

TU PROPIO SOC HÍBRIDO



12 paneles defensivos que protegen sistemas.
El ecosistema defensivo que tú también puedes desplegar, adaptar y mejorar.



UNAI RUBIO | Hacker Ético

1. MarineTraffic – Seguimiento Marítimo Global

Definición Técnica Ampliada

MarineTraffic es un sistema OSINT que monitoriza el tráfico marítimo global en tiempo real, utilizando el AIS (Automatic Identification System). El AIS transmite información dinámica (posición, velocidad, rumbo), estática (nombre, IMO, MMSI, dimensiones) y específica del viaje (destino, ETA) desde los buques hacia estaciones receptoras terrestres y satelitales. Estos datos, codificados en sentencias NMEA, se procesan y visualizan en mapas interactivos, permitiendo el análisis histórico y en tiempo real de rutas, puertos y patrones de navegación.

Usos en Ciberseguridad

- **Protección de infraestructuras críticas:** Monitorización de buques cercanos a puertos, refinerías, cables submarinos o plataformas offshore.
- **Análisis de amenazas híbridas:** Correlación entre eventos físicos (movimientos no autorizados) y ciberataques a sistemas marítimos.
- **Investigación OSINT:** Rastrear embarcaciones vinculadas a actividades ilícitas, piratería o contrabando.
- **Gestión de crisis:** Apoyo a la toma de decisiones ante incidentes de seguridad marítima o sabotaje.

Técnicas Avanzadas

- Filtros por bandera, tipo de buque, zonas restringidas y horarios inusuales.
- Exportación de datos AIS para análisis en SIEM o correlación con logs de acceso físico.
- Cruce de información con fuentes como Flightradar24 y Google Earth para obtener contexto geoespacial completo.
- Detección de spoofing AIS (manipulación de datos) mediante análisis de inconsistencias en rutas o velocidades.

Mejores Implementaciones

- Integración de MarineTraffic con sistemas de alerta temprana y plataformas de threat intelligence.
- Uso de APIs para automatizar la recolección y análisis de datos marítimos.
- Implementación de dashboards personalizados en SOC's marítimos.

Recursos y Repositorios

- Plataforma: <https://www.marinetraffic.com>
- API: <https://www.marinetraffic.com/en/ais-api-services>
- Proyecto Open Source: <https://www.openseamap.org>
- Ejemplo de decodificación NMEA en Python:

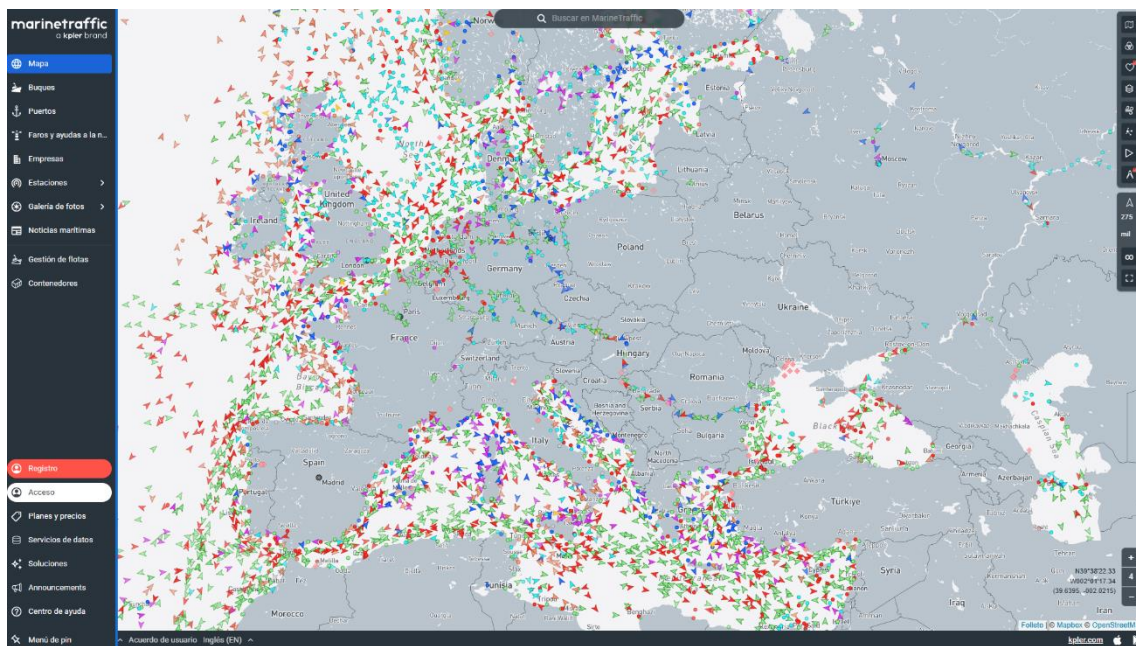
python

```
import pynmea2
```

```
msg =
```

```
pynmea2.parse('!AIVDM,1,1,,A,15Mwdc001oG?tTPK>R6g0?vN0TKH,0*3C')
```

```
print(msg)
```



2. Flightradar24 – Trazabilidad Aérea

Definición Técnica Ampliada

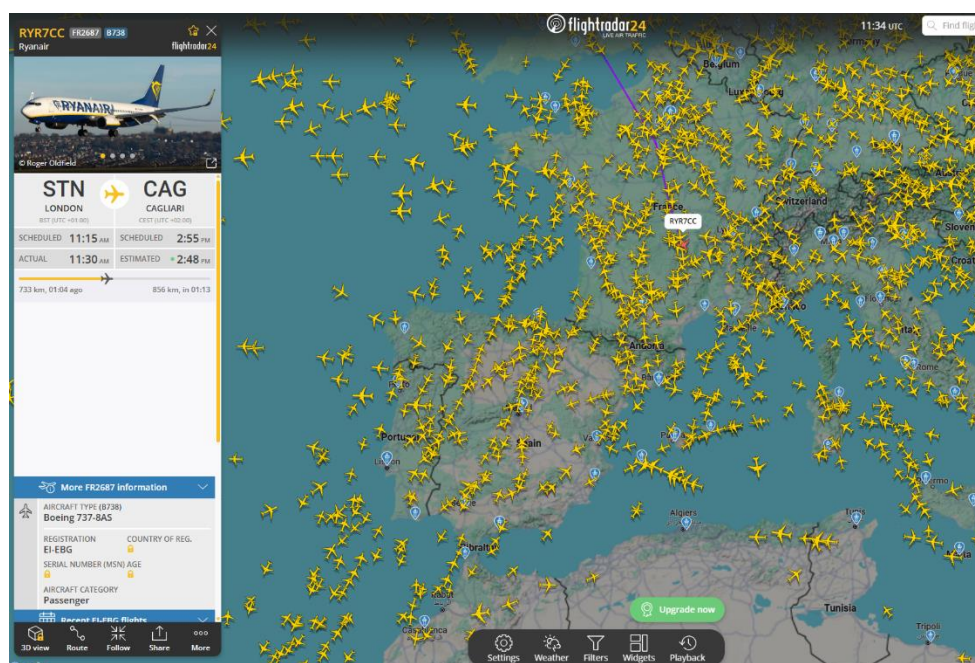
Flightradar24 monitoriza vuelos comerciales, privados y helicópteros en tiempo real usando una red global de receptores ADS-B (Automatic Dependent Surveillance–Broadcast). Los transpondedores ADS-B transmiten información de posición GPS, altitud, velocidad, rumbo y número de vuelo a 1090 MHz, que es procesada y visualizada en mapas interactivos.

Usos en Ciberseguridad

- **Protección de instalaciones:** Detección de vuelos no declarados o anómalos cerca de infraestructuras críticas.
- **Análisis forense:** Investigación de incidentes aéreos relacionados con amenazas físicas o cibernéticas.
- **Correlación OSINT:** Cruzar rutas aéreas con eventos de seguridad o movimientos sospechosos detectados por otras fuentes.

Técnicas Avanzadas

- Configuración de alertas para vuelos que sobrevuelen áreas sensibles.
- Exportación y análisis de rutas históricas para identificar patrones inusuales.
- Correlación de vuelos con eventos geopolíticos o ciberataques sincronizados.



Mejores Implementaciones

- Integración de Flightradar24 con sistemas de gestión de incidentes físicos y cibernéticos.
- Uso de su API para alimentar herramientas SIEM con datos de vuelos sospechosos.

Recursos y Repositorios

- Plataforma: <https://www.flightradar24.com>
- API: <https://www.flightradar24.com/how-it-works>
- Receptor ADS-B DIY: <https://www.rtl-sdr.com>
- Script básico de decodificación ADS-B en Python:

```
python
import pyModeS
msg = '8D40621D58C382D690C8AC2863A7'
print(pyModeS.adsb.typecode(msg))
```

3. Google Earth – OSINT Visual y Geolocalización

Definición Técnica Ampliada

Google Earth es una plataforma de visualización geoespacial que permite explorar imágenes satelitales, mapas 3D y datos geográficos. Permite importar datos en KML/KMZ, visualizar rutas, perímetros y realizar análisis temporal de cambios en infraestructuras. Puede integrarse con sensores y servicios externos mediante APIs o archivos de red.

Usos en Ciberseguridad

- **Reconocimiento físico:** Identificación de perímetros, accesos y vulnerabilidades en instalaciones críticas.
- **Análisis forense:** Comparación de imágenes históricas para detectar cambios sospechosos.
- **Soporte a operaciones:** Planificación de rutas de intervención o evacuación.

Técnicas Avanzadas

- Uso de archivos KML para marcar activos críticos y rutas de acceso.
- Automatización de actualizaciones periódicas de datos mediante enlaces de red.
- Integración con sensores (SOS) para visualizar información en tiempo real.

Mejores Implementaciones

- Visualización de activos junto con datos de amenazas (vuelos, buques, clima).
- Uso de Google Earth Engine para análisis masivo de imágenes.
- Precaución con vulnerabilidades en archivos KMZ; validación de fuentes antes de abrir.

Recursos y Repositorios

- Plataforma: <https://earth.google.com>
- Earth Engine: <https://earthengine.google.com>
- Librería Python KML: <https://simplekml.readthedocs.io>
- Ejemplo de generación de KML:

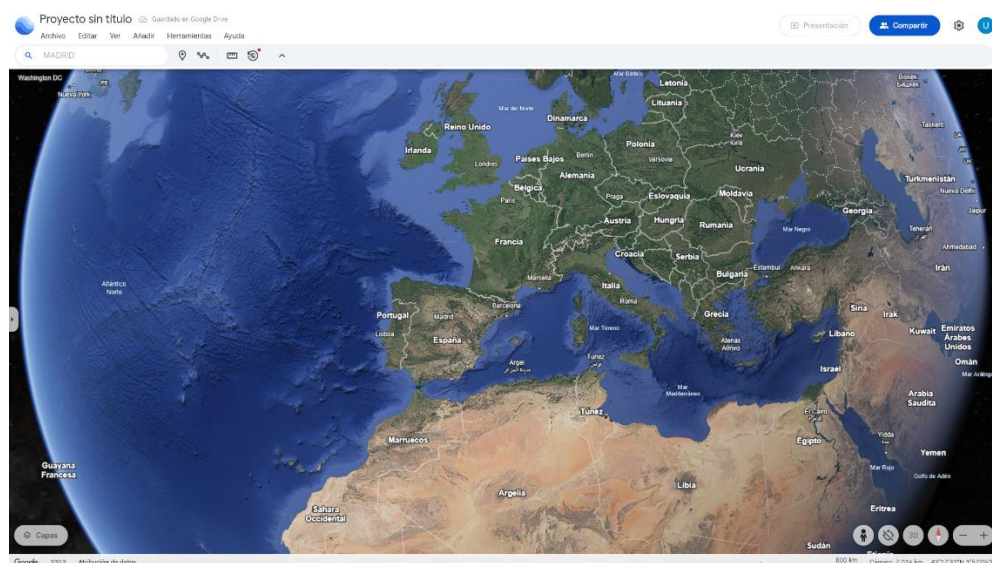
python

```
import simplekml
```

```
kml = simplekml.Kml()
```

```
kml.newpoint(name="SOC", coords=[(-3.7038, 40.4168)])
```

```
kml.save("soc_location.kml")
```



4. Windy – Meteorología Activa y Alertas Críticas

Definición Técnica Ampliada

Windy es una plataforma meteorológica avanzada que visualiza datos de viento, precipitaciones, presión, temperatura, calidad del aire y alertas naturales en tiempo real, usando modelos predictivos como ECMWF, GFS e ICON. Ofrece mapas interactivos, animaciones y capas configurables para seguimiento de tormentas, huracanes y fenómenos extremos.

Usos en Ciberseguridad

- **Continuidad operativa:** Anticipación de eventos climáticos que puedan afectar datacenters, enlaces WAN, infraestructuras OT.
- **Gestión de crisis:** Apoyo a planes de contingencia ante desastres naturales.
- **Correlación de incidentes:** Relación entre eventos meteorológicos y caídas de sistemas o picos de tráfico.

Técnicas Avanzadas

- Configuración de alertas personalizadas por email o dashboard.
- Selección de modelos meteorológicos para máxima precisión.
- Visualización de capas específicas (viento, lluvia, olas, calidad del aire) para análisis sectorial.

Mejores Implementaciones

- Integración de Windy en paneles de monitorización de SOC.
- Uso de la API para alimentar sistemas de alerta temprana.

Recursos y Repositorios

- Plataforma: <https://www.windy.com>
- API: <https://api.windy.com>
- Ejemplo de consulta meteorológica con Python:

```
python
import requests
response = requests.get("https://api.open-
meteo.com/v1/forecast?latitude=40.4168&longitude=-
3.7038&hourly=temperature_2m")
print(response.json())
```

5. Netdata (Ubuntu) – Infraestructura Viva

Definición Técnica Ampliada

Netdata es una herramienta de monitorización en tiempo real para sistemas Linux y Windows. Proporciona métricas detalladas de CPU, RAM, disco, red, procesos, servicios y aplicaciones, con visualizaciones interactivas y alertas personalizables. Permite la detección temprana de anomalías en la infraestructura del SOC.

Usos en Ciberseguridad

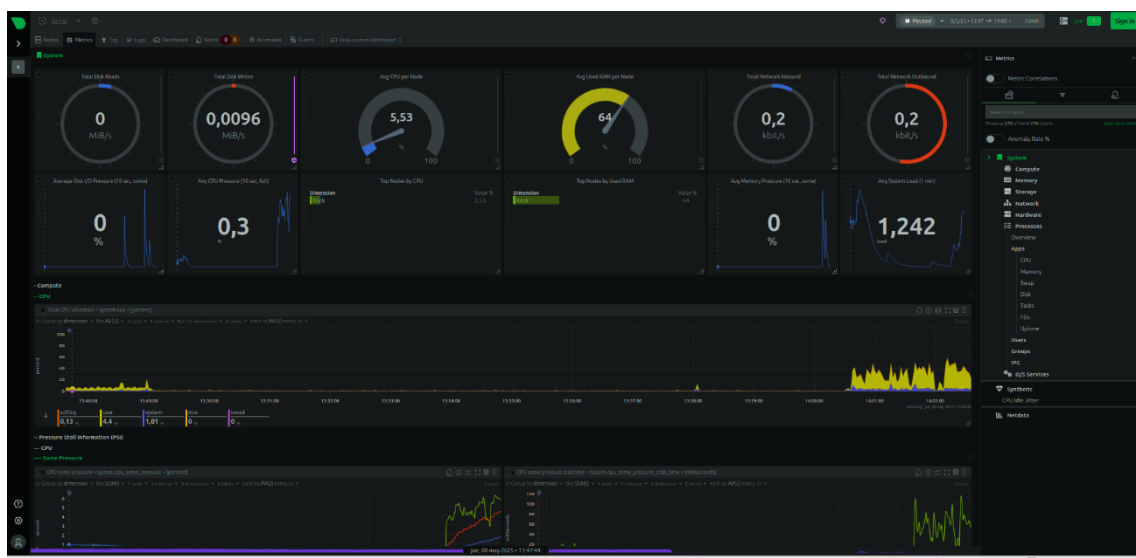
- **Detección de incidentes:** Identificación de procesos sospechosos, picos de uso de recursos o caídas de servicios.
- **Análisis forense:** Revisión de logs y métricas tras incidentes.
- **Supervisión continua:** Monitorización de nodos críticos (SIEM, firewalls, honeypots).

Técnicas Avanzadas

- Configuración de alertas por umbral y notificaciones en tiempo real.
- Integración con Prometheus/Grafana para análisis histórico.
- Uso de health checks y dashboards personalizados.

Mejores Implementaciones

- Instalación en todos los nodos críticos del SOC.
- Integración con sistemas de ticketing y respuesta automatizada.



Recursos y Repositorios

- Plataforma: <https://www.netdata.cloud>
- Repositorio: <https://github.com/netdata/netdata>
- Instalación rápida:

bash

bash <(curl -Ss <https://my-netdata.io/kickstart.sh>)

6. SpiderFoot (Kali Linux) – Escaneo OSINT Automatizado

Definición Técnica Ampliada

SpiderFoot es una plataforma OSINT automatizada que ejecuta módulos para recolectar información sobre dominios, IPs, emails, leaks, DNS, redes sociales y más. Funciona en modo CLI o web, y permite el reconocimiento pasivo sin alertar a los objetivos.

Usos en Ciberseguridad

- **Reconocimiento externo:** Identificación de la superficie de ataque expuesta de una organización.
- **Detección de leaks:** Búsqueda de credenciales, emails o datos sensibles filtrados.
- **Apoyo a investigaciones:** Análisis de amenazas externas y campañas dirigidas.

Técnicas Avanzadas

- Configuración de escaneos modulares y automatizados.
- Exportación de resultados en múltiples formatos (CSV, JSON, HTML).
- Integración con otras herramientas OSINT y SIEM.

Mejores Implementaciones

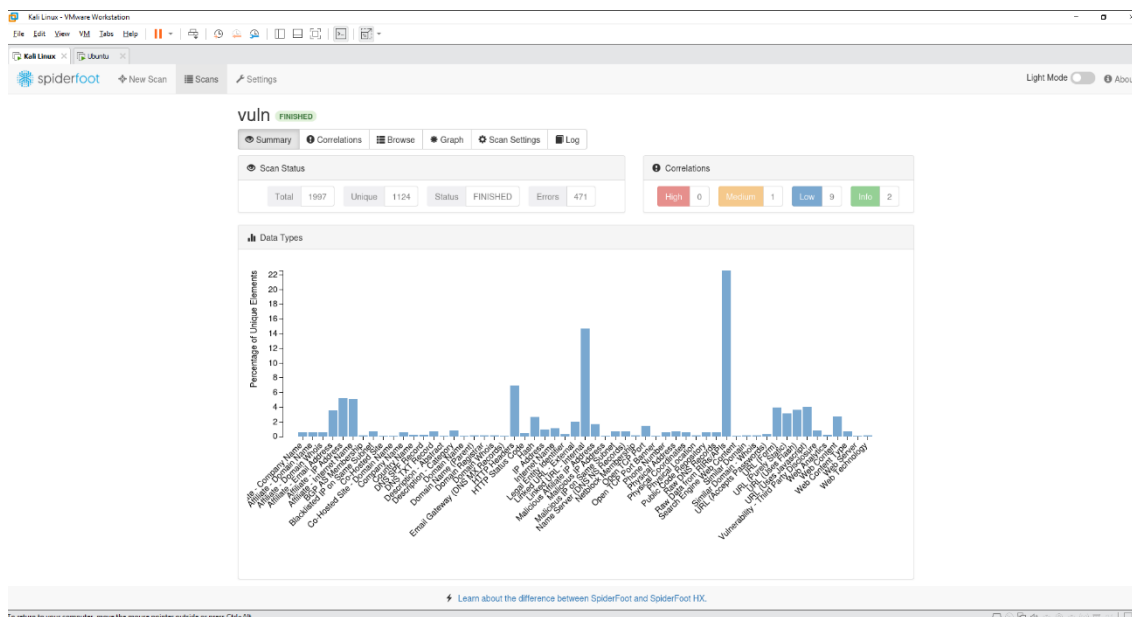
- Ejecución periódica de escaneos sobre activos propios.
- Uso combinado con análisis manual para contexto profundo.

Recursos y Repositorios

- Plataforma: <https://www.spiderfoot.net>
- Repositorio: <https://github.com/smicallef/spiderfoot>
- Ejemplo de escaneo CLI:

bash

```
python3 sf.py -s example.com -o results.html
```



7. Wireshark – Captura y Análisis de Red

Definición Técnica Ampliada

Wireshark es un analizador de protocolos de red que permite capturar, filtrar y analizar paquetes en tiempo real o desde archivos pcap. Soporta cientos de protocolos y ofrece potentes filtros de visualización, decodificación y reconstrucción de sesiones.

Usos en Ciberseguridad

- **Detección de amenazas:** Identificación de tráfico malicioso, C2, malware, exfiltración de datos.
- **Análisis forense:** Investigación de incidentes, reconstrucción de ataques y extracción de evidencias.

- **Auditoría de red:** Revisión de protocolos inseguros, credenciales en claro y configuraciones erróneas.

Técnicas Avanzadas

- Uso de filtros avanzados (por IP, puerto, protocolo, string).
- Decodificación de protocolos cifrados (SSL/TLS) con claves privadas.
- Automatización de análisis con tshark (CLI).

Mejores Implementaciones

- Integración con sistemas de captura automatizada.
- Uso combinado con reglas Sigma y análisis de logs.

Recursos y Repositorios

- Plataforma: <https://www.wireshark.org>
- Repositorio: <https://gitlab.com/wireshark/wireshark>
- Filtro básico para HTTP:

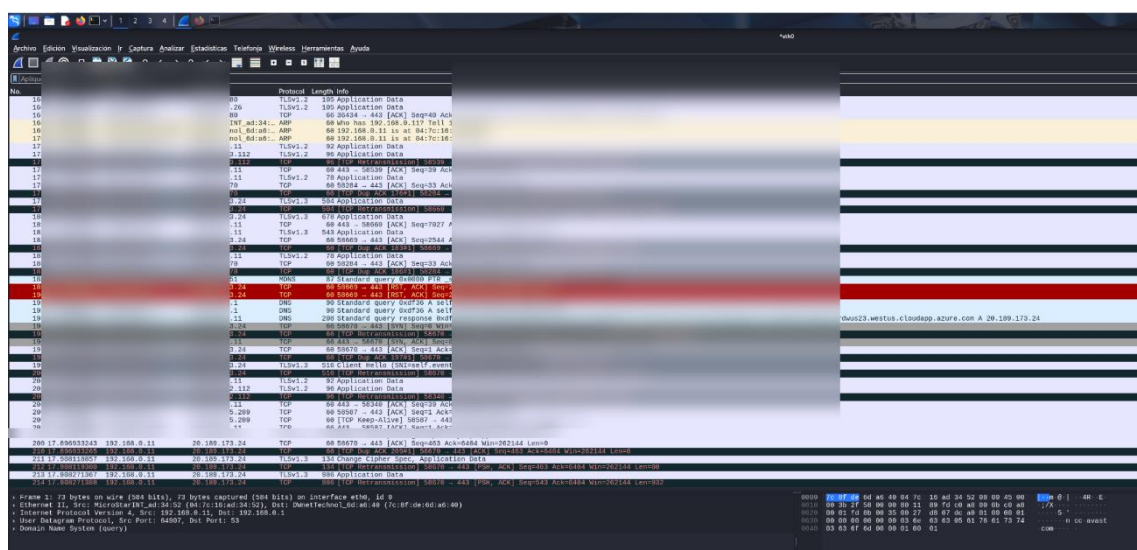
text

http.request

- Captura CLI:

bash

tshark -i eth0 -f "tcp port 80"



8. Visual Studio Code – Cheatsheet Nmap Activa

Definición Técnica Ampliada

Visual Studio Code es un editor de código multiplataforma, ideal para la gestión de scripts, cheatsheets y automatización de tareas técnicas. Permite la edición y ejecución de scripts Nmap NSE, integración con terminal y repositorios Git, y personalización mediante extensiones.

Usos en Ciberseguridad

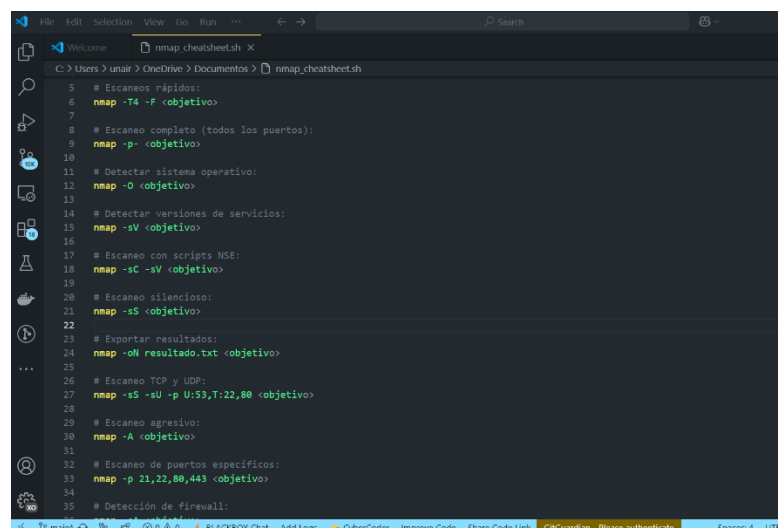
- **Gestión de auditorías:** Desarrollo y documentación de scripts de escaneo y pruebas de penetración.
- **Automatización:** Ejecución de tareas repetitivas mediante snippets y tareas programadas.
- **Control de versiones:** Seguimiento de cambios en scripts y configuraciones.

Técnicas Avanzadas

- Uso de extensiones de seguridad (Nmap, GitLens, REST Client).
- Integración de terminales embebidos para ejecución directa.
- Documentación de procedimientos técnicos y cheatsheets personalizadas.

Mejores Implementaciones

- Repositorio interno de scripts NSE y comandos útiles.
- Integración con pipelines de CI/CD para pruebas de seguridad automatizadas.
-



```
5 # Escaneo rápido:  
6 nmap -T4 -F <objetivo>  
7  
8 # Escaneo completo (todos los puertos):  
9 nmap -p- <objetivo>  
10  
11 # Detectar sistema operativo:  
12 nmap -O <objetivo>  
13  
14 # Detectar versiones de servicios:  
15 nmap -sV <objetivo>  
16  
17 # Escaneo con scripts NSE:  
18 nmap -sC -sV <objetivo>  
19  
20 # Escaneo silencioso:  
21 nmap -sS <objetivo>  
22  
23 # Exportar resultados:  
24 nmap -oN resultado.txt <objetivo>  
25  
26 # Escaneo TCP y UDP:  
27 nmap -sS -sU -p U:53,T:22,80 <objetivo>  
28  
29 # Escaneo agresivo:  
30 nmap -A <objetivo>  
31  
32 # Escaneo de puertos específicos:  
33 nmap -p 21,22,80,443 <objetivo>  
34  
35 # Detección de firewall:
```


Recursos y Repositorios

- Plataforma: <https://code.visualstudio.com>
- Extensión Nmap: <https://marketplace.visualstudio.com/items?itemName=ms-vscode.nmap>
- Ejemplo de script NSE:

lua

-- Ejemplo básico NSE para Nmap

description = *[[Prueba de banner HTTP]]*

author = "Michael"

categories = {"discovery"}

portrule = shortport.http

action = **function**(host, port)

return http.get(host, port, "/")

end

9. CVE Details – Vulnerabilidades Activas

Definición Técnica Ampliada

CVE Details es una base de datos que organiza vulnerabilidades por fabricante, producto, criticidad (CVSS), fecha y vector de ataque. Permite búsquedas avanzadas, filtrado y exportación de datos para análisis y gestión de riesgos.

Usos en Ciberseguridad

- **Gestión de vulnerabilidades:** Identificación y priorización de CVEs relevantes para los activos del SOC.
- **Apoyo a parcheo:** Planificación de actualizaciones y mitigaciones.
- **Correlación con amenazas:** Relación de CVEs con IOC y campañas activas.

Técnicas Avanzadas

- Configuración de alertas automáticas para productos clave.
- Exportación de datos para alimentar sistemas SIEM/SOAR.
- Cruce con MITRE ATT&CK para priorización de respuestas.

Mejores Implementaciones

- Integración de feeds CVE en plataformas de gestión de vulnerabilidades.
- Uso de scripts para automatizar la consulta y descarga de CVEs.

Recursos y Repositorios

- Plataforma: <https://www.cvedetails.com>
- Feed oficial: <https://nvd.nist.gov/vuln/data-feeds>
- Ejemplo de consulta CVE en Python:

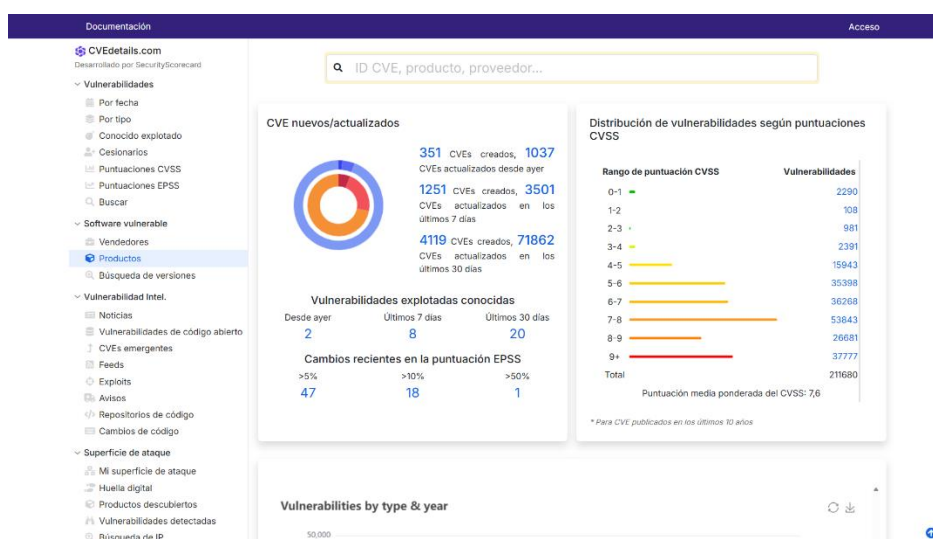
python

```
import requests
```

```
url = "https://services.nvd.nist.gov/rest/json/cves/1.0?keyword=apache"
```

```
response = requests.get(url)
```

```
print(response.json())
```



10. MITRE ATT&CK – Tácticas y Técnicas

Definición Técnica Ampliada

MITRE ATT&CK es un marco de conocimiento que clasifica tácticas, técnicas y procedimientos (TTPs) utilizados por adversarios en campañas reales. Organiza las técnicas en matrices según el entorno (Enterprise, Mobile, ICS) y permite mapear incidentes, crear casos de uso y mejorar la detección defensiva.

Usos en Ciberseguridad

- **Análisis de amenazas:** Mapeo de incidentes a técnicas ATT&CK para entender el ciclo de ataque.
- **Desarrollo de defensa:** Creación de reglas Sigma, YARA y casos de uso basados en TTPs reales.
- **Formación y simulación:** Ejercicios Red vs Blue y análisis post mortem.

Técnicas Avanzadas

- Uso de ATT&CK Navigator para visualización y planificación.
- Correlación de logs y eventos con técnicas específicas.
- Integración con plataformas SIEM/SOAR para automatización de respuestas.

Mejores Implementaciones

- Matriz ATT&CK personalizada para el entorno propio.
- Automatización de mapeo de incidentes a técnicas ATT&CK.

Recursos y Repositorios

- Plataforma: <https://attack.mitre.org>
- Navigator: <https://mitre-attack.github.io/attack-navigator/>
- Repositorio: <https://github.com/mitre/cti>
- Ejemplo de uso con Python:

```
python
import requests
url =
"https://attack.mitre.org/api.php?action=parse&page=Enterprise%20ATT%26CK%20Matrix&format=json"
response = requests.get(url)
print(response.json())
```

11. ChatGPT – Generación de Reglas Sigma

Definición Técnica Ampliada

ChatGPT es un modelo de IA conversacional capaz de generar, revisar y explicar reglas Sigma, scripts y lógica defensiva. Permite automatizar tareas de generación de contenido técnico y acelerar la respuesta ante incidentes.

Usos en Ciberseguridad

- **Generación de reglas Sigma/YARA:** Creación rápida de reglas para detección en SIEM.
- **Scripting defensivo:** Automatización de tareas y generación de scripts personalizados.
- **Soporte técnico:** Explicación de conceptos complejos y apoyo en toma de decisiones.

Técnicas Avanzadas

- Entrenamiento de modelos personalizados para el entorno del SOC.
- Integración con plataformas de gestión de conocimiento.
- Validación automática de reglas generadas antes de despliegue.

Mejores Implementaciones

- Uso de ChatGPT como copiloto defensivo en entornos críticos.
- Almacenamiento de las mejores reglas en repositorios internos.

Recursos y Repositorios

- Plataforma: <https://chat.openai.com>
- API: <https://platform.openai.com/docs/api-reference>
- Ejemplo de prompt para generación Sigma:

text

Prompt: "Genera una regla Sigma para detectar ejecución de PowerShell con argumentos sospechosos."

¿En qué puedo ayudarte?

Genera una regla Sigma para detectar intentos de fuerza bruta SSH en un entorno Linux. La regla debe incluir:

- Nivel de severidad (critical/high).
- Descripción técnica clara.
- Campos clave que la activan.
- Formato YAML válido.

Todo enfocado para defender un SOC real y compatible con Wazuh y otros SIEMs open source.



12. Kaspersky Threat Map – Ciberataques Globales

Definición Técnica Ampliada

Kaspersky Threat Map es una plataforma de visualización en tiempo real de ciberataques detectados globalmente, mostrando vectores, orígenes, destinos y tipos de amenazas. Proporciona contexto visual y apoyo situacional para SOC's y equipos de respuesta.

Usos en Ciberseguridad

- **Monitorización global:** Identificación de tendencias y campañas activas por país o sector.
- **Análisis estratégico:** Apoyo a la toma de decisiones ante picos de actividad maliciosa.
- **Formación:** Refuerzo visual para equipos Blue Team y concienciación.

Técnicas Avanzadas

- Correlación de patrones visuales con logs internos y feeds de amenazas.
- Uso de la Threat Map para justificar refuerzo de medidas defensivas.
- Integración con sistemas de alerta y dashboards.



Mejores Implementaciones

- Visualización en tiempo real en el SOC para contextualizar incidentes.
- Uso combinado con informes de inteligencia y noticias de ciberseguridad.


Recursos y Repositorios

- Plataforma: <https://cybermap.kaspersky.com>
- Alternativa API: <https://www.abuseipdb.com>
- Ejemplo de consulta de amenazas con Python:

```
python
import requests
url = "https://api.abuseipdb.com/api/v2/check"
headers = {'Key': 'YOUR_API_KEY'}
params = {'ipAddress': '8.8.8.8'}
response = requests.get(url, headers=headers, params=params)
print(response.json())
```

AVISO ÉTICO

Este documento ha sido creado con fines formativos, defensivos y educativos. Toda la información y herramientas aquí mostradas están orientadas a reforzar la ciberseguridad, no a vulnerarla.

 Su uso está sujeto a principios de ética profesional, legalidad y responsabilidad técnica.

 Conocimiento compartido = defensa colectiva.

Unai Rubio

Hacker Ético · Mayo 2025