Universidad Distrital FJDC. Facultad de Ingeniería. maestría en Ciencias de la información y las comunicaciones. Asignatura REDES. Ing. Octavio José Salcedo Parra. octavionetworking@gmail.com. Taller No. 2. octubre 24/2025. Entrega: noviembre 7 de 2025.

Objetivo General:

Analizar las debilidades inherentes a los protocolos de red (arquitectura TCP/IP) y aplicaciones Internet más utilizados, identificar posibles amenazas derivadas del diseño original o mal uso, y aplicar técnicas de defensa activa, combinadas con herramientas de inteligencia artificial para la detección automática de anomalías y amenazas.

Objetivos Específicos:

1. Identificar vulnerabilidades en protocolos como IPv4, TCP, UDP, ARP, ICMP, BGP y RIPv2.
2. Analizar cómo ciertas aplicaciones pueden ser explotadas por atacantes.
3. Estudiar casos reales de ataques basados en estas debilidades.
4. Aplicar buenas prácticas de seguridad en redes.
5. Utilizar herramientas de inteligencia artificial para el análisis predictivo y detección de anomalías en tráfico de red.
6. Aplicar direccionamiento IPv4 y encaminamiento TCP/IP.
7. Comprender profundamente la arquitectura de Internet

Nota 1: Por favor, Responda las preguntas, no colocar marco teórico o referencial en las respuestas, solamente responda las preguntas de acuerdo a lo que se pregunta, no es necesario colocar información adicional.

Cuestionario

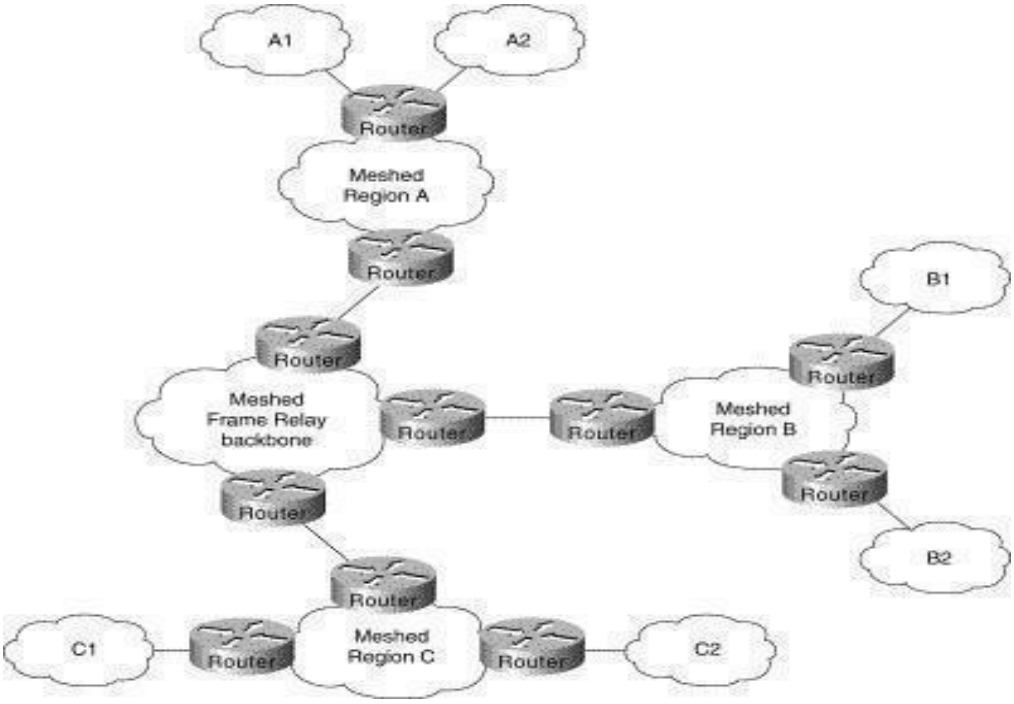
Parte I. Direccionamiento IPv4, utilizando VLSM

1) Direccione óptimamente y diseñe (puede utilizar Cisco Config Maker, Cisco Packet Tracer, GNS-3) con el protocolo IP los siguientes diseños. Por cada LAN presente suponga que posee 11400 host, 7500 host, 3630 host, 2326 host, 1520

host, 678 host respectivamente. No olvide que todas las direcciones a utilizar son direcciones públicas. Es necesario que Usted solamente utilice una (1) sola dirección pública de red para el diseño. Analice el diseño de las redes IP a utilizar antes de configurarlas y su impacto en el desempeño el respectivo escenario.

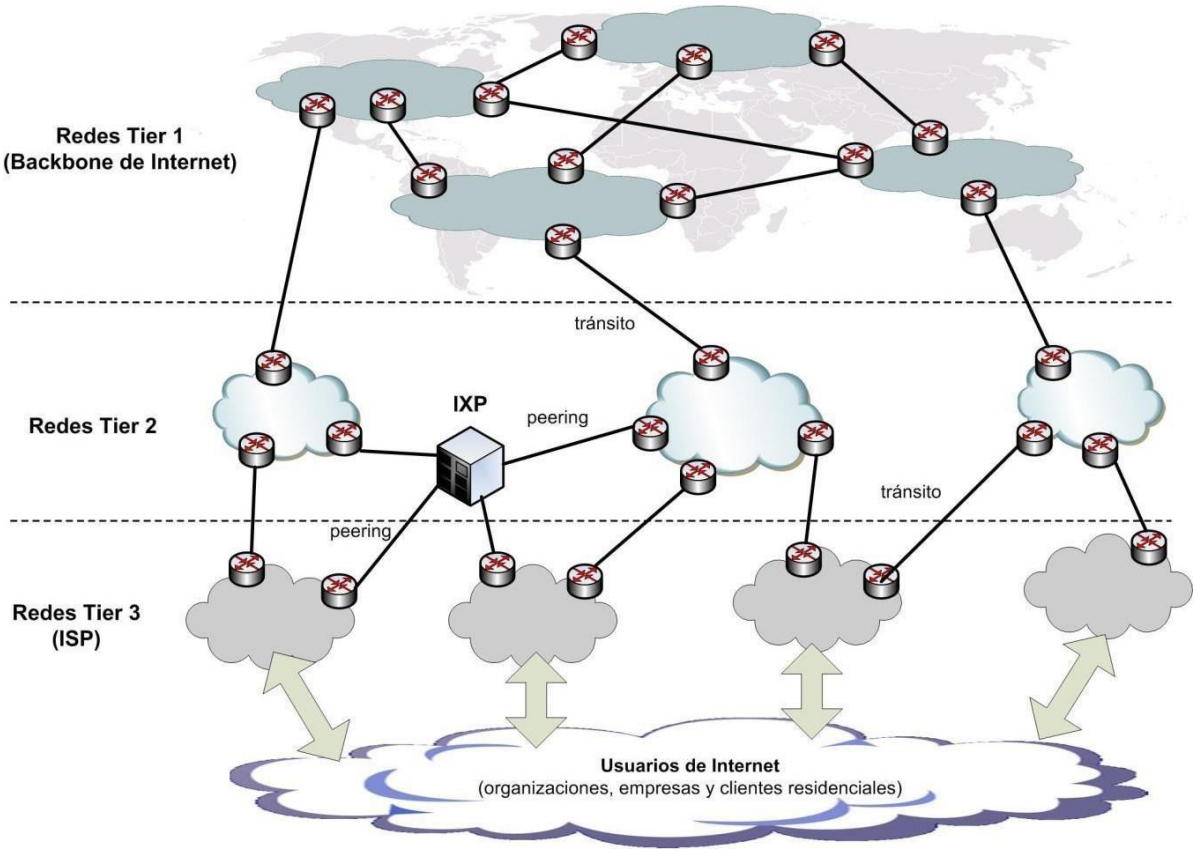
Nota: La afirmación direccione óptimamente hace precisión a la asignación precisa de la máscara de acuerdo al número de host que se necesitan para las redes LAN respectivas. De igual forma la asignación es casi precisa, para cada asignación de máscaras NO debe haber un desperdicio de direcciones públicas más 7 direcciones por sub(red).

Puede utilizar el siguiente diseño de referencia:



Parte II. Arquitectura de Internet

Diseñe y configure un pequeño modelo “prototipo del backbone de Internet”. Ver gráfica. Hay que configurar protocolos IGP (como RIPv2, OSPF) y EGP (BGP) conel Cisco Packet Tracer prototipo del backbone de Internet. Considerando el protocolo ICMP con las trazas para verificar el nivel o clasificación de los ISP (Tiers)



Parte III. Análisis de Vulnerabilidades en Protocolos y Aplicaciones de Red con Enfoque en Ciberseguridad e Inteligencia Artificial.

En los siguientes **Casos** hay que instalar una red LAN o Wi-Fi, hacer el respectivo direccionamiento, de igual forma hay configurar servidores de acuerdo con el caso y verificar la conectividad entre las estaciones y/o servidores para después montar (configurar) el caso en cuestión.

En los informes relacionados con la respuesta de cada **Caso** hay presentar el procedimiento paso a paso, respaldado con capturas de pantalla, detallando como fue o se configuro (implemento) el **Caso** respectivo, con la(s) herramienta(s) utilizada(s) para la implementación y respectiva defensa

Caso 1: Uso de ICMP para Exfiltración de Datos

Descripción: Un malware utiliza mensajes ICMP para enviar datos sensibles fuera de la red sin ser detectado por firewalls tradicionales.

Herramienta IA sugerida: Darktrace Network Detection & Response con detección de anomalías en ICMP , Splunk + modelo de IA para analizar payloads ICMP

Técnica de defensa: Bloquear tipos ICMP innecesarios , Analizar payloads de ICMP en capa profunda , Aplicar reglas estrictas en firewalls

Caso 2: Ataque de DDoS Reflejado con UDP

Descripción: Se aprovecha la naturaleza sin conexión de UDP para generar tráfico

masivo desde servicios como DNS o NTP.
Herramienta IA sugerida: AI-based traffic analysis tools (como Vectra o MixMode), Machine learning para identificar picos de tráfico UDP
Técnica de defensa: Limitar respuesta de servicios públicos a redes internas , Usar rate limiting , Filtrar puertos UDP innecesarios.

Caso 3: Interceptación de Comunicaciones en Servidores de Mensajería Instantánea

Descripción: Un atacante intercepta comunicaciones en entornos empresariales usando MITM o explotando configuraciones incorrectas en servidores XMPP u otros.
Herramienta IA sugerida: User and Entity Behavior Analytics (UEBA) para detectar comunicación inusual , Análisis de metadatos con IA para detectar fuentes sospechosas
Técnica de defensa: Usar cifrado de extremo a extremo (E2EE) , Validar certificados TLS , Monitorear conexiones inusuales

Caso 4: Interceptación de Comunicaciones en Servidores de Mensajería Instantánea

Descripción: Un atacante intercepta comunicaciones en entornos empresariales usando MITM o explotando configuraciones incorrectas en servidores XMPP u otros.
Herramienta IA sugerida: User and Entity Behavior Analytics (UEBA) para detectar comunicación inusual , Análisis de metadatos con IA para detectar fuentes sospechosas
Técnica de defensa: Usar cifrado de extremo a extremo (E2EE), validar certificados TLS, Monitorear conexiones inusuales

Caso 5: Detectar ARP Spoofing con Wireshark + IA

Capturar tráfico con Wireshark.
Identificar múltiples respuestas ARP para la misma IP Importar datos a una herramienta de IA (ej: Python + Scikit-learn) para clasificación automática

Caso 6: Análisis de Paquetes ICMP con IA

Generar tráfico ICMP normal y malicioso . Usar Wireshark + Python script para extraer características. Entrenar modelo de IA para clasificar tráfico benigno vs. Malicioso

Caso 7: Escaneo de Puertos TCP y UDP + Mitigación

Usar Nmap para escaneo SYN y UDP scan
Analizar respuesta TCP/UDP en Wireshark
Aplicar técnicas de firewalling y rate limiting

Caso 8: Análisis de Tráfico en Servidor de Mensajería (XMPP)
Usar Openfire o ejabberd
Capturar tráfico con Wireshark
Detectar intentos de conexión sin autenticación
Usar IA para identificar comportamientos anómalos

Caso 9: Internet / Infraestructura: scanning masivo y fingerprinting (p.ej. Shodan-like)

- **Objetivo**: detectar barridos de puertos y fingerprinting que preceden intrusiones.
- **Qué observar**: múltiples IPs/protocolos accediendo a servicios no comunes, patrones de conexión secuenciales a puertos.
- **Tareas**:
    1. Correlacionar logs Zeek con Suricata para identificar IPs que escanean.
    2. Crear listas de bloqueo temporales y reglas Wazuh para alertar sobre IPs con comportamiento de scanner.
    3. Preparar checklist de exposure: servicios públicos sin autenticar, versiones viejas.
- **Entrega**: lista de IOCs (IPs), reglas y plan de endurecimiento de servicios.

Caso 10 — Correlación y respuesta con IA: modelo de anomalías en tráfico + enriquecimiento automatizado

- **Objetivo**: usar IA para priorizar alarmas y detectar anomalías que firmas no cubren.
- **Qué observar/usar**: features derivados de Zeek/Suricata/Wazuh (p. ej. conexiones por minuto, bytes por sesión, flags TCP, ratio request/response).
- **Tareas**:
    1. Extraer features temporales (ventana 1–5 min).
    2. Entrenar un prototipo de detector de anomalías (IsolationForest o modelos de

PyOD) para destacar sesiones anómalas. digitalocean.com+1

3. Integrar output del modelo como score en el SIEM y usar un LLM (con prompts controlados) para sumarizar alerts y sugerir acciones de primer nivel (triage). **Nota:** usar LLMs solo para asistencia; las acciones de bloqueo deben requerir revisión humana.

- **Entrega**: notebook con pipeline de features → modelo → scoring; ejemplos de alertas priorizadas y prompt templates para el LLM.

Parte IV. Programación. Utilizando técnicas, aplicaciones e Inteligencia Artificial.

Crear una app, programa o bot para rastrear (ubicar) números telefónicos **de servicios de mensajería instantánea como** WhatsApp**,** Telegram**, o directamente la** ubicación geográfica de un celular.

Explicación, requerimientos y aspectos técnicos para desarrollar un sistema (app, programa o bot) que:

4.1 Explicación

- **Rastree la ubicación geográfica de un número telefónico** asociado a:
  - WhatsApp (limitado, indirecto)
  - Telegram (con bots)
  - Celular directamente (usando GPS)

Muestre en tiempo real o por intervalos la ubicación del usuario.

4.1. Requisitos Funcionales

Requisito	Descripción
Registro por número telefónico	El sistema debe registrar y vincular dispositivos a través del número telefónico.
Solicitud y captura de ubicación	Permitir al usuario objetivo enviar su ubicación actual o automática.
Actualización periódica	Capturar la ubicación del dispositivo cada intervalo definido (ej. cada 10 minutos).

Visualización en mapa	Mostrar ubicación en un mapa integrado (Google Maps, OpenStreetMap, etc.).
Alertas (opcional)	Alertar si el dispositivo entra o sale de una zona geográfica (geofencing).
Historial de ubicaciones	Guardar y consultar ubicaciones pasadas por fecha/hora.
Acceso seguro	Autenticación del usuario

4.3 Requisitos Técnicos

4.3.1. Aplicación móvil (necesaria para ubicación directa)

- **Android/iOS app** instalada en el dispositivo objetivo.
- Solicitud de permisos de **GPS/localización**.
- Envío periódico de coordenadas al servidor (backend).
- Vinculación con número telefónico (SMS o código de verificación).

4.3.2. Bot en Telegram (opcional)

- El bot puede solicitar ubicación manual mediante botones.
- **No puede rastrear automáticamente** sin que el usuario lo envíe.
- Requiere que el usuario inicie el chat y lo autorice.

4.3.3. Backend (servidor)

- Lenguaje: Python, Node.js, PHP o similar.
- API REST para enviar/recibir ubicación.
- Base de datos: PostgreSQL, MongoDB o Firebase.
- Sistema de autenticación (por número y token).

4.3.4 Mapa y visualización

- API de Google Maps, Mapbox o Leaflet.js para mostrar las ubicaciones.
- Soporte para geocodificación inversa (coordenadas → dirección).

4.4 Características opcionales avanzadas

4.4.1 Geofencing personalizado (notificaciones por zona)
4.4.2 Ahorro de batería mediante activación programada
4.4.3 Múltiples dispositivos vinculados (familia, empleados)
4.4.4 Interfaz web o panel administrativo
4.4.5 Estadísticas de movimiento o recorridos

Nota Importante: El taller No. 2, a desarrollar (resolver), no se está pidiendo (solicitando) ingeniera(o), que vulnere, ataque o hackea ninguno de los escenarios domésticos o empresariales a los cuales Usted pueda tener acceso. El ejercicio es carácter académico. Los participantes deben actuar como hackers éticos, sin causar daño intencional a sistemas (redes de comunicaciones) y personas. No se permite el uso de técnicas que comprometan la integridad de sistemas (redes de comunicaciones y servidores TCP/IP) reales sin permiso o autorización expresa de los propietarios de los escenarios en cuestión.