogeneidad:** cada fabricante utiliza hardware, sistemas operativos y protocolos distintos.
 2. **Volatilidad:** muchos dispositivos no guardan datos por largo plazo y sobrescriben la información rápidamente.
-

17.2 Tipos de dispositivos y posibles evidencias

Tipo de dispositivo	Ejemplos	Evidencia potencial
Domésticos inteligentes	Cámaras IP, timbres, termostatos	Grabaciones de vídeo, registros de acceso, direcciones IP
Wearables	Smartwatch, pulseras fitness	Ritmo cardíaco, GPS, pasos, horas de sueño
Asistentes virtuales	Alexa, Google Home	Historial de comandos, registros de voz
Sensores industriales	PLCs, sensores de temperatura	Logs de funcionamiento, datos de producción
Vehículos conectados	Tesla, BMW ConnectedDrive	Historial GPS, datos de telemetría, velocidad, estado del vehículo

17.3 Principales retos forenses en IoT

1. **Acceso físico limitado:** muchos dispositivos están sellados o usan hardware propietario.
 2. **Falta de estándares:** cada fabricante define su propia estructura de datos.
 3. **Almacenamiento efímero:** los datos se sobrescriben rápido (buffer circular).
 4. **Dependencia de la nube:** gran parte de la información reside en servidores externos del fabricante.
 5. **Cifrado extremo:** datos protegidos en tránsito y reposo con claves no disponibles al investigador sin orden judicial.
-

17.4 Procedimientos de adquisición forense en IoT

17.4.1 Adquisición directa del dispositivo

- Desmontaje para acceso a memoria flash/NAND.
- Uso de técnicas **chip-off** o **JTAG**.
- Riesgo: daños físicos si no se cuenta con experiencia.

17.4.2 Extracción vía interfaz de administración

- Acceso web o app móvil vinculada.
- Descarga de configuraciones, logs o archivos multimedia.

17.4.3 Adquisición de la nube asociada

- Solicitud a fabricante o proveedor de servicio (requiere orden judicial).
- Uso de APIs públicas o privadas para descarga de datos de usuario.

17.4.4 Captura de tráfico de red

- Monitorización del tráfico entre el IoT y la nube.
- Análisis de protocolos: HTTP/HTTPS, MQTT, CoAP, RTSP, etc.
- Herramientas: **Wireshark**, **tcpdump**, **IoT Inspector**.

17.5 Herramientas para análisis forense en IoT

- **IoT Inspector** – Captura y análisis de tráfico IoT en tiempo real.
- **Autopsy** con módulos personalizados – Análisis de imágenes extraídas de memoria flash.
- **Firmware Mod Kit** – Análisis y modificación de firmware IoT.
- **Binwalk** – Extracción de contenido de firmware.
- **Shodan** – OSINT para localizar dispositivos expuestos en Internet.
- **Bus Pirate** – Herramienta hardware para interactuar con interfaces como UART/SPI/I2C.

17.6 Ejemplo práctico: investigación de cámara IP

Escenario: Una cámara IP doméstica es sospechosa de haber sido utilizada para espiar a una víctima.

Procedimiento:

1. **Preservación:** desconectar la cámara de Internet pero mantener alimentación para evitar sobrescritura.
2. **Captura de tráfico:** conectar cámara a red controlada y capturar datos con Wireshark.
3. **Análisis de almacenamiento interno:** extracción de tarjeta microSD, clonación y análisis con Autopsy.
4. **Revisión de logs:** identificar accesos remotos desde direcciones IP externas.
5. **Informe:** documentar capturas de pantalla, direcciones IP, fechas y horas.

17.7 Ejemplo práctico: forense en smartwatch

Escenario: Se investiga un caso de violencia de género en el que un smartwatch podría contener evidencia de ubicación y actividad física.

Procedimiento:

1. **Sincronización con app móvil:** adquisición de datos desde el teléfono vinculado.
2. **Extracción de la nube:** acceso a cuenta del usuario con autorización judicial.
3. **Ánalisis de registros GPS:** reconstrucción de rutas y horarios.
4. **Correlación con eventos:** comparar datos de actividad con momento del incidente.

17.8 Buenas prácticas en forense IoT

- Documentar modelo, número de serie, firmware y estado del dispositivo.

- Capturar evidencia **antes** de actualizar firmware (puede borrar datos).
 - Usar **bolsas Faraday** para aislar dispositivos inalámbricos.
 - Mantener copias forenses de firmware y configuraciones.
 - Preservar logs de nube lo antes posible.
-

17.9 Errores comunes

- Manipular el dispositivo sin fotografiar su estado inicial.
 - Olvidar que gran parte de la evidencia está en la nube y no en el dispositivo.
 - No capturar tráfico de red cuando el dispositivo aún está activo.
 - Extraer memoria sin las herramientas adecuadas y dañarla.
-

17.10 Caso real documentado

En 2019, un caso judicial en EE.UU. usó como prueba los registros de un **Amazon Echo** (Alexa) que contenían comandos de voz realizados durante la noche de un asesinato. La evidencia fue obtenida mediante orden judicial directa al proveedor, y sirvió para establecer la línea temporal de los hechos.

Resumen del capítulo: El forense en IoT exige combinar técnicas tradicionales (adquisición física, análisis de memoria) con métodos modernos (captura de tráfico, extracción de nube, OSINT). La rapidez de actuación es vital debido a la volatilidad de los datos, y la coordinación legal con fabricantes y proveedores es indispensable.

Capítulo 18 – Inteligencia Artificial aplicada a la Informática Forense

18.1 Introducción

La **Inteligencia Artificial (IA)** y el **Machine Learning (ML)** han revolucionado la informática forense, permitiendo procesar grandes volúmenes de datos, detectar patrones complejos y automatizar tareas repetitivas.

En la actualidad, la IA no sustituye al perito, pero **amplifica sus capacidades**:

- Filtra y prioriza evidencia relevante.
 - Detecta anomalías en grandes conjuntos de logs.
 - Clasifica automáticamente imágenes, vídeos y documentos.
 - Predice posibles vectores de ataque o reincidencia de amenazas.
-

18.2 Áreas clave donde la IA ayuda en forense digital

1. Análisis de logs a gran escala

- Uso de algoritmos para detectar patrones atípicos en millones de registros.
- Ejemplo: un modelo supervisado detecta intentos de login fuera de horario y desde IPs inusuales.

2. Reconocimiento de imágenes y videos

- Clasificación automática de material (p. ej., identificación de armas en videos).
- Detección de manipulación digital (*deepfake detection*).

3. Procesamiento de lenguaje natural (NLP)

- Análisis semántico de correos electrónicos para detectar phishing o amenazas.
- Extracción automática de nombres, lugares y fechas de documentos.

4. Análisis de tráfico de red

- Modelos no supervisados para detectar patrones anómalos en tráfico normal.
- Ejemplo: descubrir un botnet oculto en tráfico HTTPS legítimo.

5. Predicción y perfilado

- Modelos que sugieren los activos más probables de ser atacados.
- Priorización de evidencias en función de la probabilidad de contener información clave.

18.3 Tipos de algoritmos aplicados en forense

Tipo de algoritmo	Aplicación en forense
Supervisado (SVM, Random Forest)	Clasificación de correos como legítimos o phishing
No supervisado (K-Means, DBSCAN)	Agrupación de patrones anómalos en tráfico de red
Redes neuronales (CNN, RNN)	Reconocimiento de imágenes y análisis secuencial de logs
Deep Learning	Detección de manipulación en vídeo o audio
Reglas híbridas (ML + SIEM)	Enriquecimiento automático de alertas de seguridad

18.4 Ejemplo práctico 1 – Clasificación de correos sospechosos

Escenario: Se reciben 100.000 correos en una organización; se sospecha que un 1% son phishing.

Procedimiento con IA:

1. Preprocesar: extraer encabezados, cuerpo, enlaces y adjuntos.
2. Entrenar un modelo supervisado con ejemplos previos de phishing y no phishing.
3. Ejecutar clasificación y generar lista priorizada de sospechosos.
4. Revisar manualmente solo los casos de mayor probabilidad.

Beneficio: Reducción del 99% del trabajo manual.

18.5 Ejemplo práctico 2 – Análisis de imágenes incautadas

Escenario: En una investigación criminal se incautan 2 TB de imágenes.

Procedimiento con IA:

1. Usar una CNN (red neuronal convolucional) para clasificar imágenes por categorías: documentos, personas, armas, etc.
2. Filtrar automáticamente las imágenes irrelevantes (paisajes, fondos).
3. Aplicar herramientas de detección de manipulación (*forensic image analysis*).

Beneficio: Acelera semanas de trabajo en horas, sin perder trazabilidad.

18.6 Ejemplo práctico 3 – Detección de anomalías en red

Escenario: Un banco desea detectar patrones de exfiltración de datos.

Procedimiento con IA:

1. Recopilar un mes de tráfico normal.
 2. Entrenar un modelo no supervisado para aprender el comportamiento típico.
 3. Activar el modelo en producción para alertar sobre desviaciones.
 4. Integrar con SIEM para generar alertas automáticas.
-

18.7 Herramientas y frameworks para IA forense

- **TensorFlow / PyTorch** – Entrenamiento de modelos personalizados.
 - **Scikit-learn** – Algoritmos clásicos de ML.
 - **OpenCV** – Procesamiento de imágenes y vídeo.
 - **ELK Stack + ML jobs (Elastic)** – Detección de anomalías en logs.
 - **MalwareBazaar API + modelos** – Clasificación de malware.
 - **Microsoft Sentinel / Splunk ML Toolkit** – Integración de ML en SIEM.
-

18.8 Limitaciones y riesgos

- **Falsos positivos/negativos:** un modelo mal entrenado puede generar ruido o dejar pasar amenazas.
 - **Sesgo de datos:** si el dataset de entrenamiento no es representativo, el modelo fallará.
 - **Opacidad:** algunos algoritmos de deep learning son cajas negras difíciles de explicar en un juicio.
 - **Requerimientos legales:** la evidencia generada debe ser reproducible y verificable.
-

18.9 Buenas prácticas para IA en forense

- Mantener datasets actualizados y etiquetados.
 - Documentar el proceso de entrenamiento y parámetros del modelo.
 - Validar resultados con revisión humana.
 - Integrar IA como apoyo, no sustituto, de la pericia.
 - Usar pipelines reproducibles y auditables.
-

Resumen del capítulo: La IA en informática forense es un multiplicador de capacidades, especialmente útil cuando el volumen de datos es abrumador. Su éxito depende de la calidad del dataset, la correcta integración con herramientas forenses y el control de sesgos y errores.

Capítulo 19 – Cadena de Custodia Digital

19.1 Introducción

La **cadena de custodia** es el conjunto de procedimientos que garantizan la **integridad, autenticidad y trazabilidad** de la evidencia digital desde su obtención hasta su presentación en un juicio.

En informática forense, la cadena de custodia no es un mero trámite:

- **Sin ella, la evidencia puede ser impugnada.**
- Asegura que **no se alteró** el contenido.
- Documenta **quién** manipuló la evidencia, **cuándo, dónde y para qué**.

Una evidencia técnicamente perfecta pero sin cadena de custodia es, legalmente, un **papel mojado**.

19.2 Principios básicos

1. Identificación clara de la evidencia

- Número único de caso.
- Descripción detallada (tipo, marca, modelo, número de serie, estado físico).

2. Registro continuo

- Cada vez que la evidencia cambia de manos, se debe registrar fecha, hora, persona y propósito.

3. Integridad comprobable

- Uso de funciones hash (MD5, SHA-256) para asegurar que los datos no fueron modificados.

4. Almacenamiento seguro

- En contenedores sellados, bolsas antiestáticas o bóvedas digitales con acceso controlado.

19.3 Etapas de la cadena de custodia digital

Etapa	Descripción	Ejemplo práctico
Recolección	Obtención de la evidencia desde la fuente.	Clonar un disco usando <code>dd</code> y calcular hash SHA-256.
Preservación	Proteger la evidencia contra alteraciones.	Guardar imagen forense en repositorio WORM.

Etapa	Descripción	Ejemplo práctico
Transferencia	Mover evidencia entre personas/áreas.	Entregar disco a laboratorio registrando firmas.
Análisis	Examen técnico de la evidencia.	Usar Autopsy en copia de trabajo.
Presentación	Uso de la evidencia en juicio.	Mostrar capturas, logs y hashes al tribunal.

19.4 Ejemplo práctico de registro de cadena de custodia

Caso: incautación de un disco rígido en allanamiento.

Nº de caso: 2025-DF-014
 Fecha/hora de incautación: 08/08/2025 – 14:32
 Lugar: Calle Falsa 123, Buenos Aires, Argentina
 Descripción: Disco rígido Seagate Barracuda 2TB, SN: Z4E12345
 Recibido por: Alejandro G Vera
 Función: Perito informático
 Hash inicial: SHA-256 – 5a1f...c92d
 Condiciones: Etiquetado, sellado en bolsa antiestática, precinto N° 775

Cada transferencia posterior debe incluir:

- Fecha/hora.
- Nombre y firma de quien entrega y quien recibe.
- Motivo de la transferencia.
- Estado del precinto o integridad digital.

19.5 Uso de hash en la cadena de custodia

- **Antes y después** de cada copia o transferencia, calcular el hash de la evidencia.
- Algoritmos recomendados: SHA-256 o SHA-512.
- Guardar hash en un **documento firmado** y, si es posible, en un repositorio inmutable.

Ejemplo:

```
sha256sum disco.img > disco_hash.txt
```

Resultado:

```
5a1f3b8d66df23d38e97cbbd6b2e6a7f9b47245d7d125ba0c91e12df0e3ec92d disco.img
```

19.6 Escenarios prácticos

19.6.1 Evidencia en dispositivos físicos

- Precintar con sellos numerados.
- Tomar fotografías antes y después.
- Documentar estado físico (golpes, daños).

19.6.2 Evidencia en la nube

- Registrar **método de acceso** (API, consola web).
- Guardar logs de exportación y metadatos (fecha, usuario, región).
- Aplicar hash al archivo descargado.

19.6.3 Evidencia volátil (memoria RAM, tráfico de red)

- Capturar lo antes posible.
- Documentar herramientas usadas y versión.
- Guardar hash de la captura y metadatos del sistema.

19.7 Herramientas de apoyo

- **FTK Imager** – Adquisición y verificación de hash.
- **Autopsy / Sleuth Kit** – Análisis forense manteniendo integridad.
- **dcfldd** – Clonado con verificación simultánea.
- **Magnet AXIOM** – Documentación automática de cadena de custodia.
- **HashCalc / sha256sum** – Cálculo de hash.
- **Blockchain forensics log** – Uso de blockchain para registrar transferencias.

19.8 Errores comunes

- No calcular hash en el momento de la recolección.
- Usar solo MD5 (vulnerable a colisiones).
- Manipular evidencia original sin copia forense.
- Falta de registro en una transferencia interna.
- Almacenar en dispositivos no protegidos contra escritura.

19.9 Caso real documentado

En 2020, un caso de fraude corporativo en Argentina fue desestimado porque la defensa demostró que un disco incautado estuvo 48 horas sin registro de custodia. Aunque el contenido era incriminatorio, **la brecha en la trazabilidad invalidó la prueba**.

Resumen del capítulo: La cadena de custodia digital es la columna vertebral de la validez legal de la evidencia. Sin un registro preciso, con firmas, fechas, hash y control de acceso, el trabajo técnico pierde fuerza en un proceso judicial.

Capítulo 20 – Informática Forense en Dispositivos Móviles

20.1 Introducción

Los dispositivos móviles (teléfonos, tablets y relojes inteligentes) concentran **enorme cantidad de información personal y corporativa**, incluyendo:

- **Comunicación:** llamadas, SMS, chats en aplicaciones.
- **Ubicación:** historial GPS, conexiones Wi-Fi, celdas móviles.
- **Multimedia:** fotos, vídeos, grabaciones de audio.
- **Documentos y archivos:** descargados o generados por el usuario.
- **Datos en la nube:** sincronización con servicios externos.

La informática forense móvil implica **preservar, extraer, analizar y documentar** estos datos **sin alterar el contenido original** y cumpliendo la cadena de custodia.

20.2 Tipos de adquisición en dispositivos móviles

Tipo	Características	Ejemplo de herramientas
Lógica	Extrae datos visibles al usuario y backups. No accede a datos borrados.	ADB, iTunes, Mobiledit
Física	Copia bit a bit de toda la memoria del dispositivo. Permite recuperación de datos borrados.	Cellebrite UFED, Oxygen Forensic, JTAG
File System	Extrae estructura completa del sistema de archivos, incluyendo metadatos.	Magnet AXIOM, Elcomsoft
Nube	Obtiene datos de servicios sincronizados (Google, iCloud).	Oxygen Forensic Cloud Extractor

20.3 Procedimiento general de análisis forense móvil

1. Preservación de evidencia

- Colocar en **modo avión** para evitar modificaciones remotas.
- Usar **bolsas Faraday** si hay riesgo de conexión no autorizada.
- Documentar número de serie, IMEI, modelo, estado físico.

2. Adquisición

- Elegir tipo según objetivo de la investigación y capacidades técnicas.
- Conservar siempre una **copia inalterada** de la adquisición.

3. Análisis

- Búsqueda de chats, llamadas, fotos, ubicaciones, redes sociales.
- Recuperar datos borrados si la adquisición es física.

4. Documentación

- Guardar capturas de pantalla, informes exportados, hashes.
- Mantener **registro de cadena de custodia**.

20.4 Ejemplo práctico 1 – Análisis de Android desbloqueado

Escenario: Teléfono Android Samsung Galaxy con sospecha de fraude corporativo.

Procedimiento:

1. Activar modo avión y conectar mediante **ADB**:

```
adb devices  
adb pull /sdcard/ /evidencia/
```

2. Exportar base de datos de WhatsApp:

```
/sdcard/WhatsApp/Databases/msgstore.db
```

3. Analizar con **WhatsApp Viewer** o Magnet AXIOM.

4. Revisar fotos con **ExifTool** para obtener geolocalización.

Resultado: Recuperación de conversaciones y ubicación de fotos tomadas el día del incidente.

20.5 Ejemplo práctico 2 – Extracción física de iPhone bloqueado

Escenario: iPhone 12 involucrado en caso de acoso digital.

Procedimiento:

1. Conectar a **Cellebrite UFED**.
2. Usar **checkm8 exploit** si el modelo y versión lo permiten.
3. Obtener imagen física y generar hash SHA-256.
4. Analizar con **Elcomsoft iOS Forensic Toolkit** para recuperar mensajes borrados de iMessage.

Resultado: Mensajes eliminados que confirmaban envío de contenido no autorizado.

20.6 Ejemplo práctico 3 – Recuperación de datos desde la nube

Escenario: La víctima perdió el teléfono, pero tenía activada copia en Google Drive.

Procedimiento:

1. Acceso autorizado a la cuenta de Google.
2. Usar **Oxygen Forensic Cloud Extractor** para descargar backups.
3. Analizar mensajes, contactos y fotos sincronizadas.

Resultado: Recuperación completa de conversaciones de Telegram y ubicación GPS.

20.7 Evidencias comunes en móviles

- **Registros de llamadas y SMS.**
- **Chats** (WhatsApp, Telegram, Signal, Messenger).
- **Archivos multimedia con metadatos** (EXIF: ubicación, fecha, modelo de cámara).
- **Datos de aplicaciones** (historial de navegación, credenciales).
- **Historial de ubicaciones.**
- **Datos en caché y temporales.**

20.8 Herramientas forenses móviles destacadas

- **Cellebrite UFED** – Adquisición lógica, física y nube.
- **Oxygen Forensic Detective** – Amplio soporte para apps y datos en la nube.
- **Magnet AXIOM** – Integración de móvil, PC y nube en un solo análisis.
- **Elcomsoft iOS Forensic Toolkit** – Especializado en dispositivos Apple.
- **ADB + scripts personalizados** – Alternativa económica para Android.

20.9 Buenas prácticas

- Trabajar siempre sobre una copia de la adquisición, nunca sobre el dispositivo original.
- Mantener registros de cada acción con fecha, hora y herramienta utilizada.
- Usar versiones actualizadas de software para soportar nuevos modelos y sistemas operativos.
- Documentar limitaciones (p. ej., cifrado no extraíble sin credenciales).

20.10 Errores comunes

- Conectar el móvil a un PC sin protección, provocando sincronizaciones automáticas y cambios en datos.
- Intentar adivinanzas de contraseña y provocar borrado remoto.
- No aislar el dispositivo y permitir que reciba comandos que alteren la evidencia.

20.11 Caso real documentado

En 2021, en un caso de corrupción política en Brasil, se incautó un iPhone con mensajes en Telegram. Gracias a un backup en iCloud y a herramientas de Elcomsoft, se recuperaron conversaciones borradas que demostraban coordinación ilícita. La evidencia fue aceptada en juicio por cumplirse estrictamente la cadena de custodia.

Resumen del capítulo: La informática forense en dispositivos móviles requiere un equilibrio entre **técnica, rapidez y legalidad**. Cada minuto cuenta, especialmente si el dispositivo está conectado o los datos son volátiles. La elección del método de adquisición depende del objetivo, del tipo de dispositivo y de las limitaciones legales y técnicas.

Capítulo 21 – Análisis Forense de Redes

21.1 Introducción

El **análisis forense de redes** se centra en la recolección, preservación y examen de datos que viajan por una red de computadoras, con el objetivo de:

- **Identificar intrusiones y ataques.**
- **Rastrear exfiltración de datos.**
- **Analizar incidentes en tiempo real o pasado.**
- **Correlacionar eventos con evidencia digital.**

Es una disciplina crítica en **respuesta a incidentes** porque:

- Muchas veces el atacante no deja rastros en discos locales.
 - Los logs y capturas de tráfico pueden ser la única evidencia.
-

21.2 Tipos de evidencia en redes

Fuente	Tipo de evidencia	Ejemplo
PCAP (Packet Capture)	Tráfico crudo de la red	Captura con Wireshark o tcpdump
Logs de firewall	Conexiones permitidas/bloqueadas	Cisco ASA, pfSense
Logs de servidores	Actividad de servicios	Apache, SSH, FTP
Registros de IDS/IPS	Alertas de ataques	Snort, Suricata
Flujos de red	Resumen de tráfico	NetFlow, sFlow

21.3 Procedimiento general de análisis forense de redes

1. Preservación de la evidencia

- Usar herramientas que capturen sin alterar tráfico.
- Guardar PCAP originales con hash SHA-256.

2. Filtrado de tráfico relevante

- Reducir volumen de datos aplicando filtros por IP, puerto, protocolo.

3. Análisis detallado

- Identificar patrones anómalos o maliciosos.
- Reconstruir sesiones y extraer archivos transferidos.

4. Correlación

- Cruzar datos de PCAP, logs y otros dispositivos de seguridad.

5. Documentación

- Guardar capturas de pantalla, scripts usados y resultados con fecha y hora.
-

21.4 Herramientas clave

- **Wireshark** – Análisis visual y detallado de PCAP.
 - **tcpdump** – Captura y filtrado rápido en terminal.
 - **Zeek (Bro)** – Análisis de tráfico y generación de logs de alto nivel.
 - **NetworkMiner** – Reconstrucción de archivos y sesiones.
 - **Snort / Suricata** – Detección de intrusiones.
 - **Arkime (Moloch)** – Indexación y búsqueda en grandes volúmenes de PCAP.
-

21.5 Ejemplo práctico 1 – Detección de intrusión SSH

Escenario: Servidor Linux muestra intentos fallidos de login.

Procedimiento:

1. Capturar tráfico durante el incidente:

```
tcpdump -i eth0 port 22 -w ssh_intrusion.pcap
```

2. Abrir en Wireshark y filtrar por intentos de conexión:

```
tcp.flags.syn==1 && tcp.flags.ack==0
```

3. Identificar IPs de origen y geolocalizar con herramientas OSINT.

4. Correlacionar con `/var/log/auth.log` para verificar intentos.

Resultado: Confirmación de ataque de fuerza bruta desde direcciones en Asia.

21.6 Ejemplo práctico 2 – Análisis de exfiltración de datos

Escenario: Sospecha de que un empleado filtró información a servidor externo.

Procedimiento:

1. Revisar registros NetFlow para identificar grandes volúmenes de salida.

2. Filtrar PCAP por IP de destino:

```
ip.dst == 203.0.113.45
```

3. Usar NetworkMiner para extraer archivos enviados.

4. Calcular hash y verificar si coinciden con documentos internos.

Resultado: Detección de transferencia de base de datos comprimida a servidor no autorizado.

21.7 Ejemplo práctico 3 – Detección de malware en tráfico HTTP

Escenario: Un endpoint ejecuta conexiones HTTP sospechosas.

Procedimiento:

1. Capturar tráfico completo durante 24 horas.

2. Filtrar en Wireshark:

```
http.request && !(ip.dst == 192.168.0.0/16)
```

3. Identificar patrones de *beaconing* (peticiones periódicas idénticas).

4. Extraer payloads y analizar con antivirus o sandbox.

Resultado: Identificación de troyano con comunicación a C2 (Command and Control) en Europa del Este.

21.8 Buenas prácticas

- Capturar en puntos clave de la red (SPAN port, TAP).
 - Usar relojes sincronizados (NTP) para correlación de eventos.
 - Comprimir y almacenar PCAP en formato seguro.
 - Documentar filtros aplicados para permitir replicar el análisis.
-

21.9 Errores comunes

- Filtrar demasiado pronto y perder tráfico relevante.
 - No preservar la captura original antes de procesar.
 - Usar relojes desincronizados en dispositivos, dificultando la correlación.
 - Analizar PCAP en entorno inseguro y activar malware capturado.
-

21.10 Caso real documentado

En 2017, un ataque a una empresa de energía en Ucrania fue reconstruido gracias a capturas de Zeek que mostraban comunicación con servidores C2 y cargas maliciosas vía HTTP. La evidencia fue clave para atribuir el ataque a un grupo de APT (Advanced Persistent Threat) específico.

-
- Resumen del capítulo:** El análisis forense de redes permite reconstruir incidentes invisibles para otros sistemas. La captura y el filtrado correcto, junto con herramientas adecuadas, son esenciales para obtener evidencia sólida y legalmente aceptable.
-

Capítulo 22 – Técnicas de Esteganografía y Contraesteganografía

22.1 Introducción

La **esteganografía** es el arte y ciencia de **ocultar información** dentro de otro archivo o medio de forma que pase inadvertida. En informática forense, la esteganografía es relevante porque:

- Es usada para **comunicación encubierta** entre criminales o espías.
- Permite **exfiltración de datos** camuflados en archivos aparentemente inofensivos.
- A veces se usa junto con cifrado, lo que dificulta aún más la detección.

La **contraesteganografía** consiste en aplicar técnicas para **detectar, analizar y extraer** esa información oculta.

22.2 Tipos de esteganografía digital

Medio	Método de ocultación	Ejemplo
Imágenes	Modificación de bits menos significativos (LSB) de píxeles.	Ocultar texto en un PNG.
Audio	Alteración de bits de muestras de sonido o enmascaramiento de ruido.	Mensaje en un MP3.
Vídeo	Inserción de datos en fotogramas o pistas de audio.	Texto oculto en frames de un MP4.
Texto	Uso de caracteres invisibles, espaciado o cambios tipográficos.	Espacios dobles que codifican bits.
Protocolos de red	Manipulación de campos no usados en cabeceras.	Datos en campos padding de TCP/IP.

22.3 Herramientas comunes para esteganografía

- **OpenStego** – Ocultación y extracción de datos en imágenes.

- **Steghide** – Oculta datos en imágenes y audio, con cifrado opcional.
 - **SilentEye** – GUI para esteganografía en JPG y WAV.
 - **ExifTool** – Manipulación de metadatos, a veces usados para ocultar texto.
 - **Snow** – Ocultación de datos en espacios y tabulaciones en texto.
-

22.4 Técnicas de contraesteganografía

1. Análisis visual

- Comparar imagen sospechosa con original (si existe).
- Buscar patrones anómalos o ruido excesivo.

2. Análisis estadístico

- Detectar alteraciones en la distribución de bits o colores.
- Herramientas como **StegExpose** pueden automatizar este análisis.

3. Extracción de metadatos

- Buscar campos de metadatos inusuales o con datos binarios no estándar.
- Uso de **ExifTool**.

4. Descompresión y análisis de archivos

- Algunos métodos añaden datos al final de un archivo.
- Uso de **binwalk** o editores hexadecimales.

5. Análisis de tráfico de red

- Buscar cabeceras alteradas o campos poco usados con contenido extraño.
-

22.5 Ejemplo práctico 1 – Ocultación en imagen PNG

Escenario: Un sospechoso podría haber enviado información confidencial oculta en una foto.

Procedimiento de ocultación (del atacante):

```
steghide embed -cf foto.png -ef secreto.txt -p clave123
```

Procedimiento de detección (forense):

1. Usar **StegExpose**:

```
stegexpose foto.png
```

2. Si hay indicios, intentar extracción:

```
steghide extract -sf foto.png -p clave123
```

3. Verificar hash del archivo extraído para confirmar coincidencia con datos robados.

22.6 Ejemplo práctico 2 – Detección de datos en audio

Escenario: Un archivo MP3 podría contener instrucciones ocultas.

Procedimiento:

1. Abrir en **Audacity** y observar espectrograma buscando patrones extraños.
2. Usar **Steghide** para intentar extracción:

```
steghide extract -sf mensaje.mp3
```

3. Si no hay clave, usar ataque de diccionario para intentar extraer datos.

22.7 Ejemplo práctico 3 – Contraesteganografía en documentos PDF

Escenario: Un PDF contiene datos ocultos mediante campos de formulario invisibles.

Procedimiento:

1. Usar **pdf-parser** para examinar objetos:

```
pdf-parser.py archivo.pdf
```

2. Buscar cadenas codificadas en Base64 o binario.
3. Extraer y decodificar para analizar contenido.

22.8 Buenas prácticas en investigación de esteganografía

- Trabajar sobre copias forenses del archivo original.
- Documentar herramienta, versión y comandos utilizados.
- Usar varias técnicas, ya que no existe un único método de detección infalible.
- Considerar la posibilidad de esteganografía en combinación con cifrado.

22.9 Errores comunes

- Asumir que la ausencia de detección significa ausencia de esteganografía.
- No analizar metadatos que podrían contener datos ocultos.

- Modificar el archivo antes de analizarlo, destruyendo evidencia.
-

22.10 Caso real documentado

En 2010, un grupo de espías rusos en EE.UU. utilizó imágenes publicadas en sitios web para ocultar mensajes cifrados mediante esteganografía LSB. El FBI detectó patrones anómalos y, tras extracción y descifrado, recuperó instrucciones operativas.

Resumen del capítulo: La esteganografía es una técnica poderosa para ocultar información, pero también deja huellas que un investigador forense preparado puede detectar. La clave está en combinar **análisis visual, estadístico y de metadatos**, con herramientas adecuadas y documentación rigurosa.

Capítulo 23 – Criptografía en la Informática Forense

23.1 Introducción

La **criptografía** es el conjunto de técnicas que permiten **proteger la información mediante cifrado** para que solo pueda ser leída por quien posea la clave correcta. En informática forense, la criptografía es **una barrera y una herramienta**:

- **Barrera:** dificulta el acceso a datos cifrados sin autorización.
 - **Herramienta:** asegura la integridad y autenticidad de la evidencia recolectada.
-

23.2 Conceptos clave

Concepto	Descripción	Ejemplo
Texto plano (plaintext)	Información original sin cifrar	"Contraseña123"
Texto cifrado (ciphertext)	Información transformada mediante algoritmo	VXBkbFJsZ2RrQw==
Clave	Valor secreto para cifrar/descifrar	K3f8!5Q
Algoritmo	Método matemático para cifrar	AES, RSA, ChaCha20
Hash	Resumen criptográfico no reversible	SHA-256

23.3 Tipos de criptografía

1. Simétrica

- Misma clave para cifrar y descifrar.
- Ejemplo: AES, DES, 3DES.
- Ventaja: rápida. Desventaja: requiere canal seguro para la clave.

2. Asimétrica

- Par de claves: pública y privada.
- Ejemplo: RSA, ECC.
- Usada en firmas digitales y comunicación segura.

3. Hash criptográfico

- Función unidireccional que genera huella digital del dato.
- Ejemplo: SHA-256, SHA-512, BLAKE3.
- Usado en verificación de integridad.

23.4 Criptografía como barrera en investigaciones

En muchos casos forenses, los datos están protegidos mediante:

- **Discos cifrados** (BitLocker, VeraCrypt, LUKS).
- **Archivos cifrados** (ZIP con AES, PDFs protegidos).
- **Mensajería cifrada extremo a extremo** (Signal, WhatsApp, Telegram).
- **Bases de datos cifradas** (SQLCipher).

Esto plantea retos como:

- Necesidad de credenciales o claves.
- Uso de fuerza bruta o diccionario (costoso en tiempo).
- Análisis de memoria RAM para obtener claves cargadas.
- Ataques de ingeniería social para obtener contraseñas.

23.5 Criptografía como herramienta en forense

- **Firmas digitales** para autenticar documentos periciales.
- **Sellado de tiempo** (timestamping) para demostrar cuándo existía un archivo.
- **Hashing** para preservar integridad de evidencia.
- **Cifrado de evidencia** en tránsito o almacenamiento.

23.6 Ejemplo práctico 1 – Verificación de integridad con hash

Escenario: Se recibe un archivo como evidencia y se debe verificar que no fue alterado.

Procedimiento:

```
sha256sum evidencia.img
```

Resultado:

```
f4a7e2830bf8f3b0e2d4d94e9912f84d84e6298a70cdb6e2a5b6b86bb8b1a3f evidencia.img
```

Comparar con hash original registrado al momento de la recolección.

23.7 Ejemplo práctico 2 – Recuperación de clave desde RAM

Escenario: Un disco cifrado con VeraCrypt está montado y activo.

Procedimiento:

1. Capturar memoria RAM con **Belkasoft Live RAM Capturer**.
2. Analizar volcado con **Volatility** buscando cadenas:

```
strings memoria.bin | grep -i veracrypt
```

3. Extraer clave si está presente.

Resultado: Clave recuperada y uso de imagen montada para análisis.

23.8 Ejemplo práctico 3 – Ataque de diccionario sobre ZIP cifrado

Escenario: Se encuentra un archivo ZIP protegido.

Procedimiento:

```
fcrackzip -u -D -p diccionario.txt archivo.zip
```

El diccionario contiene posibles contraseñas recolectadas durante la investigación.

23.9 Herramientas útiles en criptografía forense

- **Hashcat** – Cracking de contraseñas por GPU.
- **John the Ripper** – Ataques de diccionario, fuerza bruta y reglas.
- **GPG** – Firmas digitales y cifrado de archivos.
- **OpenSSL** – Operaciones criptográficas en general.
- **FTK Imager** – Integridad de evidencias y generación de hash.

23.10 Buenas prácticas

- Usar algoritmos seguros y actualizados (evitar MD5 o SHA-1 para integridad).
- Documentar comandos, herramientas y versiones.
- Mantener las claves de evidencia cifrada bajo custodia segura.
- Si se rompe un cifrado, registrar método, tiempo y recursos empleados.

23.11 Errores comunes

- No calcular hash antes y después de transferir evidencia.
 - Confiar en que un archivo sin contraseña no está cifrado (puede tener cifrado interno).
 - Usar contraseñas débiles para evidencia protegida.
-

23.12 Caso real documentado

En 2016, el FBI accedió a un iPhone cifrado implicado en el caso de San Bernardino. La agencia no reveló el método, pero se sospecha que se usaron técnicas combinadas de análisis de firmware y extracción de datos desde memoria volátil para evitar el borrado automático.

Capítulo 24 – Recuperación de Archivos Eliminados en HDD, SSD y Pendrive

24.1 Introducción

Cuando un archivo se elimina, en la mayoría de los sistemas operativos **no se borra físicamente el contenido**, sino que se marca el espacio como libre. Esto significa que, **hasta que se sobrescriba**, es posible recuperarlo con herramientas adecuadas.

En informática forense, la recuperación de archivos eliminados puede aportar evidencia clave, como:

- Documentos borrados para ocultar un delito.
 - Fotografías o videos eliminados antes de un allanamiento.
 - Historiales de chat o correos suprimidos.
-

24.2 Factores que afectan la recuperación

1. Tipo de dispositivo

- **HDD**: mayor probabilidad de recuperación.
- **SSD**: el comando TRIM reduce posibilidades, ya que borra bloques automáticamente.
- **Pendrive/Flash**: comportamiento similar a SSD, aunque no siempre usan TRIM.

2. Sistema de archivos

- **NTFS** (Windows), **exFAT/FAT32** (pendrives, tarjetas SD), **EXT4** (Linux), **APFS/HFS+** (MacOS).
- Algunos mantienen metadatos más tiempo que otros.

3. Tiempo desde el borrado

- Cuanto más rápido se actúe, más alta la probabilidad de éxito.

4. Sobrescritura posterior

- Instalar software de recuperación en el mismo disco puede sobrescribir evidencia.
-

24.3 Procedimiento general de recuperación

1. Preservación de la evidencia

- Nunca trabajar sobre el medio original.
- Hacer imagen forense del dispositivo:

```
dd if=/dev/sdX of=evidencia.img bs=4M conv=noerror,sync  
sha256sum evidencia.img > hash.txt
```

2. Análisis en copia

- Usar herramientas forenses para examinar estructura y sectores.

3. Recuperación de archivos

- Buscar entradas en tabla de archivos o firmas de datos en sectores sin asignar.

4. Documentación

- Registrar herramienta, versión, parámetros, fecha y hora.
-

24.4 Herramientas comunes

Herramienta	Uso principal
Autopsy/The Sleuth Kit	Recuperación y análisis forense de sistemas de archivos.
TestDisk	Restaurar particiones y recuperar archivos borrados.
PhotoRec	Recuperar archivos por firma, incluso sin tabla de archivos.
R-Studio	Herramienta comercial muy completa.
Scalpel	Recuperación basada en cabeceras y pies de archivo.

24.5 Ejemplo práctico 1 – Recuperación en HDD con NTFS

Escenario: Se eliminó una carpeta de fotos en un disco duro mecánico.

Procedimiento:

1. Crear imagen forense (**dd** o FTK Imager).
2. Abrir en **Autopsy**:
 - Seleccionar “Deleted Files” para listar archivos borrados.

- Filtrar por extensión .jpg.
3. Exportar archivos recuperados a carpeta de trabajo.
 4. Verificar integridad visual y metadatos EXIF.

Resultado: Recuperación del 95% de las fotos intactas.

24.6 Ejemplo práctico 2 – Recuperación en SSD con TRIM activo

Escenario: Un SSD Samsung EVO con Windows 10 eliminó un documento sensible.

Procedimiento:

1. Imagen forense completa.
2. Analizar con **PhotoRec** buscando por firma .docx.
3. Hallar solo fragmentos incompletos, ya que TRIM sobrescribió bloques.

Conclusión: En SSD modernos con TRIM activo, la recuperación suele ser parcial o imposible si pasó tiempo.

24.7 Ejemplo práctico 3 – Pendrive con FAT32

Escenario: Pendrive de 16GB con videos eliminados.

Procedimiento:

1. Crear imagen:

```
dd if=/dev/sdb of=pendrive.img bs=4M conv=noerror,sync
```

2. Analizar con **TestDisk**:

- Seleccionar "Advanced" > "Undelete".
- Recuperar archivos .mp4.

3. Verificar con reproductor de vídeo.

Resultado: Recuperación completa de todos los videos.

24.8 Recuperación avanzada – Análisis de sectores

Si la tabla de archivos está corrupta:

- Usar **Scalpel** o **bulk_extractor** para buscar patrones de inicio y fin de archivos.
 - Esto permite recuperar fragmentos aunque el nombre original se pierda.
-

24.9 Buenas prácticas

- **Actuar rápido:** cuanto más tiempo pase, más probable que se sobrescriba.
 - Trabajar siempre en **imagen forense**.
 - Documentar cada paso y guardar logs de herramientas.
 - Usar varias herramientas: lo que una no encuentra, otra puede recuperarlo.
-

24.10 Errores comunes

- Ejecutar herramientas en el disco original y sobrescribir datos.
 - Montar la unidad en modo lectura/escritura sin protección.
 - No calcular hash antes y después de la adquisición.
-

24.11 Caso real documentado

En 2019, en un caso de fraude en Argentina, se recuperaron correos electrónicos borrados de un pendrive con FAT32 usando TestDisk. La defensa argumentó que el borrado era intencional, pero los metadatos mostraron que la eliminación fue posterior a una citación judicial, lo que agravó la causa.

Resumen del capítulo: La recuperación de archivos eliminados es una habilidad esencial en informática forense. En HDD es más viable, en SSD depende del TRIM, y en pendrives la probabilidad es intermedia. La clave está en preservar la evidencia y trabajar sobre copias.

Capítulo 25 – Análisis Forense de Memoria RAM

25.1 Introducción

La **memoria RAM** es el espacio temporal donde un sistema almacena datos que necesita en uso inmediato. En un análisis forense, la RAM puede contener:

- Contraseñas en texto claro.
- Claves de cifrado cargadas.
- Procesos en ejecución.
- Conexiones de red activas.
- Fragmentos de chats o documentos abiertos.
- Malware residente solo en memoria (*fileless malware*).

La dificultad radica en que **la RAM es volátil**: su contenido desaparece al apagar el equipo. Por eso, **la adquisición debe ser inmediata** si el sistema está encendido.

25.2 Usos del análisis de RAM en informática forense

- **Recuperar credenciales y sesiones** activas.
- **Analizar malware en vivo**.
- **Rastrear actividad reciente** no escrita en disco.

- **Extraer archivos y datos temporales.**
 - **Capturar evidencias de cifrados montados** (VeraCrypt, BitLocker).
-

25.3 Procedimiento general

1. Asegurar la escena

- Mantener encendido el equipo (no reiniciar ni apagar).
- Evitar uso innecesario para no sobrescribir datos en RAM.

2. Adquisición

- Usar herramientas especializadas para capturar el volcado.
- Guardar el archivo generado en un medio externo seguro.

3. Cálculo de hash

- Verificar integridad del volcado.

```
sha256sum memoria.bin > memoria_hash.txt
```

4. Análisis

- Procesar el volcado con herramientas forenses de memoria.
- Buscar procesos, conexiones, credenciales, DLLs inyectadas.

5. Documentación

- Registrar fecha, hora, comandos y herramientas utilizadas.
-

25.4 Herramientas de adquisición de RAM

Herramienta	Plataforma	Comentarios
Belkasoft Live RAM Capturer	Windows	Gratuita, captura en vivo sin alterar mucho el sistema.
Dumpli	Windows	Simple, un solo ejecutable, genera volcado completo.
WinPmem	Windows	Integrada en Rekall, formato compatible con Volatility.
LiME	Linux	Módulo de kernel para volcado de RAM.
FTK Imager	Windows/Linux	Puede extraer RAM además de discos.

25.5 Herramientas de análisis

- **Volatility Framework** – Herramienta de línea de comandos muy potente.
- **Rekall** – Similar a Volatility, con mejoras en rendimiento.
- **Belkasoft Evidence Center** – Suite comercial.

- **X-Ways Forensics** – Integración de análisis de RAM y disco.
-

25.6 Ejemplo práctico 1 – Listar procesos activos

Escenario: Se sospecha que un malware se ejecuta solo en memoria.

Procedimiento:

1. Capturar RAM con **Dumpli** → `memoria.bin`.

2. Analizar con Volatility:

```
volatility -f memoria.bin --profile=Win10x64_19041 pslist
```

3. Identificar procesos desconocidos o sin firma.

Resultado: Detección de proceso `svchost.exe` (nombre similar a proceso legítimo pero malicioso).

25.7 Ejemplo práctico 2 – Recuperar contraseñas de navegador

Escenario: Un usuario borró historial, pero tenía sesión activa en Chrome.

Procedimiento:

1. Capturar RAM.

2. Buscar cadenas en memoria:

```
strings memoria.bin | grep -i "password"
```

3. Usar **Volatility** para extraer espacio de memoria del proceso Chrome.

Resultado: Recuperación de credenciales en texto claro para un sitio web clave en la investigación.

25.8 Ejemplo práctico 3 – Extraer conexiones de red activas

Escenario: Servidor sospechoso de enviar datos a IP desconocida.

Procedimiento:

```
volatility -f memoria.bin --profile=Win10x64_19041 netscan
```

Analizar IPs y puertos, geolocalizar IP externa.

Resultado: Detección de conexión saliente a servidor ubicado en Europa del Este.

25.9 Ejemplo práctico 4 – Claves de cifrado en RAM

Escenario: Un disco VeraCrypt está montado y se necesita acceder a él.

Procedimiento:

1. Captura inmediata de RAM.
2. Uso de módulo de Volatility para búsqueda de patrones de clave.
3. Montaje del volumen usando la clave extraída.

Resultado: Acceso completo al disco sin romper el cifrado.

25.10 Buenas prácticas

- Capturar RAM lo antes posible en incidentes en vivo.
- Usar herramientas confiables y documentadas.
- Guardar volcado y hash en repositorio seguro.
- Realizar análisis en equipo aislado, nunca en el original.

25.11 Errores comunes

- Apagar o reiniciar el equipo antes de capturar la memoria.
- Instalar herramientas de adquisición en el disco investigado (usar USB).
- No registrar la versión del sistema operativo y perfil adecuado para análisis.

25.12 Caso real documentado

En 2014, un análisis de RAM permitió descubrir el malware *Chewbacca*, que operaba sin escribir archivos en disco para evadir detección. El volcado mostró la memoria del proceso con datos de tarjetas de crédito capturados en tiempo real.

Resumen del capítulo: El análisis de memoria RAM es esencial para casos de malware avanzado, recuperación de claves y reconstrucción de sesiones activas. Es una técnica delicada que requiere rapidez, precisión y documentación exhaustiva.

Capítulo 26 – Análisis Forense de Correos Electrónicos

26.1 Introducción

El **correo electrónico** es una de las fuentes de evidencia más utilizadas en investigaciones forenses digitales, ya que:

- Se usa para **phishing**, estafas, extorsiones y espionaje.
- Contiene **metadatos valiosos** como fecha, remitente real, IP de origen y servidores intermedios.
- Puede incluir **archivos adjuntos** maliciosos o sensibles.

El análisis forense de emails busca **verificar autenticidad, rastrear su origen y contenido, y preservarlos de forma legalmente admisible**.

26.2 Tipos de evidencia en correos electrónicos

Tipo de evidencia	Ejemplo	Importancia
Contenido del mensaje	Texto del cuerpo del email	Determina el objetivo o intención.
Encabezados (headers)	From, To, Date, Received, Message-ID	Rastro técnico para trazabilidad.
Adjuntos	Documentos, imágenes, scripts	Pueden contener malware o información clave.
Metadatos	Información de creación, rutas de envío	Ayuda a atribuir el mensaje a un origen real.
Logs del servidor	Registros SMTP, IMAP, POP3	Corroboran entrega o recepción.

26.3 Fuentes de datos para la investigación

- **Clientes de correo** (Outlook, Thunderbird).
 - **Servidores corporativos** (Exchange, Postfix, Zimbra).
 - **Webmail** (Gmail, Outlook.com, Yahoo).
 - **Archivos exportados** (.pst, .ost, .mbox, .eml).
-

26.4 Procedimiento general de análisis

1. Preservación

- Exportar email en formato original (EML, MSG, MBOX).
- Calcular hash antes de análisis.

2. Examen de encabezados

- Revisar campos **Received** en orden inverso.
- Identificar IPs de servidores intermedios.
- Verificar dominios y certificados.

3. Verificación de autenticidad

- Revisar firmas SPF, DKIM y DMARC.
- Confirmar si el remitente está falsificado.

4. Análisis de adjuntos

- Escanear con antivirus y en sandbox.
- Analizar metadatos (ExifTool, PDF Parser).

5. Correlación

- Cruzar con logs de servidores, registros DNS y eventos de seguridad.

26.5 Herramientas clave

- **FTK Imager** – Exporta correos y calcula hash.
- **MailXaminer** – Análisis avanzado de emails.
- **X1 Social Discovery** – Recuperación de datos de redes y correo.
- **ExifTool** – Extracción de metadatos de adjuntos.
- **mxtoolbox.com** – Verificación de registros DNS y listas negras.
- **VirusTotal** – Análisis de adjuntos sospechosos.

26.6 Ejemplo práctico 1 – Verificación de remitente real

Escenario: Un usuario recibe un email supuestamente de su banco solicitando datos.

Procedimiento:

1. Abrir encabezados completos (en Gmail: "Mostrar original").

2. Buscar el primer campo **Received** desde abajo:

```
Received: from mail.fakebank.com (203.0.113.45)
```

3. Usar **whois** y geolocalizar IP:

```
whois 203.0.113.45
```

4. Detectar que la IP no pertenece al banco, sino a un servidor en otro país.

Resultado: Confirmación de phishing con remitente falsificado.

26.7 Ejemplo práctico 2 – Detección de malware en adjunto

Escenario: Email con archivo **factura.pdf.exe** recibido por un empleado.

Procedimiento:

1. Descargar adjunto en entorno aislado.
2. Calcular hash y subir a **VirusTotal**.

3. Analizar comportamiento en **Any.Run** (sandbox).
4. Confirmar que se trata de un troyano bancario.

Resultado: Evidencia sólida para proceder con medidas de seguridad y denuncia.

26.8 Ejemplo práctico 3 – Confirmar entrega y lectura

Escenario: Se necesita probar que un email fue entregado antes de una fecha límite legal.

Procedimiento:

1. Revisar logs de servidor SMTP de salida:

```
250 2.0.0 OK id=1q2w3e-0004ef-9g
```

2. Verificar registro de recepción en servidor destino.
3. Revisar reportes de lectura (si existe confirmación de lectura activada).

Resultado: Prueba documental de entrega en plazo.

26.9 Análisis de firmas SPF, DKIM y DMARC

- **SPF**: define qué servidores pueden enviar en nombre del dominio.
- **DKIM**: firma digital que garantiza que el contenido no fue alterado.
- **DMARC**: política de autenticación y reporte.

Ejemplo de verificación en Linux:

```
opendkim-testmsg < correo.eml
```

26.10 Buenas prácticas

- Conservar siempre la versión original del email.
- Analizar encabezados sin alterar el archivo.
- Usar entornos aislados para adjuntos peligrosos.
- Documentar resultados de geolocalización, sandbox y análisis.

26.11 Errores comunes

- Analizar solo el cuerpo del mensaje y no los encabezados.
- Descargar adjuntos en el sistema operativo principal.
- Confiar solo en la dirección "From" sin validar IPs y dominios.

26.12 Caso real documentado

En 2021, una campaña de phishing contra empresas argentinas usó correos con adjuntos de Excel maliciosos. El análisis de encabezados mostró servidores de envío en Europa del Este y plantillas VBA que robaban credenciales de Microsoft 365. Gracias a la preservación de los correos y sus hashes, las pruebas fueron aceptadas en el juicio.

Resumen del capítulo: El análisis forense de correos electrónicos requiere extraer encabezados, verificar autenticidad, analizar adjuntos y correlacionar con registros de servidores. Hecho correctamente, puede ser evidencia clave en investigaciones de fraude, phishing y amenazas corporativas.

Capítulo 27 – Análisis Forense de Dispositivos Móviles

27.1 Introducción

Los **dispositivos móviles** (smartphones y tablets) son una fuente masiva de evidencia digital, ya que concentran:

- Comunicaciones (SMS, llamadas, mensajería instantánea).
- Archivos multimedia (fotos, videos, grabaciones).
- Datos de geolocalización y actividad.
- Aplicaciones financieras, de redes sociales y de trabajo.
- Navegación web e historiales de búsqueda.

En informática forense, el análisis móvil busca **extraer, preservar y analizar información** sin alterar el contenido, garantizando su validez legal.

27.2 Desafíos del análisis móvil

- **Cifrado por defecto** (iOS, Android modernos).
 - **Bloqueo por PIN, patrón o huella**.
 - **Protecciones de fabricante** (Secure Enclave en Apple, Knox en Samsung).
 - **Variedad de modelos, SO y versiones**.
 - **Datos en la nube** (Google Drive, iCloud, OneDrive).
-

27.3 Tipos de extracción forense

Tipo de extracción	Descripción	Ejemplo de herramientas
Lógica	Copia de datos accesibles vía APIs oficiales o conexión USB.	Cellebrite UFED, MOBILedit

Tipo de extracción	Descripción	Ejemplo de herramientas
Física	Imagen bit a bit de la memoria interna, incluyendo espacio borrado.	Cellebrite, XRY, chip-off
File system	Acceso a estructura de archivos completa, sin sectores libres.	Oxygen Forensic Suite
Cloud	Descarga de datos asociados a la cuenta del dispositivo.	Elcomsoft Cloud Extractor

27.4 Procedimiento general

1. Preservación

- Colocar el dispositivo en **modo avión**.
- Guardar en **bolsa Faraday** si se requiere aislamiento RF.
- Evitar apagado si no se conoce el PIN (puede activar cifrado al reiniciar).

2. Documentación

- Fotografiar estado del dispositivo (pantalla, hora, apps abiertas).
- Registrar modelo, número de serie, IMEI.

3. Extracción

- Seleccionar método según acceso disponible (lógico, físico, cloud).
- Usar herramientas certificadas.

4. Análisis

- Revisar llamadas, SMS, WhatsApp, Telegram, GPS, fotos, historial web, etc.
- Buscar datos borrados en imagen física.

5. Reporte

- Presentar hallazgos con capturas y hashes de archivos exportados.

27.5 Herramientas más usadas

- **Cellebrite UFED** – Estándar en fuerzas de seguridad.
- **MSAB XRY** – Extracción y análisis.
- **Oxygen Forensic Detective** – Análisis avanzado de apps y datos de nube.
- **Magnet AXIOM** – Buena integración con evidencias de PC y móviles.
- **ADB (Android Debug Bridge)** – Extracción manual en Android.
- **Elcomsoft iOS Forensic Toolkit** – Extracción de backups y datos cifrados.

27.6 Ejemplo práctico 1 – Extracción lógica de Android

Escenario: Un móvil Android desbloqueado necesita análisis rápido.

Procedimiento:

1. Conectar vía USB con **ADB** habilitado.
2. Listar directorios:

```
adb shell ls /sdcard/
```

3. Copiar carpeta de WhatsApp:

```
adb pull /sdcard/WhatsApp/ /evidencia/whatsapp/
```

4. Calcular hash de cada archivo exportado.

Resultado: Copia completa de chats y medios sin root.

27.7 Ejemplo práctico 2 – Extracción física de iPhone con jailbreak

Escenario: iPhone bloqueado, pero con jailbreak activo.

Procedimiento:

1. Usar **iOS Forensic Toolkit** para volcado físico.
2. Extraer base de datos de mensajes:

```
/private/var/mobile/Library/SMS/sms.db
```

3. Analizar con **DB Browser for SQLite**.

Resultado: Recuperación de mensajes borrados aún presentes en base de datos.

27.8 Ejemplo práctico 3 – Extracción de datos en la nube

Escenario: Se conoce usuario y contraseña de Google asociados al dispositivo.

Procedimiento:

1. Usar **Elcomsoft Cloud Extractor**.
2. Descargar historial de ubicaciones, contactos y Google Drive.
3. Calcular hash y almacenar.

Resultado: Acceso a evidencias aunque el dispositivo físico esté dañado.

27.9 Recuperación de datos borrados en móviles

- En Android, si se dispone de extracción física, se pueden recuperar fotos y documentos de espacio no asignado.
 - En iOS, es más limitado por el cifrado, pero puede recuperarse desde backups antiguos.
-

27.10 Buenas prácticas

- Usar siempre herramientas forenses certificadas.
 - No conectar el dispositivo a redes externas.
 - Documentar todo el proceso con fotos y logs.
 - Mantener cadena de custodia estricta.
-

27.11 Errores comunes

- Apagar dispositivo con cifrado activo (bloquea acceso).
 - Usar métodos no forenses que alteran datos.
 - Conectar a cuenta personal y mezclar evidencias.
-

27.12 Caso real documentado

En 2022, en una investigación de narcotráfico en Argentina, la extracción física de un Android permitió recuperar chats borrados de WhatsApp y coordenadas GPS de reuniones. Los datos fueron presentados en juicio con cadena de custodia completa, resultando en condenas firmes.

Resumen del capítulo: El análisis forense de dispositivos móviles combina conocimientos de hardware, software y seguridad. Es un campo en constante evolución debido a nuevas protecciones y cifrados, por lo que requiere actualización continua.

Capítulo 28 – Análisis Forense de Redes

28.1 Introducción

El **análisis forense de redes** se centra en capturar, examinar y reconstruir el tráfico de datos para identificar:

- Ataques ciberneticos.
- Filtración de información.
- Comunicación con servidores maliciosos.
- Actividad no autorizada en sistemas corporativos.

A diferencia del análisis de disco o memoria, la evidencia de red **es efímera**: si no se capture en el momento, se pierde. Por eso es clave tener configurados sistemas de captura continua o reaccionar rápido ante un incidente.

28.2 Tipos de datos de red útiles en forense

Tipo de dato	Ejemplo	Uso en la investigación
PCAP (packet capture)	Archivos .pcap de Wireshark o tcpdump	Reconstrucción de sesiones y análisis de payloads.
Logs de firewall	Reglas aplicadas, bloqueos	Identificar intentos de intrusión.
NetFlow / sFlow	Estadísticas de flujo IP	Analizar patrones de tráfico.
DNS logs	Consultas y respuestas	Detectar C2 y phishing.
Proxy logs	Navegación HTTP/HTTPS	Rastrear sitios visitados.

28.3 Procedimiento general de análisis de red

1. Captura

- Usar herramientas como **tcpdump**, **Wireshark**, o sensores como Zeek (Bro).
- Preferir captura en puntos estratégicos (switch core, firewall).

2. Preservación

- Guardar capturas en formato original (.pcap).
- Calcular hash para integridad.

3. Análisis

- Reconstruir sesiones TCP.
- Revisar peticiones HTTP, conexiones TLS, consultas DNS.
- Buscar patrones de ataque (scanning, DoS, exfiltración).

4. Correlación

- Cruzar con registros de sistemas, firewalls y antivirus.
- Comparar con inteligencia de amenazas (IOC).

28.4 Herramientas más usadas

- **Wireshark** – Análisis visual de tráfico.
- **tcpdump** – Captura de paquetes desde terminal.
- **Zeek (Bro)** – Monitoreo y análisis avanzado de tráfico.
- **Suricata / Snort** – Detección de intrusos en red.
- **NetworkMiner** – Reconstrucción de archivos y sesiones.
- **Moloch/Arkime** – Indexación y búsqueda en grandes volúmenes de PCAP.

28.5 Ejemplo práctico 1 – Captura en vivo con tcpdump

Escenario: Servidor Linux sospechoso de recibir conexiones maliciosas.

Procedimiento:

```
tcpdump -i eth0 -w captura.pcap
```

- Captura todo el tráfico de la interfaz `eth0`.
- Detener después de unos minutos y analizar en Wireshark.

Resultado: Detección de múltiples intentos de conexión SSH desde IP extranjera.

28.6 Ejemplo práctico 2 – Reconstrucción de sesión HTTP

Escenario: Un usuario descargó un archivo malicioso desde la web.

Procedimiento:

1. Abrir `captura.pcap` en **Wireshark**.
2. Filtrar tráfico HTTP:

```
http.request
```

3. Click derecho en la petición → **Follow TCP Stream**.
4. Guardar contenido como archivo.

Resultado: Recuperación del archivo malicioso y confirmación de su URL de origen.

28.7 Ejemplo práctico 3 – Detección de exfiltración de datos

Escenario: Sospecha de robo de información vía DNS.

Procedimiento:

1. Cargar PCAP en **Zeek**.
2. Revisar logs de DNS (`dns.log`).
3. Detectar consultas con cadenas largas y codificadas (possible *DNS tunneling*).

Resultado: Confirmación de exfiltración usando dominio controlado por atacante.

28.8 Análisis de tráfico cifrado (TLS/HTTPS)

- Revisar metadatos: SNI, certificados, IP de destino.
- Identificar conexiones sospechosas por patrón horario o volumen.
- Si se dispone de clave privada del servidor, descifrar tráfico:

```
tshark -r captura.pcap -o "ssl.keylog_file:clave.key"
```

28.9 Buenas prácticas

- Iniciar captura ante cualquier alerta de seguridad.
- Usar *timestamps* precisos para correlacionar eventos.
- Documentar filtros, herramientas y versiones usadas.
- Mantener PCAP originales sin modificaciones.

28.10 Errores comunes

- Capturar tráfico en el lugar equivocado (sin ver todo el flujo).
- Filtrar demasiado y perder paquetes relevantes.
- No sincronizar relojes de captura con NTP (dificulta correlación).

28.11 Caso real documentado

En 2018, una investigación en una empresa financiera en Buenos Aires detectó que un empleado filtraba datos a un servidor en Europa. El análisis de PCAP reveló que usaba *DNS tunneling* para enviar archivos ZIP cifrados. Con esta evidencia se logró la confesión del implicado.

Resumen del capítulo: El análisis forense de redes es vital para incidentes en tiempo real y detección de ataques. Capturas bien hechas y correlacionadas con otras evidencias pueden ser decisivas en un juicio.

Capítulo 29 – Esteganografía y su Detección Forense

29.1 Introducción

La **esteganografía** es el arte y ciencia de **ocultar información dentro de otro archivo o medio** sin que sea evidente su existencia. A diferencia de la criptografía, que protege el contenido pero no oculta su presencia, la esteganografía **busca pasar desapercibida**.

Ejemplos comunes:

- Mensajes ocultos en imágenes JPEG o PNG.
- Archivos insertados en audio o vídeo.
- Texto escondido en espacios en blanco o en el código de un documento.

En informática forense, detectar esteganografía es crucial porque puede usarse para:

- Comunicaciones secretas entre ciberdelincuentes.

- Exfiltración de datos sin levantar sospechas.
- Almacenamiento encubierto de material ilegal.

29.2 Principios básicos de la esteganografía

- 1. Medio portador (cover file)** El archivo donde se ocultará la información (imagen, audio, vídeo, documento).
- 2. Carga útil (payload)** La información que se quiere esconder (texto, clave, archivo).
- 3. Método de inserción** Algoritmo o técnica usada para ocultar la información.
- 4. Clave o contraseña** Opcional, para proteger el acceso a la información oculta.

29.3 Técnicas comunes de esteganografía

Técnica	Descripción	Ejemplo
LSB (Least Significant Bit)	Modificar el bit menos significativo de cada pixel o muestra de audio.	Ocultar texto en imagen BMP.
Metadatos falsos	Insertar datos en campos EXIF o ID3.	Clave oculta en metadatos de una foto.
Espacios invisibles	Usar caracteres no imprimibles en texto.	Mensaje en saltos de línea.
Uso de contenedores	Archivos comprimidos incrustados en otro formato.	ZIP escondido dentro de un PNG.
Esteganografía de red	Mensajes en cabeceras TCP/DNS.	Datos en campos TTL o ID de paquetes.

29.4 Herramientas para ocultar información

- **Steghide** (imágenes y audio)

```
steghide embed -cf foto.jpg -ef secreto.txt -p clave123
```

- **OpenStego** (interfaz gráfica, multiplataforma)
- **Snow** (ocultación en espacios en blanco en texto)
- **SilentEye** (ocultación en imágenes y audio con GUI)
- **Camouflage** (archivos ejecutables disfrazados de imágenes)

29.5 Procedimiento general de detección forense

1. Preservación

- Calcular hash y trabajar sobre copia.

2. Inspección visual

- Buscar alteraciones sospechosas (ruido extraño, metadatos extraños).

3. Análisis de metadatos

- Herramientas: `exiftool`, `mediainfo`.

```
exiftool foto.jpg
```

4. Búsqueda de firmas

- Usar `binwalk` para detectar archivos incrustados.

```
binwalk archivo.png
```

5. Extracción

- Usar herramientas específicas según el formato.

6. Validación

- Confirmar que el contenido extraído es la carga oculta.

29.6 Ejemplo práctico 1 – Detección en imagen JPEG

Escenario: Se sospecha que una imagen enviada por email contiene datos ocultos.

Procedimiento:

1. Analizar metadatos:

```
exiftool sospechosa.jpg
```

Detectar campo EXIF inusualmente grande.

2. Usar `steghide` para intentar extracción:

```
steghide extract -sf sospechosa.jpg -p clave123
```

3. Extraer archivo `documento.pdf`.

Resultado: Confirmación de uso de esteganografía con LSB.

29.7 Ejemplo práctico 2 – Archivos dentro de un PNG

Escenario: Se recibe un PNG sospechoso de contener malware.

Procedimiento:

```
binwalk -e imagen.png
```

El comando extrae un archivo ZIP escondido. Dentro, se encuentra un ejecutable malicioso.

Resultado: Evidencia de exfiltración de malware disfrazado.

29.8 Ejemplo práctico 3 – Texto oculto en documento Word

Escenario: Un archivo DOCX parece normal pero pesa demasiado.

Procedimiento:

1. Cambiar extensión a `.zip` y extraer.
2. Revisar carpeta `word/media/` y `docProps/`.
3. Encontrar imagen con payload oculto.

Resultado: Recuperación de archivo de texto con direcciones de envío ilegales.

29.9 Buenas prácticas de detección

- Usar múltiples herramientas: ninguna detecta todos los casos.
 - No abrir directamente archivos sospechosos (pueden ejecutar código).
 - Documentar el proceso y los hashes de los archivos extraídos.
-

29.10 Errores comunes

- Confiar solo en análisis visual.
 - No revisar metadatos.
 - No considerar esteganografía en tráfico de red o documentos PDF.
-

29.11 Caso real documentado

En 2010, el FBI descubrió que una red de espías rusos en EE. UU. usaba esteganografía en imágenes publicadas en sitios web para enviar información cifrada. El hallazgo se logró mediante análisis de patrones en bits de imágenes descargadas.

Resumen del capítulo: La esteganografía es una técnica poderosa para ocultar datos y su detección forense requiere una combinación de análisis visual, revisión de metadatos y herramientas especializadas. Una investigación efectiva demanda paciencia, conocimientos técnicos y creatividad.

Capítulo 30 – Criptografía Aplicada en Informática Forense

30.1 Introducción

La **criptografía** es la ciencia de proteger información mediante técnicas matemáticas que la hacen ilegible sin la clave adecuada. En informática forense, la criptografía se presenta en **dos contextos principales**:

1. **Protección legítima de datos** (confidencialidad y privacidad).
2. **Encubrimiento malicioso** (ocultar evidencias, datos robados o comunicaciones ilícitas).

El trabajo forense consiste en:

- Determinar si un dato está cifrado.
 - Evaluar si es posible descifrarlo legal y técnicamente.
 - Documentar intentos y resultados.
-

30.2 Conceptos básicos

- **Cifrado simétrico**: misma clave para cifrar y descifrar (ej. AES, DES, ChaCha20).
 - **Cifrado asimétrico**: par de claves pública y privada (ej. RSA, ECC).
 - **Hash criptográfico**: función que genera huella única, no reversible (ej. SHA-256).
 - **Firmas digitales**: validan autenticidad e integridad de datos.
 - **Criptografía de curva elíptica (ECC)**: usada en dispositivos móviles y blockchain.
-

30.3 Usos legítimos y maliciosos

Uso legítimo	Uso malicioso
ProTEGER comunicaciones (TLS/SSL)	Ocultar malware en archivos cifrados
Cifrar discos corporativos	Cifrar evidencia para evitar análisis
Firmas digitales en contratos	Firmas falsas para manipular documentos
VPN corporativa	Tunelización para exfiltrar datos

30.4 Cómo detectar datos cifrados

1. **Análisis de entropía** Datos cifrados presentan entropía cercana a 8 bits/byte.

```
binwalk --entropy archivo.bin
```

2. **Extensiones comunes** .aes, .gpg, .pgp, .enc.
 3. **Estructura de encabezados** Archivos PGP empiezan con bytes específicos (0x99).
-

30.5 Procedimiento general de análisis forense de datos cifrados

1. Preservación

- Calcular hash y trabajar sobre copia.

2. Identificación

- Reconocer tipo de cifrado o contenedor.

3. Adquisición de claves

- Búsqueda en memoria RAM.
- Recuperación de credenciales desde navegadores, gestores de contraseñas, archivos temporales.

4. Intento de descifrado

- Ataques de diccionario, fuerza bruta o híbridos.
- Herramientas como hashcat o john the ripper.

5. Documentación

- Registrar herramientas, parámetros y resultados.
-

30.6 Herramientas clave

- **GnuPG** – Cifrado/descifrado PGP.
 - **VeraCrypt** – Montar volúmenes cifrados (si clave disponible).
 - **hashcat** – Cracking de hashes y contraseñas.
 - **John the Ripper** – Ataques de diccionario y fuerza bruta.
 - **Cryptohaze Multiforcer** – Ataques masivos por GPU.
 - **FTK Imager** – Detección de volúmenes cifrados.
-

30.7 Ejemplo práctico 1 – Recuperar clave de volumen VeraCrypt desde RAM

Escenario: Disco cifrado montado durante incautación.

Procedimiento:

1. Capturar memoria RAM inmediatamente.

2. Usar **volatility** para buscar patrones de clave:

```
volatility -f memoria.bin --profile=Win10x64_19041 strings | grep -i  
veracrypt
```

3. Extraer clave y montar volumen en modo solo lectura.

Resultado: Acceso total al contenido cifrado sin romper el algoritmo.

30.8 Ejemplo práctico 2 – Descifrado de archivo PGP con clave encontrada

Escenario: Archivo **mensajes.pgp** sospechoso.

Procedimiento:

1. Analizar PC del sospechoso en busca de **.gnupg/secring.gpg**.
2. Usar:

```
gpg --decrypt mensajes.pgp
```

con la clave encontrada y passphrase recuperada.

Resultado: Texto plano revelando comunicaciones ilegales.

30.9 Ejemplo práctico 3 – Hash cracking

Escenario: Archivo con contraseñas en SHA-256 sin sal.

Procedimiento:

1. Usar diccionario de contraseñas:

```
hashcat -m 1400 hashes.txt rockyou.txt
```

2. Comparar resultados y validar.

Resultado: Recuperación de credenciales para otras evidencias.

30.10 Buenas prácticas

- Documentar todos los intentos de descifrado.
- No alterar los archivos cifrados originales.

- Usar entornos aislados para pruebas de cracking.
 - Mantener confidencialidad si se recupera información sensible.
-

30.11 Errores comunes

- Intentar descifrado sin evaluar legalidad.
 - Usar herramientas sin control de logs.
 - No capturar RAM cuando el volumen estaba montado.
-

30.12 Caso real documentado

En 2016, un investigador forense en Alemania recuperó claves de cifrado TrueCrypt analizando volcados de RAM de equipos incautados. La clave estaba en texto claro en un segmento de memoria porque el contenedor estaba montado en el momento de la captura.

Resumen del capítulo: La criptografía es un reto en informática forense: si bien es extremadamente segura, los errores humanos (claves en RAM, contraseñas débiles) pueden abrir la puerta a su recuperación. El análisis requiere tanto conocimientos técnicos como procedimientos legales claros.

Capítulo 31 – Análisis de Malware y Reverse Engineering

31.1 Introducción

El **análisis de malware** es la disciplina que estudia programas maliciosos para comprender su funcionamiento, origen y posibles mitigaciones. En informática forense, el objetivo no es solo identificar el malware, sino **documentar de forma exhaustiva** su comportamiento y preservar la evidencia para un eventual proceso judicial. La **ingeniería inversa** (reverse engineering) es la técnica que permite desensamblar y analizar código binario para entenderlo sin disponer del código fuente. Esto se vuelve crítico cuando el malware es desconocido o altamente ofuscado. Analizar malware implica riesgos técnicos (ejecución no controlada, infección de sistemas) y legales (posible manipulación de software protegido), por lo que debe hacerse siempre en entornos **sandbox** o máquinas virtuales aisladas, con **snapshots** listos para revertir el estado del sistema.

31.2 Tipos de malware

Tipo	Descripción	Ejemplo
Virus	Infectan archivos ejecutables y se replican al abrirlos.	CIH, Salty
Gusanos (Worms)	Se propagan automáticamente por la red.	WannaCry
Troyanos	Se disfrazan de software legítimo.	Zeus

Tipo	Descripción	Ejemplo
Ransomware	Cifran archivos y exigen pago.	Locky, REvil
Spyware	Espían la actividad del usuario.	FinFisher
Rootkits	Ocultan procesos y archivos para persistencia.	Alureon
Botnets	Dispositivos infectados controlados remotamente.	Mirai

31.3 Etapas del análisis de malware

1. Análisis estático

- Examinar el archivo sin ejecutarlo.
- Herramientas: **strings**, **binwalk**, **PEiD**, **die**.
- Objetivos: detectar *strings* sospechosos, bibliotecas usadas, empaquetadores.

Ejemplo:

```
strings malware.exe | grep http
```

Salida: URL de servidor C2 (Command and Control).

2. Análisis dinámico

- Ejecutar el malware en entorno controlado.
- Herramientas: **Cuckoo Sandbox**, **Any.Run**, **Process Monitor**.
- Objetivos: registrar cambios en sistema, tráfico de red, procesos creados.

3. Ingeniería inversa

- Desensamblar y descompilar binario.
- Herramientas: **IDA Pro**, **Ghidra**, **Radare2**.
- Objetivos: descubrir funciones ocultas, desencriptar cadenas, entender lógica.

31.4 Indicadores de Compromiso (IOCs)

Los **IOCs** son rastros que deja el malware y permiten identificar su presencia.

Ejemplos:

- Hashes de archivos maliciosos (MD5, SHA256).
- Nombres de procesos inusuales.
- Claves de registro modificadas.
- IPs y dominios de C2.

Formato de tabla IOC:

Tipo	Valor	Observaciones
SHA256	3f4d...e91a	Binario detectado en análisis
IP	185.203.118.5	Servidor C2 en Rumanía
Dominio	update-service.info	Usado para descarga de payload

31.5 Ejemplo práctico 1 – Detección de comportamiento malicioso en Windows

Escenario: Archivo `factura.pdf.exe` detectado en un correo.

Procedimiento:

1. Calcular hash y registrarlo.
2. Ejecutar en VM con **Process Monitor** activado.
3. Observar:
 - Creación de `C:\Users\User\AppData\Roaming\update.exe`.
 - Modificación de clave `HKCU\Software\Microsoft\Windows\CurrentVersion\Run`.
4. Capturar tráfico con Wireshark → conexión a IP desconocida por puerto 8080.

Resultado: Confirmación de persistencia y comunicación con C2.

31.6 Ejemplo práctico 2 – Extracción de clave de cifrado de Ransomware

Escenario: Ransomware activo en VM cifrando archivos.

Procedimiento:

1. Usar **Process Explorer** para identificar PID del malware.
2. Abrir con **x64dbg** y buscar en memoria secuencia de clave AES.
3. Guardar clave y probar descifrado con herramienta personalizada.

Resultado: Recuperación de archivos sin pagar rescate.

31.7 Buenas prácticas en análisis de malware

- Usar siempre máquinas virtuales aisladas sin conexión a redes corporativas.
- Tomar *snapshots* antes de ejecución.
- Documentar cada paso y capturas de pantalla.
- No subir muestras a servicios públicos si es investigación sensible.
- Asegurar que hashes y nombres de archivo se registren en informe final.

31.8 Errores comunes

- Ejecutar malware en máquina host sin aislamiento.
 - Confiar solo en antivirus para detección.
 - No calcular hashes antes de manipular archivo.
 - No registrar IOCs para correlación posterior.
-

31.9 Caso real documentado

En 2017, WannaCry infectó miles de equipos en Argentina. El análisis forense reveló que el vector inicial fue un escaneo masivo de SMB y la ejecución automática del exploit *EternalBlue*. Capturar la muestra y aislarla permitió generar IOCs y bloquear la propagación en varias organizaciones locales.

Resumen del capítulo: El análisis de malware y la ingeniería inversa son fundamentales para entender amenazas, generar defensas y producir evidencia sólida. El éxito depende de un enfoque metódico, entornos seguros y documentación precisa.

Capítulo 32 – Análisis Forense en Entornos Virtualizados y Cloud

32.1 Introducción

La adopción masiva de **virtualización** y **computación en la nube** ha transformado la forma en que las organizaciones gestionan sus recursos. Sin embargo, estos entornos plantean nuevos retos a la informática forense: múltiples capas de abstracción, recursos distribuidos geográficamente, datos volátiles y dependencia de terceros para acceder a la información. En un servidor físico tradicional, el forense puede incautar el disco y analizarlo directamente. En un entorno virtualizado, los datos pueden estar en **imágenes de discos virtuales**, snapshots, o incluso almacenados en sistemas distribuidos bajo control de un proveedor de nube. Además, los entornos cloud (AWS, Azure, Google Cloud) exigen conocimientos de APIs, logs específicos y procedimientos legales para acceder a información, lo que convierte al perito en un investigador tanto técnico como jurídico.

32.2 Principales desafíos

Desafío	Descripción	Impacto
Volatilidad de datos	Recursos creados y destruidos en segundos (instancias, contenedores).	Evidencia efímera, difícil de preservar.
Dependencia del proveedor	Acceso a logs y snapshots limitado por políticas.	Necesidad de órdenes judiciales o acuerdos.
Multitenencia	Varios clientes comparten recursos físicos.	Riesgo de acceso a datos de terceros.

Desafío	Descripción	Impacto
Jurisdicción	Datos alojados en otro país.	Conflictos legales internacionales.

32.3 Evidencias típicas en entornos virtualizados y cloud

- **Archivos de disco virtual** (`.vmdk`, `.vdi`, `.qcow2`).
- **Snapshots** de máquinas virtuales.
- **Logs de hipervisor** (VMware ESXi, Hyper-V, Proxmox).
- **Registros de consola de administración**.
- **Logs de servicios cloud** (CloudTrail en AWS, Activity Logs en Azure, Cloud Audit Logs en GCP).
- **Capturas de tráfico virtual** (vSwitch, tap interfaces).

32.4 Procedimiento forense general

1. Identificación

- Localizar el recurso sospechoso (VM, contenedor, bucket S3, instancia EC2).
- Determinar proveedor y tecnología usada.

2. Preservación

- Crear snapshot inmutable de la instancia/disco virtual.
- Exportar en formato forense y calcular hash.

3. Adquisición

- Descargar imagen de disco virtual o exportar datos desde API.
- Ejemplo AWS:

```
aws ec2 create-snapshot --volume-id vol-1234567890abcdef0
```

4. Análisis

- Montar imagen en entorno seguro.
- Revisar logs de hipervisor o consola.
- Correlacionar con datos de red y registros cloud.

5. Reporte

- Incluir detalles de API usadas, hashes, permisos solicitados y cadena de custodia.

32.5 Ejemplo práctico 1 – Recuperación de datos en VM de VMware

Escenario: Un servidor virtual de contabilidad es borrado intencionalmente.

Procedimiento:

1. Acceder al datastore de VMware ESXi.
2. Localizar archivo **.vmdk** de la VM.
3. Clonar y montar imagen con FTK Imager.
4. Recuperar archivos borrados con Autopsy.

Resultado: Recuperación de registros contables eliminados horas antes.

32.6 Ejemplo práctico 2 – Investigación en AWS

Escenario: Sospecha de acceso no autorizado a un bucket S3.

Procedimiento:

1. Revisar logs de **AWS CloudTrail**:

```
aws cloudtrail lookup-events --lookup-attributes  
AttributeKey=EventName,AttributeValue=GetObject
```

2. Identificar IP origen y usuario IAM.
3. Correlacionar con registros de facturación y geolocalización.

Resultado: Confirmación de descarga no autorizada por credenciales comprometidas.

32.7 Ejemplo práctico 3 – Contenedores Docker

Escenario: Un contenedor ejecuta un script de minería de criptomonedas.

Procedimiento:

1. Listar contenedores:

```
docker ps -a
```

2. Crear copia del sistema de archivos del contenedor:

```
docker export <container_id> > contenedor.tar
```

3. Analizar scripts y binarios extraídos.

Resultado: Identificación de malware de minería y direcciones de monedero asociadas.

32.8 Buenas prácticas

- Trabajar con permisos de solo lectura en snapshots.
 - Documentar cada comando y API usada para preservar cadena de custodia.
 - Usar herramientas nativas del proveedor siempre que sea posible.
 - Solicitar asistencia del equipo de seguridad del proveedor si es un caso legal.
-

32.9 Errores comunes

- Descargar datos sin calcular hash inicial.
 - No preservar metadatos de API y logs del proveedor.
 - Montar discos virtuales directamente en sistemas de análisis sin aislamiento.
-

32.10 Caso real documentado

En 2021, un equipo forense en Brasil investigó un ataque de ransomware en infraestructura híbrida (VMware + AWS). Al combinar snapshots de VMs locales con logs de AWS CloudTrail, se detectó que el acceso inicial fue a través de una cuenta IAM comprometida. Esto permitió reconstruir la secuencia exacta de comandos ejecutados por el atacante.

Resumen del capítulo: El análisis forense en entornos virtualizados y cloud requiere conocimientos técnicos, legales y procedimentales avanzados. Las evidencias son volátiles, distribuidas y dependientes de terceros, por lo que actuar rápido y con metodología es esencial.

Capítulo 33 – Internet de las Cosas (IoT) y Dispositivos Inteligentes en Informática Forense

33.1 Introducción

El **Internet de las Cosas (IoT)** engloba todo aquel conjunto de dispositivos físicos conectados a la red que recopilan, procesan e intercambian datos sin intervención humana directa. Esto incluye desde cámaras de seguridad IP, pulseras deportivas, electrodomésticos inteligentes, hasta vehículos conectados. En la informática forense, el IoT presenta oportunidades y retos únicos: estos dispositivos generan grandes volúmenes de datos potencialmente valiosos (geolocalización, imágenes, registros de uso), pero muchas veces están limitados en almacenamiento, cifran la información o dependen de servicios en la nube controlados por terceros. La evidencia IoT puede ser la pieza clave en un caso: un reloj inteligente registrando actividad física durante un crimen, o una cerradura inteligente mostrando accesos en horarios críticos.

33.2 Retos en la investigación forense IoT

Desafío	Descripción	Impacto
---------	-------------	---------

Desafío	Descripción	Impacto
Volatilidad de datos	Los registros se sobrescriben rápido por capacidad limitada.	Riesgo de pérdida si no se actúa pronto.
Diversidad de sistemas	Cada fabricante usa su propio firmware y protocolos.	Necesidad de conocimientos muy variados.
Dependencia de la nube	Datos almacenados en servidores remotos.	Requiere acceso legal al proveedor.
Cifrado propietario	Formatos no documentados, cifrado a medida.	Obstaculiza análisis sin ingeniería inversa.

33.3 Fuentes de evidencia en IoT

- **Almacenamiento local del dispositivo** (memoria flash, tarjeta SD).
- **Aplicaciones móviles vinculadas** (logs, credenciales).
- **Tráfico de red del dispositivo** (PCAPs de comunicación).
- **Plataformas cloud asociadas** (AWS IoT, Google Cloud IoT Core).
- **Logs de control de acceso** (cerraduras inteligentes, sistemas de alarma).

33.4 Procedimiento forense general

1. Preservación física

- Desconectar de la red si hay riesgo de borrado remoto.
- Proteger contra escritura (en memorias extraíbles).

2. Adquisición de datos

- Conexión por puertos de depuración (UART, JTAG).
- Extracción de imágenes de firmware.
- Descarga de logs de la aplicación asociada.

3. Análisis

- Examinar bases de datos SQLite o JSON locales.
- Revisar metadatos de archivos (imágenes, audios).
- Analizar patrones de conexión en tráfico capturado.

4. Correlación

- Cruzar información con otros sistemas (cámaras, geolocalización).
- Comparar con registros de red corporativa o ISP.

33.5 Ejemplo práctico 1 – Cerradura inteligente

Escenario: Se sospecha que una cerradura inteligente fue manipulada para un robo.

Procedimiento:

1. Extraer logs desde la app móvil vinculada.
2. Analizar registros:

```
{  
    "fecha": "2025-08-05T02:13:00Z",  
    "usuario": "desconocido",  
    "metodo": "codigo"  
}
```

3. Correlacionar con cámara de seguridad.

Resultado: Se confirma acceso no autorizado a las 02:13.

33.6 Ejemplo práctico 2 – Reloj inteligente en investigación criminal

Escenario: Un sospechoso afirma estar dormido durante un homicidio.

Procedimiento:

1. Extraer datos de actividad física del smartwatch (pasos, ritmo cardíaco).
2. Herramienta: [Health Data Importer](#) para exportar datos en CSV.
3. Analizar:

```
2025-08-01 23:45 - 120 bpm  
2025-08-01 23:47 - 132 bpm (actividad intensa)
```

Resultado: Evidencia contradice coartada, mostrando actividad física en horario del crimen.

33.7 Ejemplo práctico 3 – Cámara IP y tráfico de red

Escenario: Cámara IP comprometida transmite vídeo a servidor externo.

Procedimiento:

1. Capturar tráfico con [tcpdump](#):

```
tcpdump -i eth0 -w camara.pcap
```

2. Analizar en Wireshark, filtrar conexiones salientes.
3. Detectar transmisión continua a IP en otro país.

Resultado: Confirmación de exfiltración de vídeo en tiempo real.

33.8 Buenas prácticas

- Intervenir lo antes posible para evitar sobrescritura de datos.
 - Documentar conexiones físicas y estado inicial del dispositivo.
 - Mantener copias bit a bit del firmware y almacenamiento.
 - No actualizar el firmware antes de extraer datos.
-

33.9 Errores comunes

- Encender el dispositivo sin medidas de aislamiento (puede borrar logs).
 - No capturar tráfico de red antes de desconexión.
 - Ignorar la aplicación móvil asociada como fuente de evidencia.
-

33.10 Caso real documentado

En 2019, un caso en EE. UU. fue resuelto gracias a datos de un altavoz inteligente que registró comandos de voz en la hora del crimen. La correlación de estos registros con el tráfico de red y la ubicación del dispositivo confirmó la presencia del sospechoso en la escena.

Resumen del capítulo: Los dispositivos IoT pueden ser testigos silenciosos de incidentes críticos. Un investigador forense debe combinar técnicas de adquisición física, análisis de red y extracción de datos en la nube para obtener el máximo valor de estas evidencias.

Capítulo 34 – Blockchain y Criptomonedas en Investigación Forense

34.1 Introducción

El **Blockchain** es una tecnología de registro distribuido que almacena transacciones en bloques enlazados criptográficamente. Es la base de criptomonedas como **Bitcoin**, **Ethereum**, **Monero**, entre muchas otras. En el contexto de la informática forense, el Blockchain presenta un escenario peculiar: es **público y transparente** en la mayoría de las redes, pero las identidades de los participantes están seudonimizadas. Esto significa que cualquiera puede ver todas las transacciones, pero vincularlas a una persona real requiere técnicas de análisis adicionales, acceso a datos externos o intervención de exchanges. Las criptomonedas son atractivas para cibercriminales debido a su naturaleza descentralizada, la rapidez de las transferencias y, en algunos casos, el alto nivel de anonimato que ofrecen.

34.2 Tipos de criptomonedas y su impacto forense

Tipo	Ejemplos	Nivel de anonimato	Dificultad de rastreo
Transparencia total	Bitcoin, Litecoin	Medio (todas las transacciones visibles)	Media
Privacidad mejorada	Zcash, Dash	Alta (cifrado selectivo de datos)	Alta
Anonimato total	Monero	Muy alta (ofuscación nativa)	Muy alta

34.3 Retos en el análisis forense de criptomonedas

1. Seudonimato

- Direcciones no están asociadas directamente a identidades reales.

2. Uso de mixers

- Servicios que mezclan fondos para romper la trazabilidad.

3. Exchanges no regulados

- Operan en jurisdicciones con poca cooperación internacional.

4. Monedas centradas en la privacidad

- Monero y Zcash limitan severamente el análisis público.

34.4 Fuentes de evidencia

- **Registros públicos en el Blockchain**

- Exploradores como *blockchain.com*, *etherscan.io*.

- **Logs de exchanges**

- Información de KYC (Know Your Customer) y direcciones usadas.

- **Wallets locales**

- Archivos de clave privada (*wallet.dat*, *keystore*).

- **Capturas de tráfico de red**

- Puede contener datos de transacciones antes de propagarse.

- **Metadatos en correos o chats**

- Direcciones de wallets compartidas.

34.5 Procedimiento forense general

1. Identificación

- Detectar direcciones de wallet o hashes de transacciones relevantes.

2. Preservación

- Capturar información del Blockchain y exportar en formato seguro.

3. Análisis de transacciones

- Seguir el rastro de fondos usando herramientas de *blockchain analysis*.

4. Correlación con datos externos

- Vincular direcciones a usuarios de exchanges mediante KYC.

5. Reporte

- Incluir gráficas de flujo de fondos y análisis temporal.
-

34.6 Herramientas de análisis

- **Chainalysis** (comercial) – Análisis avanzado y visualización de transacciones.
 - **CipherTrace** (comercial) – Inteligencia financiera blockchain.
 - **Blockchain.com Explorer** – Explorador gratuito de Bitcoin.
 - **Etherscan.io** – Explorador de Ethereum y tokens ERC-20.
 - **GraphSense** (open source) – Análisis de grafos de transacciones.
-

34.7 Ejemplo práctico 1 – Rastrear transacción de ransomware en Bitcoin

Escenario: Una víctima paga 0.5 BTC a un wallet indicado por los atacantes.

Procedimiento:

1. Buscar transacción en *blockchain.com* por ID.
2. Seguir transferencias posteriores.
3. Detectar envío a un exchange regulado.
4. Solicitar información a través de orden judicial.

Resultado: Identificación del titular de la cuenta receptora.

34.8 Ejemplo práctico 2 – Recuperar wallet.dat de disco incautado

Escenario: Se encuentra un archivo **wallet.dat** de Bitcoin en un HDD.

Procedimiento:

1. Copiar archivo y calcular hash.
2. Usar **bitcoin-core** para importarlo:

```
bitcoin-cli importwallet wallet.dat
```

3. Extraer direcciones y saldos.

Resultado: Acceso a fondos y transacciones históricas.

34.9 Ejemplo práctico 3 – Análisis de Ethereum y contratos inteligentes

Escenario: Sospecha de fraude en token ERC-20.

Procedimiento:

1. Analizar contrato en *etherscan.io*.
2. Revisar funciones y transferencias.
3. Detectar *minting* no autorizado de tokens.

Resultado: Confirmación de manipulación del contrato para emitir tokens sin respaldo.

34.10 Buenas prácticas

- Documentar cada consulta y captura del Blockchain.
- Usar múltiples fuentes para validar datos (exploradores y APIs).
- Mantener entornos aislados para trabajar con wallets recuperadas.

34.11 Errores comunes

- Confiar únicamente en exploradores sin respaldo de datos.
- No calcular hash de archivos de wallets antes de analizarlos.
- Ignorar tokens o criptomonedas menos conocidas que pueden ocultar fondos.

34.12 Caso real documentado

En 2020, un grupo de estafadores en Argentina fue identificado tras rastrear pagos en Bitcoin realizados por víctimas de una estafa piramidal. El análisis forense blockchain reveló el uso de un exchange local regulado, lo que permitió vincular direcciones a identidades reales mediante datos KYC.

Resumen del capítulo: El análisis forense de blockchain y criptomonedas requiere combinar técnicas de rastreo público con datos privados de exchanges y otros intermediarios. Aunque el anonimato es un reto, los errores humanos y la interacción con sistemas regulados suelen abrir la puerta a la identificación.

Capítulo 35 – Legislación Vigente en Argentina

35.1 Introducción

La informática forense en Argentina se encuentra en la intersección entre la **tecnología**, el **derecho penal**, el **derecho procesal** y la **protección de datos personales**. Si bien no existe una "Ley de Informática Forense" como tal, hay un conjunto de normas, leyes y resoluciones que regulan el manejo de evidencia digital, el cibercrimen y la privacidad. Un perito forense debe conocer estas leyes no solo para **recolectar y analizar evidencias legalmente**, sino también para asegurar que su trabajo sea **admisible en juicio** y no viole derechos constitucionales.

35.2 Marco legal principal

Norma / Ley	Tema principal	Aplicación en informática forense
Ley 25.326 (Protección de Datos Personales)	Regula el tratamiento de datos personales.	Limita el acceso y almacenamiento de información sensible.
Ley 26.388 (Delitos Informáticos)	Incorpora delitos informáticos al Código Penal.	Define hacking, fraude informático y daño de datos.
Código Procesal Penal Federal	Procedimiento de obtención y preservación de prueba.	Determina cómo se deben incautar dispositivos y datos.
Ley 27.078 (Argentina Digital)	Marco regulatorio de telecomunicaciones y TICs.	Regula acceso a redes y responsabilidades de proveedores.
Ley 11.723 (Propiedad Intelectual)	Protección de software y obras digitales.	Define delitos de copia no autorizada.
Ley 26.388 – Art. 153 a 155	Protección de la intimidad.	Penaliza acceso indebido a comunicaciones electrónicas.

35.3 Principios jurídicos clave

1. Cadena de custodia

- La evidencia digital debe preservarse intacta, documentando cada manipulación.

2. Admisibilidad

- La obtención debe respetar derechos constitucionales (art. 18 CN: inviolabilidad de correspondencia y domicilio).

3. Proporcionalidad

- Las medidas deben ser proporcionales al hecho investigado.

4. Autenticidad

- Uso de hashes para demostrar integridad de la prueba.

35.4 Procedimiento legal típico en incautación digital

1. Orden judicial

- Excepto casos de flagrancia o urgencia extrema.

2. Allanamiento

- Se asegura el lugar y se evita la manipulación previa de equipos.

3. Preservación

- Desconexión física, imágenes forenses bit a bit.

4. Documentación

- Acta detallada con dispositivos, números de serie, estado, hora y firmas.

5. Análisis

- En laboratorio autorizado o designado por el juez.

35.5 Ejemplo práctico – Aplicación de Ley 26.388

Escenario: Un empleado accede sin autorización a la base de datos de clientes y vende la información.

Aplicación legal:

- **Art. 153 bis:** acceso indebido a sistema informático protegido.
- **Art. 157 bis:** revelación de datos personales sin consentimiento.
- **Sanciones:** penas de prisión y multas.

35.6 Obligaciones del perito forense

- Respetar el alcance de la orden judicial.
- No acceder a datos irrelevantes para la investigación.
- Mantener confidencialidad sobre la información obtenida.
- Declarar en juicio en calidad de testigo experto si es requerido.

35.7 Buenas prácticas legales

- Verificar siempre que la recolección de evidencia esté respaldada por orden judicial.
- Usar herramientas que registren logs de todas las operaciones realizadas.
- Evitar manipular dispositivos encendidos sin antes documentar su estado.

35.8 Errores comunes

- Realizar peritajes sin autorización judicial previa.
- No registrar adecuadamente la cadena de custodia.

- Usar herramientas que modifiquen metadatos sin dejar registro.
-

35.9 Caso real documentado

En 2018, un caso de fraude bancario en Buenos Aires fue desestimado parcialmente porque los dispositivos fueron analizados sin orden judicial previa y sin cadena de custodia documentada. Esto llevó a que las pruebas fueran declaradas **inadmisibles**, a pesar de que contenían evidencias claras del delito.

Resumen del capítulo: En Argentina, la labor forense digital debe ajustarse estrictamente al marco legal. El desconocimiento o incumplimiento de las leyes puede invalidar por completo una investigación, sin importar la solidez técnica de la evidencia.

Capítulo 36 – Legislación Internacional y Convenios

36.1 Introducción

En un mundo hiperconectado, los delitos informáticos rara vez respetan fronteras. Un ataque originado en un país puede afectar a víctimas en varios continentes y utilizar infraestructura distribuida globalmente. En informática forense, esto significa que la evidencia digital puede estar **fragmentada en múltiples jurisdicciones**, alojada en servidores de diferentes países, o bajo custodia de proveedores internacionales que responden a marcos legales ajenos al nuestro. Para afrontar estos casos, los investigadores deben apoyarse en **convenios internacionales, tratados bilaterales y marcos de cooperación** que faciliten el intercambio de información y asistencia judicial.

36.2 Principales convenios y tratados

Convenio / Tratado	Alcance	Aplicación forense
Convenio de Budapest (2001)	Tratado del Consejo de Europa sobre ciberdelincuencia.	Establece estándares para tipificación de delitos y cooperación internacional.
Convenio Interamericano contra la Ciberdelincuencia (OEA)	Miembros de la OEA.	Fomenta armonización legislativa y asistencia judicial recíproca.
MLAT (Mutual Legal Assistance Treaties)	Tratados bilaterales de asistencia legal.	Solicitudes formales de evidencia digital entre países.
Convención de Palermo (ONU)	Lucha contra el crimen organizado transnacional.	Aplica cuando el cibercrimen es parte de actividades criminales mayores.

36.3 Retos legales en investigaciones internacionales

1. Jurisdicciones conflictivas

- Leyes de privacidad o protección de datos más estrictas que impiden compartir información.

2. Diferencias en tipificación

- Lo que es delito en un país puede no serlo en otro.

3. Tiempos de respuesta

- Procesos judiciales internacionales pueden tardar meses.

4. Falta de cooperación

- Estados que no son parte de convenios clave o que limitan colaboración.
-

36.4 Procedimientos típicos de cooperación internacional

1. Solicitud de asistencia legal mutua (MLA)

- Se envía a través de las cancillerías o ministerios de justicia.

2. Preservación de datos

- El país receptor ordena a un proveedor conservar la información mientras se procesa la solicitud.

3. Entrega formal de evidencia

- Se transfiere la información cumpliendo requisitos legales del país solicitante y del proveedor.

4. Uso en juicio

- La evidencia debe presentarse con documentación que respalde su autenticidad.
-

36.5 Fuentes de evidencia en contextos internacionales

- **Proveedores de servicios globales**

- Google, Microsoft, Meta, Amazon Web Services.

- **Registros de dominio y WHOIS**

- ICANN y registradores regionales.

- **Logs de CDNs y servicios de entrega de contenido**

- Cloudflare, Akamai.

- **Datos de tráfico internacional**

- Puntos de intercambio de internet (IXPs).
-

36.6 Ejemplo práctico 1 – Ransomware con servidor C2 en Europa

Escenario: Un ataque de ransomware en Argentina utiliza un servidor de comando y control en Alemania.

Procedimiento:

1. Identificar IP del C2 mediante análisis de tráfico.
2. Solicitar preservación de datos al ISP alemán vía convenio de Budapest.
3. Obtener logs que revelan la dirección IP de conexión del atacante.

Resultado: Localización de sospechoso en país vecino.

36.7 Ejemplo práctico 2 – Fraude con criptomonedas en Asia

Escenario: Víctimas argentinas transfieren fondos a un exchange en Singapur.

Procedimiento:

1. Recolectar hashes de transacciones en blockchain.
2. Solicitar a Singapur datos KYC del exchange mediante MLAT.
3. Vincular identidad a pasaporte usado en registro.

Resultado: Identificación de la persona que retiró los fondos.

36.8 Buenas prácticas en cooperación internacional

- Conocer qué convenios internacionales ha firmado tu país.
- Iniciar preservación de datos de inmediato para evitar pérdida.
- Mantener comunicación directa con puntos de contacto de cibercrimen (24/7 contact points).
- Preparar solicitudes claras y específicas para agilizar el proceso.

36.9 Errores comunes

- Solicitudes vagas que obligan a rehacer el trámite.
- No considerar la diferencia en zonas horarias al pedir logs.
- Esperar demasiado antes de activar canales de cooperación.

36.10 Caso real documentado

En 2022, un ataque de phishing contra empresas argentinas fue rastreado hasta un servidor en Estonia. Gracias al Convenio de Budapest, la fiscalía argentina logró que Estonia preservara los registros y entregara la información en menos de 15 días, permitiendo identificar a una red de ciberdelincuentes en tres países.

Resumen del capítulo: La legislación internacional y los convenios de cooperación son esenciales para enfrentar el cibercrimen transnacional. El tiempo, la precisión en las solicitudes y el conocimiento de los marcos legales son claves para el éxito de una investigación.

Capítulo 37 – Gestión de Evidencia Digital y Cadena de Custodia

37.1 Introducción

En cualquier investigación forense, la evidencia digital solo tiene valor si se puede **demostrar su autenticidad e integridad**. Esto es especialmente crítico en entornos judiciales, donde la defensa buscará invalidar la prueba si detecta cualquier irregularidad. La **gestión de evidencia digital** abarca todas las actividades relacionadas con la **recolección, preservación, almacenamiento, transporte y análisis** de datos digitales. La **cadena de custodia**, por su parte, es el conjunto de procedimientos documentados que certifican que la evidencia se mantuvo intacta desde su incautación hasta su presentación en juicio.

37.2 Conceptos clave

Concepto	Definición	Relevancia forense
Evidencia digital	Información almacenada o transmitida en formato electrónico susceptible de ser usada como prueba.	Base de todo análisis forense.
Cadena de custodia	Registro cronológico de la posesión, transferencia y manipulación de la evidencia.	Garantiza integridad y admisibilidad.
Integridad	Ausencia de modificaciones no autorizadas en los datos.	Evita impugnaciones legales.
Autenticidad	Prueba de que la evidencia es lo que se afirma que es.	Respaldada por hashes y documentación.

37.3 Etapas de la gestión de evidencia digital

1. Identificación

- Localizar posibles fuentes de evidencia (HDD, SSD, smartphones, logs, nube).
- Determinar relevancia según el caso.

2. Preservación

- Evitar alteraciones.
- Usar bloqueadores de escritura y técnicas de imagen forense.

3. Adquisición

- Clonar medios completos (bit a bit).
- Generar hash antes y después para validar integridad.

4. Almacenamiento

- Guardar en contenedores seguros, con acceso restringido.

- Etiquetar con código único de identificación.

5. Transporte

- Usar embalaje sellado y registro de traslado.
- Incluir firma de cada persona que recibe la evidencia.

6. Análisis

- Trabajar siempre sobre copias forenses.
- Documentar cada procedimiento.

7. Presentación

- Redactar informe pericial con metodología, resultados y conclusiones.

37.4 Ejemplo de formulario de cadena de custodia

ID Evidencia	Descripción	Fecha/Hora	De	Para	Firma
001-HDD-2025	Disco rígido 1TB Seagate	08/08/2025 10:30	Oficial Pérez	Perito Vera	[Firma]

37.5 Ejemplo práctico 1 – Incautación de servidor corporativo

Escenario: Servidor con registros de acceso sospechosos.

Procedimiento:

1. Fotografiar estado inicial (cables, pantallas, LEDs).
2. Etiquetar servidor y embalaje.
3. Retirar HDD usando guantes y colocar en bolsa antiestática.
4. Documentar en formulario de cadena de custodia.
5. Realizar imagen forense en laboratorio.

Resultado: Integridad preservada, evidencia aceptada en juicio.

37.6 Ejemplo práctico 2 – Smartphone en escena de crimen

Escenario: Un teléfono encontrado desbloqueado y encendido.

Procedimiento:

1. Activar modo avión para aislarlo.
2. Conectar a fuente de energía para evitar apagado.
3. Extraer datos mediante herramienta forense (Cellebrite, Oxygen).
4. Registrar toda manipulación en cadena de custodia.

Resultado: Evidencia digital admitida como prueba clave.

37.7 Herramientas y técnicas para preservar integridad

- **Hashing:** MD5, SHA-1, SHA-256 para verificación.
 - **Bloqueadores de escritura:** Tableau, WiebeTech.
 - **Contenedores forenses:** Formato E01 (EnCase), AFF (Advanced Forensic Format).
-

37.8 Buenas prácticas

- Trabajar siempre sobre copias y no sobre el original.
 - Usar doble verificación de hash por peritos distintos.
 - Mantener evidencias en sala segura con registro de acceso.
 - Documentar absolutamente cada transferencia de custodia.
-

37.9 Errores comunes

- Falta de firmas o datos en la cadena de custodia.
 - No calcular hash antes del análisis.
 - Almacenar evidencias junto a material no relacionado sin separación física.
-

37.10 Caso real documentado

En 2021, un caso de fraude informático en Córdoba fue desestimado porque la defensa probó que un HDD incautado no tenía registro de quién lo custodiaba durante 48 horas. Esa brecha en la cadena de custodia fue suficiente para invalidar toda la evidencia.

Resumen del capítulo: La cadena de custodia es la columna vertebral de cualquier investigación forense digital. Sin ella, la mejor evidencia técnica puede quedar fuera del proceso judicial.

Capítulo 38 – Redacción de Informes Periciales

38.1 Introducción

El **informe pericial** es el documento formal que plasma de manera clara, objetiva y técnica los hallazgos de una investigación forense digital. No es solo un resumen de lo encontrado: es una pieza clave en el proceso judicial, que debe poder ser entendida por jueces, fiscales, abogados y, en algunos casos, jurados sin conocimientos técnicos profundos. Un buen informe pericial combina **precisión técnica, claridad narrativa y respaldo documental**, asegurando que la evidencia presentada sea **comprendible, verificable y admisible**.

38.2 Objetivos de un informe pericial

1. **Documentar el trabajo realizado:** procedimientos, herramientas y técnicas.
2. **Presentar hallazgos:** describir evidencia relevante y su significado.

3. **Demostrar integridad:** mostrar que los datos no fueron alterados.
 4. **Respaldar conclusiones:** con análisis objetivos y reproducibles.
 5. **Servir como prueba legal:** cumpliendo con requisitos formales.
-

38.3 Estructura recomendada

Sección	Contenido
Portada	Título, número de caso, fecha, perito responsable.
Índice	Lista de secciones y anexos.
Resumen ejecutivo	Síntesis no técnica de los hallazgos.
Metodología	Procedimientos y herramientas utilizadas.
Resultados	Evidencia encontrada, con capturas y tablas.
Análisis	Interpretación técnica de los resultados.
Conclusiones	Resumen objetivo y claro de implicaciones.
Anexos	Hashes, logs, copias de actas, capturas completas.

38.4 Buenas prácticas de redacción

- Usar lenguaje claro y evitar jerga innecesaria.
 - Incluir diagramas y capturas de pantalla con etiquetas claras.
 - Citar versiones exactas de herramientas usadas.
 - Numerar páginas y secciones para referencia rápida.
 - Mantener tono objetivo y evitar opiniones personales no sustentadas.
-

38.5 Ejemplo de fragmento de informe

Caso: 2025-INV-042

Perito: Alejandro G. Vera

Fecha: 08/08/2025

Resumen Ejecutivo:

Durante el análisis del HDD incautado en el domicilio del sospechoso, se recuperaron 152 archivos eliminados que contenían información financiera de clientes. El hash SHA-256 calculado antes y después del análisis coincide, garantizando integridad.

Metodología:

1. Imagen forense realizada con FTK Imager v4.5.0.
2. Hash SHA-256: e3b0c44298fc1c149afbf4c8996fb...
3. Recuperación de archivos con Autopsy v4.19.

Resultados:

- Carpeta /Finanzas/ con 38 documentos Excel modificados el 05/08/2025.
- Evidencia de conexión a servidor IP 185.23.55.12.

38.6 Ejemplo práctico 1 – Caso de fraude interno

Escenario: Investigación en empresa por filtración de datos.

Estructura aplicada:

1. Portada con datos de caso.
2. Resumen ejecutivo para directivos.
3. Metodología paso a paso (herramientas, fechas, procedimientos).
4. Resultados con capturas de red y documentos recuperados.
5. Conclusiones vinculando hallazgos con el sospechoso.

Resultado: Informe aceptado en tribunal y comprensión total por parte de no técnicos.

38.7 Ejemplo práctico 2 – Caso con evidencia en la nube

Escenario: Chats sospechosos en plataforma de mensajería.

Procedimiento:

- Captura de mensajes con timestamps y metadatos.
- Exportación de logs en formato CSV.
- Verificación de integridad con hash SHA-1.
- Inclusión en informe con capturas comparativas.

Resultado: Evidencia digital presentada de forma visual y verificable.

38.8 Errores comunes

- No explicar términos técnicos, dejando al juez en la duda.
- Incluir capturas sin contexto ni descripción.
- No detallar la versión de herramientas utilizadas.
- Presentar conclusiones sin respaldo en evidencias concretas.

38.9 Caso real documentado

En 2020, un peritaje digital en una causa de fraude electrónico fue cuestionado porque el informe usaba abreviaturas técnicas sin explicación. La defensa argumentó que el juez no podía entender el alcance real de la prueba, y parte de la evidencia fue descartada por "falta de claridad".

Resumen del capítulo: Un informe pericial sólido es tanto un documento técnico como una herramienta de comunicación. Debe traducir el lenguaje de las máquinas al lenguaje del derecho, sin perder precisión ni objetividad.

Capítulo 39 – Presentación de Evidencias en Juicio

39.1 Introducción

Presentar evidencia digital en un juicio no es simplemente mostrar un archivo o un log: es **convencer al tribunal** de que esa prueba es auténtica, íntegra, relevante y obtenida de forma legal. En la sala, el perito forense digital deja de ser solo un técnico y se convierte en un **comunicador**, capaz de traducir conceptos complejos en explicaciones claras para jueces, fiscales, defensores e incluso jurados sin formación tecnológica. El objetivo es que la evidencia **sea admitida, comprendida y valorada** adecuadamente.

39.2 Requisitos para la admisibilidad

Criterio	Descripción	Ejemplo
Autenticidad	La evidencia es lo que se afirma que es.	Hash del archivo coincide con el capturado en incautación.
Integridad	No ha sido modificada desde su recolección.	Registro continuo de cadena de custodia.
Relevancia	Está relacionada con los hechos investigados.	Log que muestra acceso del sospechoso a un sistema.
Obtención legal	Recolectada cumpliendo las normas procesales.	Orden judicial para incautación de servidor.

39.3 Rol del perito en juicio

1. Exponer metodología

- Explicar herramientas, técnicas y pasos seguidos.

2. Mostrar evidencia

- Presentar capturas, gráficos y tablas de forma clara.

3. Responder preguntas

- Tanto del juez como de abogados de ambas partes.

4. Mantener objetividad

- No tomar partido, solo explicar hallazgos.

39.4 Preparación previa al juicio

- Revisar informe pericial completo.

- Practicar explicación en lenguaje no técnico.
 - Preparar **material visual**: diagramas, líneas de tiempo, flujos de transacciones.
 - Anticipar posibles objeciones y preguntas de la defensa.
-

39.5 Técnicas de exposición efectiva

- **Analogías**: comparar un hash con una huella digital.
 - **Visualización**: usar gráficos de redes o cronologías para mostrar secuencias.
 - **Segmentación**: dividir explicación en pasos simples.
 - **Repetición clave**: reforzar puntos esenciales como integridad y autenticidad.
-

39.6 Ejemplo práctico 1 – Caso de intrusión remota

Escenario: El perito debe demostrar que un atacante accedió a un servidor corporativo.

Presentación:

1. Mostrar línea de tiempo con fechas y horas de accesos sospechosos.
2. Presentar log original con hash calculado.
3. Explicar correlación con dirección IP asignada al sospechoso.
4. Concluir que hubo acceso no autorizado.

Impacto: El juez entiende la secuencia y acepta la prueba como legítima.

39.7 Ejemplo práctico 2 – Caso de fraude con criptomonedas

Escenario: La fiscalía necesita mostrar cómo se transfirieron fondos robados.

Procedimiento en sala:

- Proyectar gráfico de blockchain mostrando las transacciones.
- Marcar direcciones involucradas y montos.
- Explicar uso de exchange y vinculación a identidad real.

Resultado: Prueba clara y visual que sostiene la acusación.

39.8 Buenas prácticas

- Mantener calma ante preguntas de la defensa.
 - No usar lenguaje técnico innecesario.
 - Llevar copias impresas y digitales del informe.
 - Referirse siempre a la evidencia como "presunta" hasta veredicto final.
-

39.9 Errores comunes

- Contradecir el propio informe por falta de preparación.

- Usar capturas poco legibles o confusas.
 - Responder con opiniones sin sustento técnico.
-

39.10 Caso real documentado

En 2019, un caso de abuso de sistemas informáticos en Buenos Aires tuvo un vuelco cuando el perito, en su exposición, mostró una infografía simple y clara que permitió al tribunal comprender un complejo flujo de datos. La defensa no pudo refutar la presentación, y la prueba fue decisiva para la condena.

Resumen del capítulo: Presentar evidencia digital en juicio requiere habilidades técnicas y comunicativas. El perito debe ser un puente entre la tecnología y el derecho, asegurando que la evidencia sea comprensible, relevante y aceptada.

Capítulo 40 – Buenas Prácticas, Retos y Futuro de la Informática Forense

40.1 Introducción

La informática forense es un campo en constante evolución, impulsado por la innovación tecnológica y por la creatividad —a veces devastadora— de quienes cometen delitos informáticos. Cerrar este libro implica dejar al lector con una visión integral: **qué hacer bien hoy, qué problemas enfrentará mañana y cómo adaptarse**. El éxito del perito digital no dependerá solo de su dominio técnico, sino de su capacidad de actualización continua, ética profesional y visión global.

40.2 Buenas prácticas esenciales

Área	Buena práctica	Impacto
Recolección de evidencia	Usar bloqueadores de escritura y calcular hash antes y después.	Garantiza integridad y admisibilidad.
Documentación	Registrar cada paso en cadena de custodia.	Asegura trazabilidad.
Análisis	Trabajar siempre sobre copias forenses.	Previene daños al original.
Comunicación	Usar lenguaje claro en informes y juicios.	Mejora comprensión del tribunal.
Actualización profesional	Capacitación continua en nuevas amenazas y herramientas.	Mantiene relevancia profesional.

40.3 Retos actuales de la informática forense

1. Cifrado generalizado

- Dispositivos móviles y servicios de mensajería como WhatsApp y Signal usan cifrado de extremo a extremo, dificultando el acceso legal a datos.

2. Evidencia en la nube

- La dispersión de datos en múltiples servidores y países complica la preservación.

3. Anonimato reforzado

- Redes como Tor y criptomonedas centradas en privacidad (Monero) elevan el desafío.

4. Volumen y variedad de datos

- Petabytes de logs, imágenes y registros requieren técnicas avanzadas de filtrado y análisis.

5. Legislaciones divergentes

- Diferencias legales entre países retrasan o imposibilitan investigaciones.
-

40.4 Futuro de la informática forense

El horizonte apunta a una **forensia digital más automatizada, colaborativa e integrada con inteligencia artificial**. Algunas tendencias clave:

- **Análisis asistido por IA**: detección automática de patrones sospechosos en grandes volúmenes de datos.
 - **Herramientas en la nube forense**: laboratorios virtuales para análisis distribuidos.
 - **Forensia de IoT**: técnicas especializadas para dispositivos conectados en hogares y ciudades inteligentes.
 - **Blockchain forense**: métodos para auditar transacciones y contratos inteligentes.
 - **Colaboración internacional en tiempo real**: redes 24/7 para respuesta inmediata a incidentes globales.
-

40.5 Ejemplo práctico – Integración de IA en un laboratorio forense

Escenario: Un laboratorio recibe 50 TB de datos de servidores corporativos.

Solución futura:

1. Herramienta de IA analiza logs y genera alertas de patrones anómalos.
2. Sistema clasifica correos por relevancia usando PLN (procesamiento de lenguaje natural).
3. Perito revisa solo los elementos marcados como de alto interés.

Beneficio: Ahorro del 80% del tiempo de revisión manual.

40.6 Recomendaciones para el perito del futuro

- Aprender **programación y scripting** para automatizar tareas repetitivas.
- Dominar **herramientas de análisis en nube y contenedores**.

- Formarse en **ciberinteligencia** para anticipar amenazas.
 - Mantener vínculos con redes internacionales de respuesta a incidentes.
 - Desarrollar habilidades **de comunicación y docencia** para transmitir resultados.
-

40.7 Reflexión final

La informática forense no es solo un conjunto de técnicas; es un **compromiso ético con la verdad y la justicia**. Cada dispositivo, cada log y cada byte analizado representan piezas de un rompecabezas que puede cambiar el destino de una persona o una organización. La velocidad del cambio tecnológico significa que los conocimientos de hoy pueden volverse obsoletos mañana. El perito que triunfará será el que nunca deje de aprender, que combine rigor técnico con claridad comunicativa, y que entienda que la tecnología es solo una herramienta: lo que realmente marca la diferencia es la mente crítica que la utiliza.

Cierre del libro: Con esta obra, "La Biblia de la Informática Forense", no solo has recorrido técnicas y metodologías, sino que te has adentrado en el marco legal, las estrategias de presentación y el horizonte futuro de la profesión. Ahora, la responsabilidad recae en ti: aplicar estos conocimientos con integridad, perseverancia y adaptabilidad.
