

Sistema de Monitoreo OSINT (Open Source Intelligence) para la Detección y Notificación Temprana de Sitios Web Comprometidos.

Jose Alejandro Serna Castañeda

Informe final para optar al título de Ingeniera de Telecomunicaciones

Semestre de Industria

Asesor Interno
Prof. Jaime Alberto Vergara Tejada, MSc

Asesor Externo
Francisco Javier Muñoz Cortes, MSc

Universidad de Antioquia
Facultad de Ingeniería
Ingeniería de telecomunicaciones
Medellín - Antioquia
2025

Cita	(Serna Castañeda Jose, 2025)
Referencia **Estilo APA 7 (2020)**	Serna Castañeda, J. (2025). Sistema de Monitoreo OSINT (Open Source Intelligence) para la Detección y Notificación Temprana de Sitios Web Comprometidos. Semestre de Industria, Ingeniería de Telecomunicaciones, Universidad de Antioquia, Medellín, 2025.

creative commons

UNIVERSIDAD DE ANTIOQUIA
Vicerrectoría de Docencia

Sistema de Bibliotecas

Centro de Documentación Ingeniería (CENDOI)

Repositorio Institucional: http://bibliotecadigital.udea.edu.co

Universidad de Antioquia - www.udea.edu.co

Rector: John Jairo Arboleda Céspedes.
Decano/Director: Julio César Saldarriaga Molina.
Jefe departamento: Eduard Emiro Rodríguez Ramírez.

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

Dedicatoria

Dedico este trabajo a mi familia, que siempre fue mi apoyo incondicional, brindándome su amor, paciencia y motivación en cada etapa de este camino. A mis amigos, por las risas interminables, las anécdotas que quedarán para siempre y por estar ahí en todo momento. Y a mis compañeros, con quienes compartí largas jornadas de estudio, esfuerzo y aprendizajes que nos hicieron crecer juntos. Gracias a cada uno por ser parte de este recorrido y por hacer que cada paso valiera la pena.

Agradecimientos

Me siento profundamente agradecido con mi familia, quienes fueron un pilar fundamental en este largo proceso, brindándome siempre su apoyo incondicional y confianza invaluable en cada paso. Formar parte de la Universidad de Antioquia ha sido una experiencia invaluable y significativa, y agradezco profundamente a mis profesores por compartir con dedicación sus conocimientos, su guía y por ser una inspiración en mi crecimiento académico y profesional.

Con especial gratitud, agradezco a Aligo Defensores Informáticos, donde realicé mi práctica académica, por brindarme la oportunidad de aprender, confiando en mí incluso sin experiencia.
A todos aquellos que, de una u otra forma, hicieron parte de este recorrido y aportaron a mi crecimiento, muchas gracias.

Tabla de contenido

Resumen..... 7
Abstract..... 8
1. Introducción..... 9
2. Objetivos..... 10
    2.1 Objetivo general..... 10
    2.2 Objetivos específicos..... 10
3. Presentación de la empresa..... 11
4. Marco teórico..... 11
5. Metodología..... 15
6. Análisis de resultados..... 23
7. Conclusiones y recomendaciones..... 27
Referencias..... 28
Anexos..... 29

Lista de figuras

Fig 1: Ejemplos banners defacements 12
Fig 2: IPC2 (computador industrial). 16
Fig 3: Módulos del sistema 16
Fig 4: Repositorio deepdarkCTI. 18
Fig 5: Interfaz web zone-xsec.com/archive 19
Fig 6: Interfaz web mirror-h.org/archive 20
Fig 7: Ejemplo Notificación de alerta en Discord 22
Fig 8: Historico de entradas en mirror-h.org 24
Fig 9: Mapa de calor y diagrama de países con más reportes de defacement. 24
Fig 10: Grupos hackers con más ataques defacement reportados en mirror-h.org 25
Fig 11: Evidencias indisponibilidad sitio mirror-h.org 25
Fig 12: Front-End del sitio zone-xsec.com donde se visualiza su control anti DDoS 26
Fig 13: Prohibición de entrada al sitio zone-xsec.com 26
Fig 14: Cookie implementada por el sitio zone-xsec.com 27

Siglas, acrónimos y abreviaturas

MSc	Magister Scientiae
PhD	Philosophiae Doctor
UdeA	Universidad de Antioquia
OSINT	Open Source Intelligence
ML	Machine Learning
KNN	K-nearest neighbors
SVM	Support Vector Machine
SQL	Structured Query Language
FTP	File Transfer Protocol
FIG	Figura
DDoS	Distributed Denial Of Service
NAC	Network Access Control
SOC	Security Operations Center

Resumen

Este proyecto presenta el diseño, desarrollo e implementación de un sistema de monitoreo OSINT (Open Source Intelligence) enfocado en la detección y notificación temprana de sitios web que han sido comprometidos mediante ataques de tipo defacement. A lo largo del documento se definen en detalle los componentes del sistema, su estructura general, así como las tecnologías utilizadas para su funcionamiento. También se describen las metodologías aplicadas para la recopilación, análisis, resultados y hallazgos.

El sistema fue sometido a pruebas funcionales que permitieron validar su desempeño, obteniendo resultados positivos en la detección y notificación de incidentes de seguridad o ataques defacement. Sin embargo, se identificaron ciertas limitaciones relacionadas con la estabilidad y confiabilidad de las fuentes de información utilizadas, lo que abre la puerta a futuras mejoras. A pesar de estos inconvenientes, se concluye que el sistema es funcional y representa una herramienta útil para el monitoreo OSINT.

Palabras clave: osint, defacement, hacking.

Abstract

This project presents the design, development, and implementation of an OSINT (Open Source Intelligence) monitoring system focused on the early detection and notification of websites that have been compromised through defacement attacks. The system's components, overall structure, and the technologies used for its operation are described in detail throughout the document. The methodologies applied for the collection, analysis, and filtering of information from open sources are also described.

The system underwent functional testing that validated its performance, yielding positive results in detecting and notifying security incidents or defacement attacks. However, certain limitations were identified regarding the stability and reliability of the information sources used, which opens the door to future improvements. Despite these challenges, it is concluded that the system is functional and represents a valuable tool for OSINT monitoring.

***Keywords*:** osint, defacement, hacking.

1. Introducción

En el contexto actual de la ciberseguridad, los ataques a sitios web representan una amenaza constante para empresas de todos los sectores. Estos incidentes no solo generan pérdidas económicas, sino que también afectan la reputación y la confianza de los clientes[1]. Entre los diferentes tipos de ataques, el defacement es una técnica común utilizada por atacantes para modificar el contenido de los sitios web afectados, dejando mensajes visibles y exponiendo vulnerabilidades de seguridad. La detección temprana de estos ataques es crucial para mitigar sus efectos y evitar mayores impactos.

Aligo Defensores Informáticos S.A.S. es una empresa colombiana especializada en la protección de infraestructuras digitales, ofreciendo soluciones avanzadas en ciberseguridad para sus clientes. Durante el desarrollo de mi práctica académica en la empresa, se identificó la necesidad de un sistema automatizado para la detección de defacement, con el fin de monitorear, registrar y notificar estos incidentes de manera eficiente.

Esta problemática cobró especial relevancia tras un evento particular ocurrido en una reunión con un cliente. Mientras se realizaba una demostración en vivo de los sitios web que publican reportes de defacement, se identificó un caso relacionado con Colombia. Este hallazgo puso de manifiesto la importancia de contar con herramientas que permitan detectar rápidamente estos ataques, evitando daños a la reputación y seguridad de las organizaciones afectadas.

Diversas soluciones de monitoreo han abordado la detección de sitios web comprometidos; sin embargo, muchas dependen de reportes manuales o búsquedas específicas mediante técnicas como el Google dorking[2]. Aunque existen plataformas que publican información sobre ataques de defacement, estas no siempre permiten una integración directa o un análisis automatizado. Ante esta problemática, el presente proyecto propone el desarrollo de un sistema de monitoreo automatizado basado en técnicas de web scraping, orientado a la recolección de información sobre ataques de defacement en sitios web. El sistema extrae datos clave como la fecha del ataque, el grupo atacante, la URL afectada y el país de origen del sitio comprometido.

Este proyecto es relevante porque permite fortalecer las estrategias de defensa en ciberseguridad, aportando a la ingeniería una solución práctica para la detección y monitoreo de ataques. Además, su implementación en Aligo Defensores Informáticos S.A.S. Tiene el potencial de optimizar los servicios de la empresa, estableciendo una nueva herramienta que amplíe su portafolio de soluciones de seguridad digital.

2. Objetivos

2.1 Objetivo general

Desarrollar un sistema automático de detección y notificación de sitios web comprometidos, dirigido a empresas colombianas y clientes de ALIGO DEFENSORES INFORMÁTICOS SAS, que genere alertas tempranas y se despliegue en una infraestructura estable y segura, utilizando herramientas para el monitoreo y análisis de la información recopilada de los sitios de reportes de defacement.

2.2 Objetivos específicos

- Definir los componentes y la estructura del sistema a implementar, considerando los requisitos funcionales y no funcionales, para asegurar que la solución cumpla con los objetivos establecidos y funcione de manera eficiente y estable.

- Desarrollar un componente que detecte sitios web comprometidos que cuente con un módulo para procesar la información recopilada, garantizando la identificación precisa de páginas y recursos afectados.

- Implementar un sistema de información que se integre a la solución, utilizando herramientas que faciliten la automatización de las notificaciones oportunas sobre posibles amenazas asegurando que las notificaciones lleguen al equipo responsable de su gestión.

- Analizar y evaluar el desempeño del sistema, verificando su efectividad en la detección, notificación y procesamiento de la información recopilada.

3. Presentación de la empresa

Aligo Defensores Informáticos S.A.S. es una empresa colombiana con sede en Medellín y más de 15 años de experiencia en ciberseguridad. Especializada en la protección de infraestructuras digitales y la seguridad de la información, ofrece soluciones adaptadas a los riesgos y necesidades específicas de cada industria.

Cuenta con un equipo interdisciplinario que desarrolla estrategias de seguridad tanto ofensiva como defensiva. En la parte defensiva, proporciona monitoreo y análisis de eventos de seguridad, detección y respuesta ante incidentes, gestión de vulnerabilidades, implementación de controles de acceso a la red (NAC) y administración de centros de operaciones de seguridad (SOC). En el ámbito ofensivo, realiza pruebas de penetración en redes y aplicaciones, auditorías de infraestructura, evaluación de configuraciones de seguridad y simulaciones de ataques avanzados para identificar y mitigar posibles brechas.

4. Marco teórico

El defacement o desfiguración de sitios web es una técnica utilizada en el hacktivismo, donde los atacantes modifican el contenido de una página para transmitir un mensaje político, ideológico o de protesta. En muchos casos, el objetivo es generar conciencia sobre una causa específica, desacreditar a una institución o incurrir en un daño reputacional al alterar la imagen pública de la entidad afectada.

Los ataques de defacement pueden tener diversas motivaciones dependiendo del atacante, algunos buscan evidenciar fallas en la seguridad, como en el caso de quienes advierten sobre vulnerabilidades y recomiendan corregirlas. Otros tienen motivaciones políticas o ideológicas, usando estos ataques como medio de protesta contra gobiernos o conflictos internacionales. También existen quienes lo hacen por reconocimiento dentro de la comunidad hacker, sin un objetivo más allá de demostrar sus habilidades. Como se observa en los ejemplos mostrados en la Fig 1, los mensajes varían desde advertencias sobre ciberseguridad hasta declaraciones con carga política o intentos de ganar notoriedad. Estas diferencias reflejan la variedad de actores

involucrados en este tipo de incidentes y la importancia de contar con mecanismos efectivos de detección y respuesta.

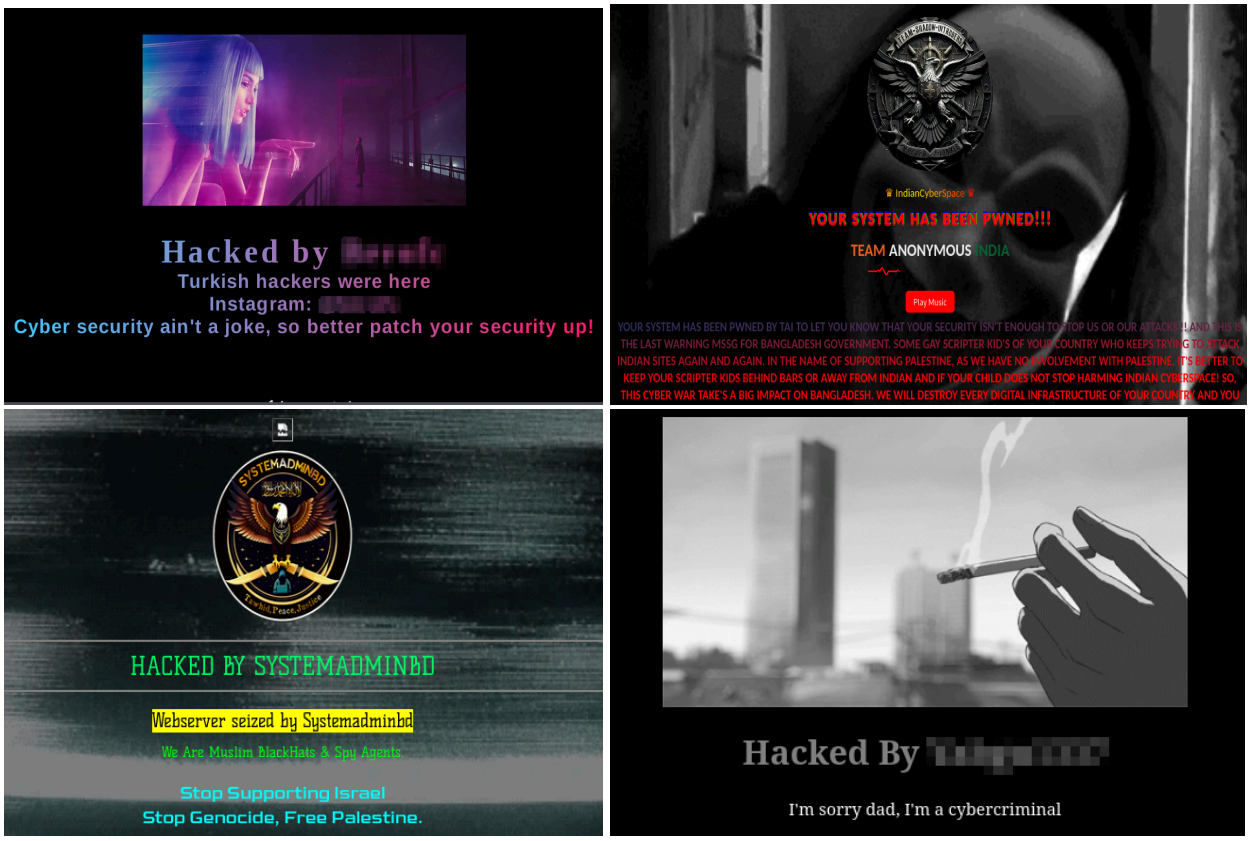


Fig 1: Ejemplos banners defacements

En Colombia, este fenómeno ha cobrado relevancia debido a conflictos sociales y políticos, donde grupos hacktivistas han dirigido ataques contra entidades gubernamentales, empresas privadas e instituciones educativas. Estos ataques pueden estar motivados por protestas contra decisiones gubernamentales, desacuerdos con políticas institucionales o la búsqueda de reconocimiento dentro de la comunidad[3]. Aunque en algunos casos los atacantes buscan exponer fallos de seguridad, también se pueden generar pérdidas económicas y afectar la reputación de las organizaciones comprometidas [4]. Los ataques de defacement suelen aprovechar vulnerabilidades en servidores y aplicaciones web, facilitando el acceso no autorizado para modificar su contenido[5].

Según los datos recopilados por Zone-H, los métodos más utilizados para desfigurar sitios web entre 2010 y 2016 incluyen vulnerabilidades no especificadas en aplicaciones web (27.22%), seguidas de la inyección SQL (18.00%) y otros métodos variados (18.00%). Luego, se encuentran los ataques de fuerza bruta (7.00%), la inclusión de archivos maliciosos (6.39%), la explotación de vulnerabilidades conocidas (6.32%) y la manipulación de enlaces mediante URL Poisoning (3.76%). Otros métodos registrados incluyen la intrusión en servidores FTP (3.11%), técnicas de ingeniería social (3.00%), errores de configuración en permisos y accesos (2.38%), intrusiones en servidores SSH (2.18%), ataques a servidores de correo (1.15%), ataques DNS (0.60%) y ataques de MiTM (Man-In-The-Middle) con un 0.30%. Estos datos evidencian la diversidad de estrategias utilizadas por los atacantes[3].

Los ataques de defacement no solo generan problemas de seguridad, sino que también provocan afectaciones operativas, reputacionales y riesgos adicionales. A nivel operativo, una vez que un sitio web ha sido comprometido, puede volverse inaccesible o mostrar contenido no autorizado, lo que interrumpe el acceso a información o servicios críticos. La recuperación del sitio implica costos significativos para las organizaciones, que deben invertir en restaurar copias de seguridad, reparar vulnerabilidades y reforzar sus medidas de protección para evitar futuros incidentes [3].

En términos de reputación, la presencia de contenido ofensivo o inapropiado en un sitio afectado puede debilitar la confianza de los usuarios y clientes. Para organizaciones que manejan información sensible o dependen de su imagen pública, un ataque exitoso puede tener consecuencias graves en la percepción de seguridad y confiabilidad. Los clientes pueden dudar de la capacidad de la entidad para proteger su información, lo que podría traducirse en pérdidas económicas o reducción en la lealtad de los usuarios [3]. Además de los daños inmediatos, los ataques de defacement pueden representar riesgos adicionales para los sistemas comprometidos y sus visitantes. En algunos casos, los atacantes aprovechan estas intrusiones para propagar malware, afectando a quienes acceden al sitio modificado. Asimismo, el hecho de que un sitio haya sido desfigurado sugiere la existencia de vulnerabilidades que podrían ser explotadas en ataques más graves, como el robo de datos o propagación de todo tipo de malware[3].

Para analizar el histórico de los ataques de defacement, existen varias bases de datos que almacenan registros de estos incidentes. Una de las más conocidas es Zone-H, que ha registrado ataques desde 2001 y recopila información detallada sobre cada incidente, incluyendo el dominio afectado, la fecha y hora del ataque, el nombre del atacante, el sistema operativo y servidor web comprometido, la metodología de ataque, el tipo de intrusión y si el ataque fue masivo o individual [1]. Sin embargo, Zone-H no es la única base de datos disponible para este propósito. También existen Mirror-H y Zone-Xsec, plataformas similares que registran y documentan ataques de defacement reportados por los propios atacantes o por investigadores de seguridad.

Los métodos de detección de ataques de defacement han evolucionado con el tiempo para adaptarse a las nuevas técnicas utilizadas por los atacantes. Existen tres enfoques principales para la detección proactiva de estos incidentes: detección basada en anomalías, detección basada en firmas y técnicas de aprendizaje automático. La detección basada en anomalías emplea métodos como la verificación de checksums y herramientas de comparación de diferencias para identificar modificaciones en el contenido web. Por otro lado, la detección basada en firmas utiliza patrones previamente identificados de ataques para generar alertas. Sin embargo, uno de los enfoques más avanzados es el uso de aprendizaje automático, donde algoritmos como redes neuronales convolucionales (CNN), árboles de decisión y máquinas de soporte vectorial (SVM) permiten clasificar páginas web como legítimas o comprometidas con alta precisión. Estas técnicas han demostrado reducir significativamente las falsas alarmas y mejorar la capacidad de respuesta ante incidentes de seguridad [4].

Además de los métodos de detección, las acciones frente a un ataque de defacement deben ejecutarse con rapidez para mitigar su impacto. Entre las medidas inmediatas se encuentran la restauración de copias de seguridad, la identificación y cierre de vulnerabilidades explotadas y la implementación de herramientas de monitoreo continuo. Asimismo, la integración de firewalls de aplicaciones web (WAF), autenticación multifactor (MFA) y controles de acceso estrictos puede reducir la exposición a estos ataques. En un nivel más avanzado, el uso de inteligencia artificial y modelos de machine learning ha permitido desarrollar soluciones proactivas que no solo detectan ataques en tiempo real, sino que también predicen posibles vulnerabilidades antes de ser explotadas [4].

Para reducir el riesgo de ataques de defacement y fortalecer la seguridad de los sitios web, es fundamental adoptar buenas prácticas que minimicen las vulnerabilidades explotadas por los atacantes. Según el Grupo de Respuestas a Emergencias Cibernéticas de Colombia (COLCERT) y el Ministerio TIC [6], estas son las prácticas para prevenir los ataques:

- Mantener actualizados todos los componentes del sitio.
- Eliminación de complementos innecesarios.
- Protección contra inyecciones SQL y XSS.
- Utilizar HTTPS y SSL/TLS.
- Implementar el principio de privilegio mínimo.
- Capacitación continua del personal.
- Auditorías y escaneos regulares.
- Gestión adecuada de archivos cargados.
- Protocolos de respuesta.

5. Metodología

Para el desarrollo del proyecto, se llevaron a cabo diversas actividades organizadas en cuatro fases: identificación y análisis de requisitos, procesamiento de información, integración del sistema y validación del desempeño. A continuación, se describe cada una en detalle.

1. Identificación y análisis de requisitos

1.1. Identificación y análisis de los requisitos funcionales y no funcionales

Se realizó un análisis de los requerimientos del sistema, diferenciando entre requisitos funcionales (acciones que debe realizar el sistema) y no funcionales (rendimiento, seguridad, escalabilidad). Para ello, se llevaron a cabo reuniones con los líderes de Aligo y los asesores de la práctica, con el fin de discutir el proyecto y determinar cómo, dónde y cuál sería la infraestructura adecuada y disponible para funcionar como laboratorio de pruebas. Como resultado de estas reuniones, se estableció que el personal de TI proporciona soporte para la implementación de una VLAN aislada dentro de la red de Aligo, con acceso a internet pero sin

conexión a la red interna, permitiendo así realizar consultas a internet y facilitar la experimentación necesaria para el desarrollo del proyecto. Además, se brindó un PC o servidor satélite mostrado en la Fig 2, el cual se encargará de contener los componentes del proyecto y de la ejecución de las tareas necesarias dentro del laboratorio de pruebas.



Fig 2: IPC2 (computador industrial).

1.2. Especificación de módulos del sistema

Se definieron tres módulos esenciales para el sistema: recopilación de información, encargado de realizar solicitudes HTTP a sitios que reportan defacements; procesamiento de datos, que filtra la respuesta HTTP obtenida eliminando contenido irrelevante, la almacena y la compara con un conjunto de reglas para determinar si es necesario generar una alerta; y notificación de alertas, que comunica los eventos detectados a través de un canal en Discord. La Figura 3 presenta la estructura general del sistema.



Fig 3: Módulos del sistema

1.3. Diseño de la arquitectura del sistema

Se definieron los componentes principales del sistema, su interacción y la infraestructura necesaria para su implementación. La arquitectura se diseñó con un enfoque modular a nivel de código, estructurando el sistema en funciones independientes dentro de un mismo script. Esto permite una mayor organización, reutilización de código y facilidad para realizar mantenimientos o ampliaciones en el futuro.

El sistema está compuesto por los siguientes elementos clave:

- Módulo de recopilación de información: Responsable de realizar consultas HTTP a los sitios web que reportan defacements, obteniendo los datos relevantes.
- Módulo de procesamiento de datos: Encargado de analizar, estructurar y almacenar la información obtenida para su posterior uso.
- Módulo de notificación de alertas: Evalúa cuándo se debe generar una alerta y define los medios para su notificación, utilizando Discord como canal de comunicación.
- Almacenamiento de datos: Los datos recopilados se guardan en un archivo CSV, permitiendo su estructuración y posterior uso en otros módulos.
- Visualización de datos: Se implementó una integración con Elastic, donde la información almacenada en el CSV se procesa y se visualiza en series de tiempo.
- Entorno de ejecución: El sistema opera sobre una máquina con sistema operativo Debian, dentro de una VLAN aislada de la red interna con acceso a internet, proporcionando un entorno seguro para la ejecución de las consultas y procesamiento de la información.

1.4. Selección de herramientas y tecnologías

Se evaluaron diversas tecnologías para el desarrollo del sistema, priorizando la compatibilidad con los requisitos del entorno, el rendimiento y la facilidad de mantenimiento. Se optó por Debian como sistema operativo por ser una distribución open source, estable y ligera, lo cual permitió una implementación rápida en el entorno de pruebas. El sistema se desarrolló mediante un script en Python, organizado en funciones específicas que separan claramente las responsabilidades dentro del código, lo que facilita su mantenimiento y escalabilidad. Entre las funcionalidades implementadas, destaca el módulo de alertas, diseñado para integrarse con Discord y enviar notificaciones automáticas a los canales correspondientes. Por otro lado, se

utilizó Elastic como herramienta externa para la visualización de datos en series de tiempo. Para ello, el sistema genera archivos CSV con la información procesada, los cuales se cargan posteriormente en Elastic para su análisis.

2. Procesamiento de información

2.1. Investigación y selección de fuentes de datos

Para garantizar la calidad de los datos recopilados, se establecieron los siguientes criterios de selección de fuentes:

- Disponibilidad y vigencia: Se priorizaron sitios que están activos y actualizados de manera frecuente. Para obtener fuentes sobre defacement, se utilizó el repositorio de deepdarkCTI[7],un repositorio de código abierto Fig 4, que mantiene listados de fuentes en línea sobre amenazas en la deep web y dark web, así como sitios afectados por defacement.

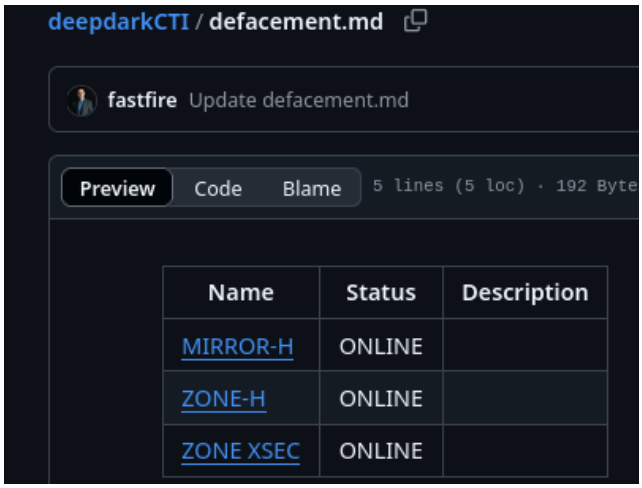


Fig 4: Repositorio deepdarkCTI.

- Accesibilidad de la información: Se seleccionaron fuentes que permiten la visualización y extracción de datos, de los 3 sitios listados por deepdarkCTI se escogen dos de los sitios [8][9] y se excluye un sitio [10] ya que tiene implementado CAPTCHAS que dificultan las consultas.

- Fiabilidad y reputación: Se consideraron fuentes reconocidas en la comunidad de ciberseguridad y hacking, que han demostrado ofrecer información verídica y relevante.
- Compatibilidad con el sistema: Se verificó que la información pudiera ser procesada eficientemente por el sistema, asegurando formatos adecuados para su recopilación y análisis.

En las figuras 5 y 6 se muestra la interfaz gráfica de los sitios escogidos para el desarrollo del sistema

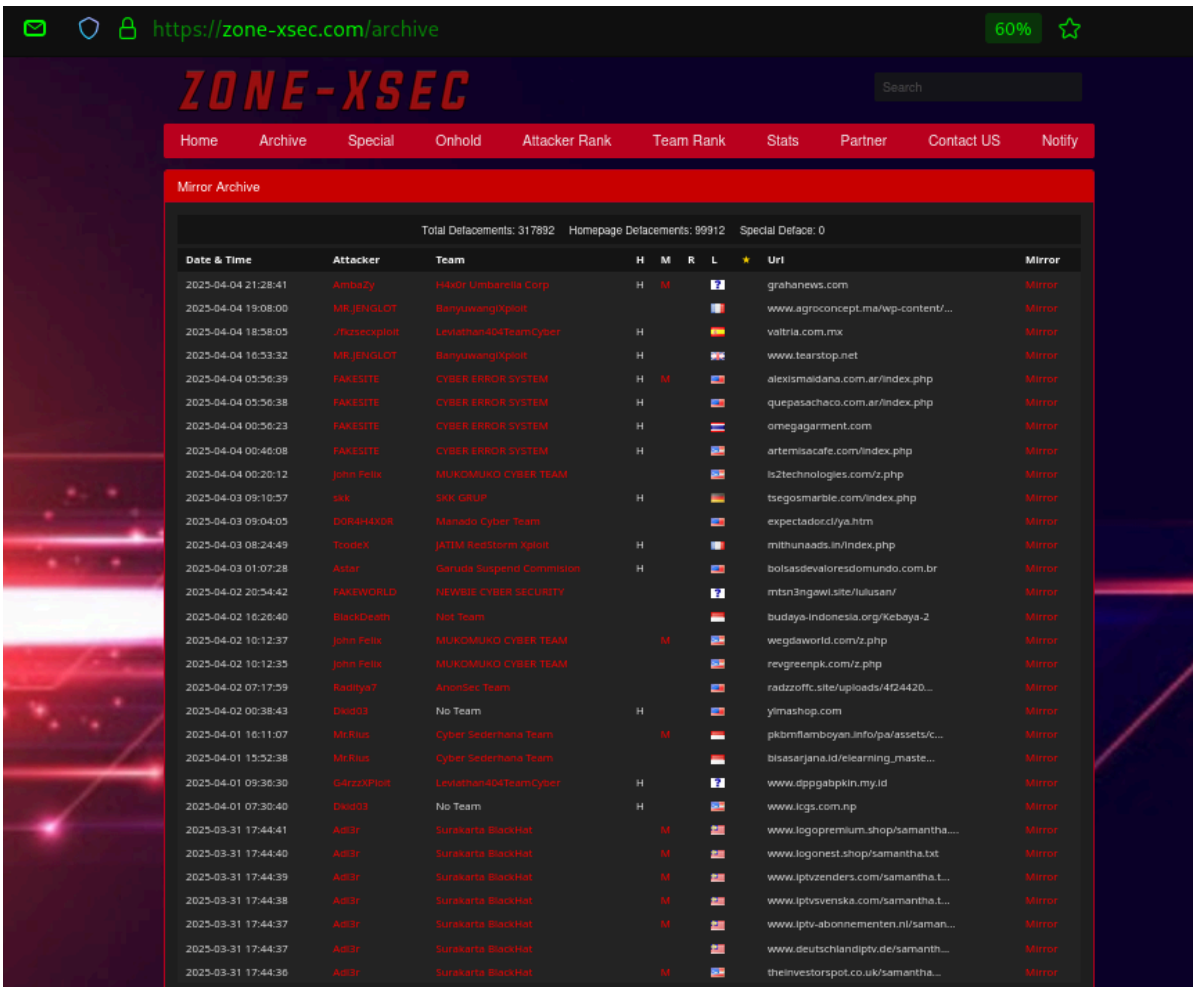


Fig 5: Interfaz web zone-xsec.com/archive

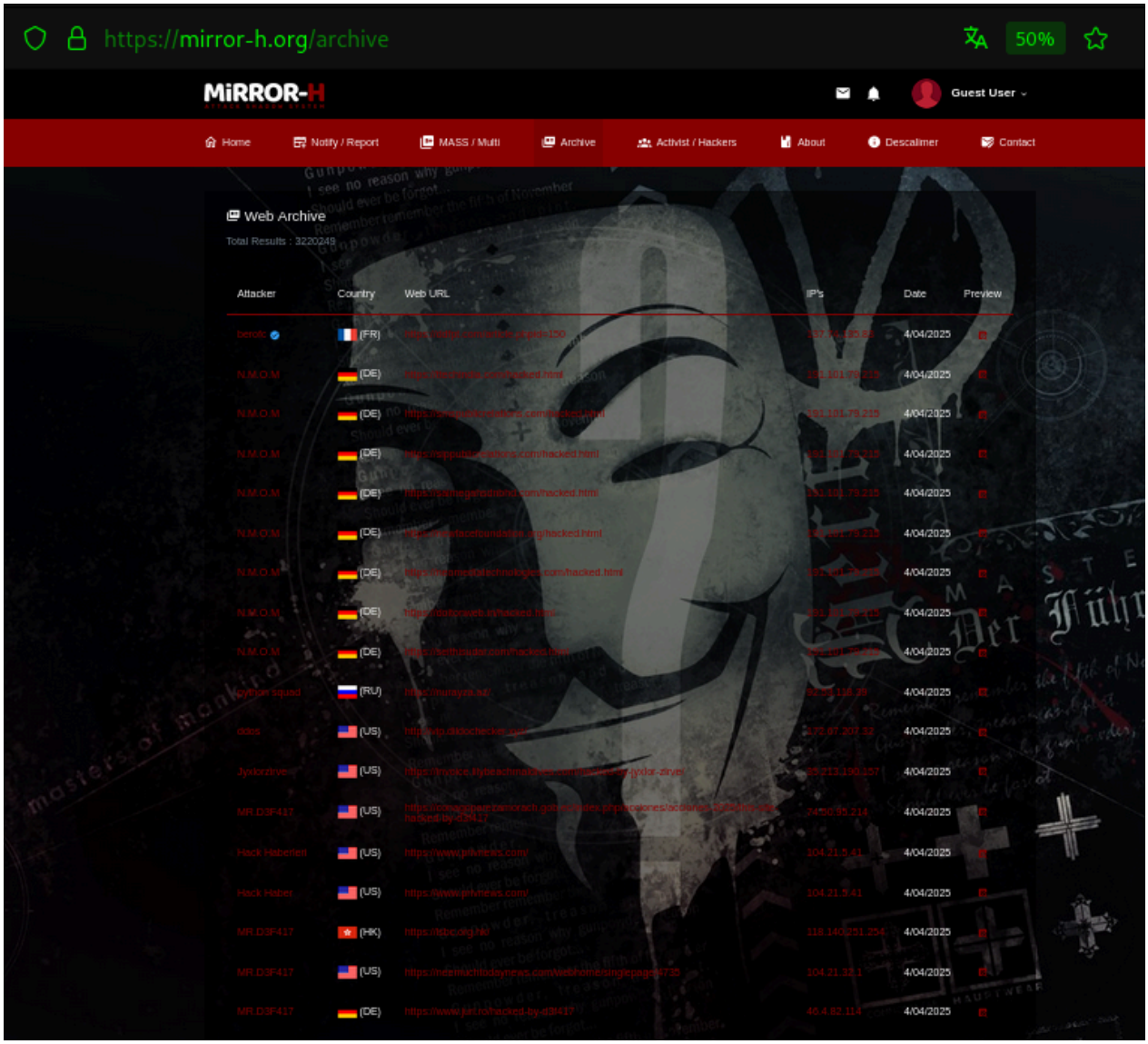


Fig 6: Interfaz web mirror-h.org/archive

2.2. Implementación del módulo de procesamiento de datos

Se desarrolló un módulo encargado de procesar la información obtenida desde sitios web que reportan defacements. Para ello, se implementó un mecanismo que realiza solicitudes HTTP a dichas fuentes, extrayendo los datos relevantes de sus páginas web. El proceso se llevó a cabo en Python, utilizando el paquete OS y el comando cURL para obtener el contenido de las páginas y aplicando técnicas de parsing y scrapping para identificar y extraer las secciones del código HTML que contienen la información necesaria. Posteriormente, los datos extraídos se estructuraron en un formato adecuado para su análisis y almacenamiento.

Para reducir el tiempo de procesamiento, se establecieron criterios de filtrado que permiten descartar información irrelevante y asegurar que sólo se conserven los campos esenciales como URL afectada, país de la URL afectada, nombre del atacante, y link de copia del ataque. Finalmente, la información procesada se almacena en un archivo CSV, facilitando su uso por otros módulos del sistema, como la detección de alertas y la visualización de datos en Elastic.

2.3. Definición del intervalo de escaneo

Se estableció un intervalo de 30 minutos para el escaneo de sitios de defacement, considerando la frecuencia de actualización de los datos. La configuración se ajustó para garantizar la recopilación periódica de información sin interrupciones. Para ello, se implementó una tarea programada en el sistema operativo, encargada de ejecutar cada media hora el script de Python responsable de realizar las consultas HTTP a los sitios web de interés. Este proceso permite que el módulo de recopilación de información se ejecute de manera automatizada, asegurando que los datos extraídos sean actualizados constantemente. Además, el sistema de recopilación incluye un mecanismo que detecta nuevas entradas, evitando la duplicación de información en los archivos CSV. De esta manera, solo se almacenan datos nuevos, reduciendo el procesamiento y asegurando que los módulos de análisis y notificación trabajen con información relevante y actualizada.

2.4. Comparación de resultados y análisis de precisión

Para validar el correcto funcionamiento del sistema de recopilación, se realizó una verificación manual de los sitios de defacement con el fin de comprobar que las nuevas entradas fueran correctamente detectadas y almacenadas. Adicionalmente, cuando no se registraban nuevas entradas, se revisaban los archivos CSV para asegurarse de que el sistema estuviera descartando correctamente la información duplicada. Este proceso permitió identificar posibles fallos en la extracción y almacenamiento de datos, lo que facilitó ajustes en los módulos de recopilación y procesamiento para mejorar la precisión y eficiencia del sistema.

3. Integración del sistema y generación de alertas

3.1. Configuración del sistema de almacenamiento y gestión de datos

Se implementó un sistema de almacenamiento para gestionar los datos recopilados, asegurando su compatibilidad con el resto de la solución y facilitando el acceso eficiente a la información, usando archivos .CSV.

3.2. Integración con herramientas de automatización

Se integró el sistema con herramientas que permiten el envío automático de alertas tempranas en función de los datos procesados. Para garantizar una comunicación oportuna, se estableció un protocolo basado en un webhook de Discord.

3.3. Validación del sistema de alertas

Se llevaron a cabo pruebas para verificar que las alertas se generan correctamente y fueran enviadas a los destinatarios previstos, en la Fig 7, se muestra un ejemplo de las notificaciones que se reciben en el webhook de Discord.

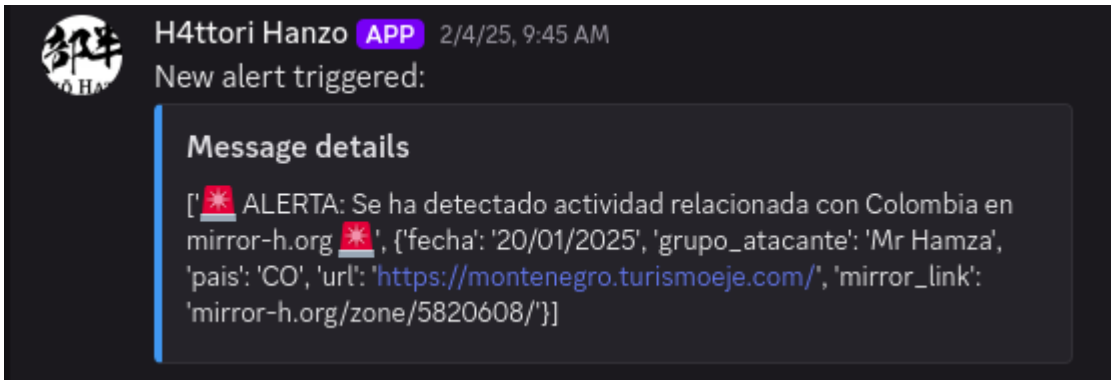


Fig 7: Ejemplo Notificación de alerta en Discord

4. Pruebas y documentación

4.1. Diseño del protocolo de pruebas

Se definió un protocolo para evaluar el desempeño del sistema:

- Entorno y método de prueba: La prueba se realizará en el mismo entorno definido para la ejecución del sistema, verificando manualmente la salida del script en consola y comparándola con la información visible en el sitio web a través del navegador. Además, se validará que la cantidad de entradas o reportes obtenidos por cada página del sitio coincida con lo mostrado en la web, siendo en ambos casos un total de 30 reportes de defacement por página.

- Criterios de éxito: La salida del script debe coincidir exactamente con la información del sitio, sin errores ni datos faltantes.
- Criterios de fallo: Cualquier diferencia entre la salida del script y el contenido del sitio será considerada un fallo.

4.2. Revisión y ajustes del sistema

Con base en los resultados de las pruebas, se realizaron ajustes en los módulos desarrollados para optimizar su rendimiento y corregir posibles fallas

4.3. Documentación y elaboración del informe final

Se documentaron los hallazgos y ajustes realizados en el sistema. Finalmente, se elaboró este informe detallado sobre el desempeño, resultados obtenidos y conclusiones del proyecto.

6. Análisis de resultados

Durante la ejecución del proyecto se realizaron dos pruebas. La primera consistió en recorrer de forma completa el historial del sitio mirror-h.org, el cual ofrece acceso a todos los registros históricos publicados desde el año 2015. Para ello, se desarrolló un script que automatiza la navegación y recolección de datos, el cual tomó aproximadamente 12 días en completar el recorrido por todo el historial disponible. Como resultado, se obtuvo una base de datos consolidada en formato CSV con todos los incidentes documentados hasta la fecha de ejecución de la prueba. Dicho archivo se entrega como anexo a este informe.

Gracias a la integración con la herramienta Elasticsearch, fue posible generar una serie de gráficos, En la Figura 8 se presenta el historial de entradas registradas en mirror-h.org, evidenciando una actividad constante desde 2016, con una ausencia total de registros durante el año 2022, probablemente debido a que el sitio no estuvo disponible. La Figura 9 muestra la distribución de los reportes de defacement por país, donde Estados Unidos concentra la mayoría de los ataques registrados (46.65%), seguido por Turquía (4.46%), Alemania (3.57%) y Países Bajos (3.49%). Los reportes sin país asociado (NA) representan un 10.76% del total, mientras que la categoría “Other” agrupa el 31.07% restante, correspondiente a todos los demás países que

no se mencionan de forma individual. Por último, la Figura 10 refleja los grupos hackers con más reportes de ataques, donde el 75.31% corresponde a actores clasificados como “Other”, mientras que los grupos más identificables son Yahudi (6.12%), FurkanLoren (5.54%), Cyber Team (4.58%) y TurkHacks (4.07%), entre otros con menor participación.

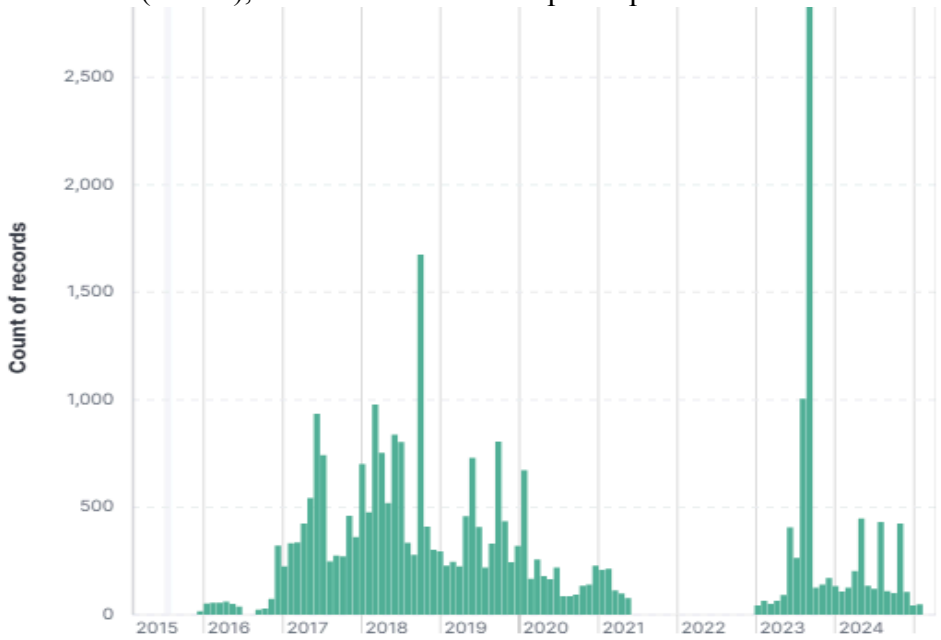


Fig 8: Historico de entradas en mirror-h.org

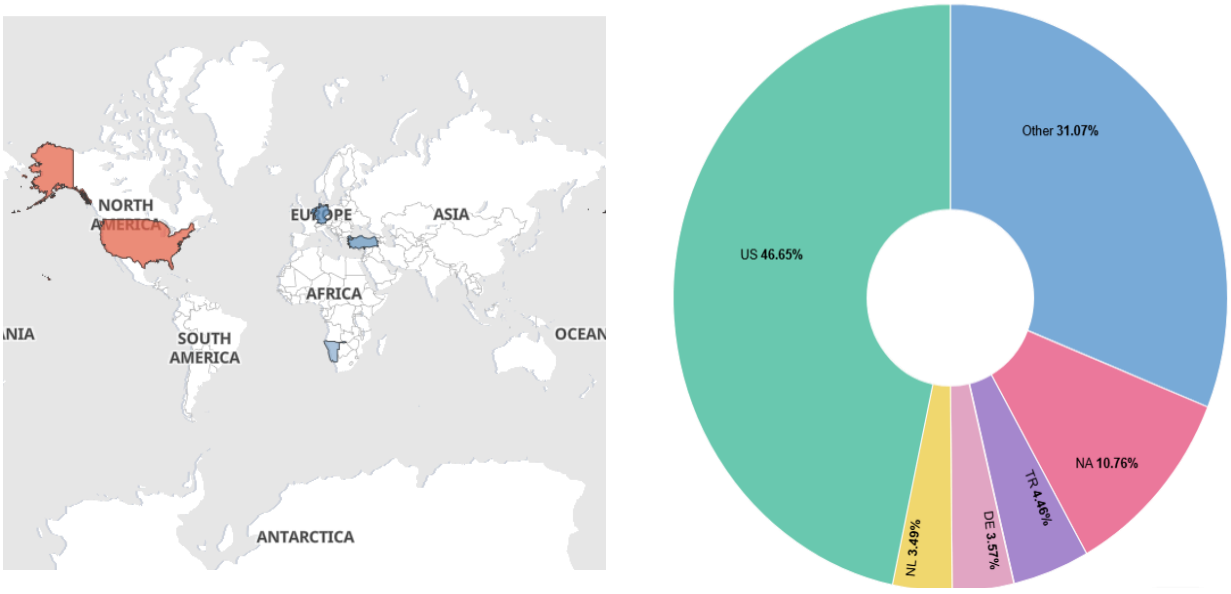


Fig 9: Mapa de calor y diagrama de países con más reportes de defacement.

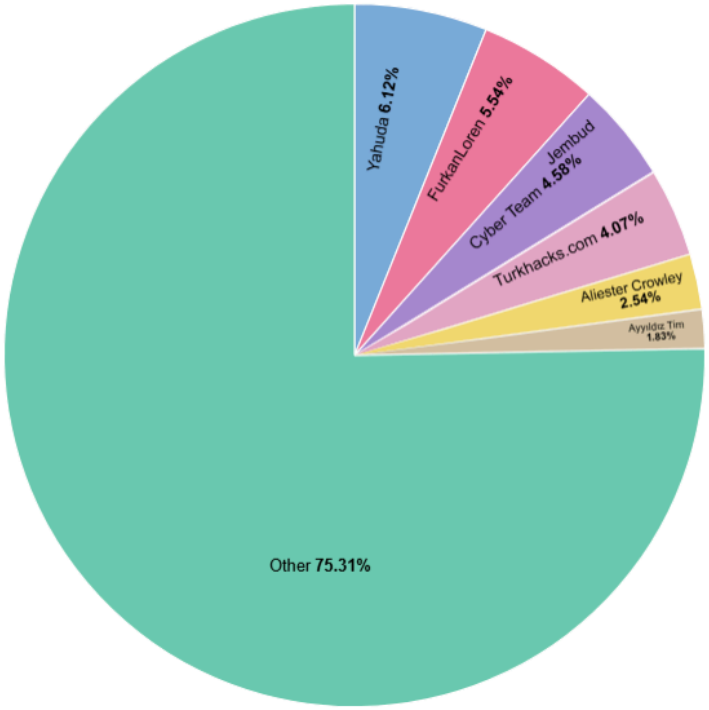


Fig 10: Grupos hackers con más ataques defacement reportados en mirror-h.org

Una vez recopilado el histórico completo, se procedió a configurar pruebas automáticas utilizando crontab, con el fin de detectar y registrar nuevos incidentes que se vayan publicando en los sitios mirror-h.org y zone-xsec.com de manera continua y actualizada. Durante estas pruebas se identificaron varios inconvenientes técnicos. En el caso de mirror-h.org, se observó que el sitio presenta pérdida de conexión o caídas frecuentes Fig 11, lo que impide una recolección de datos constante y dependemos de la disponibilidad del sitio.

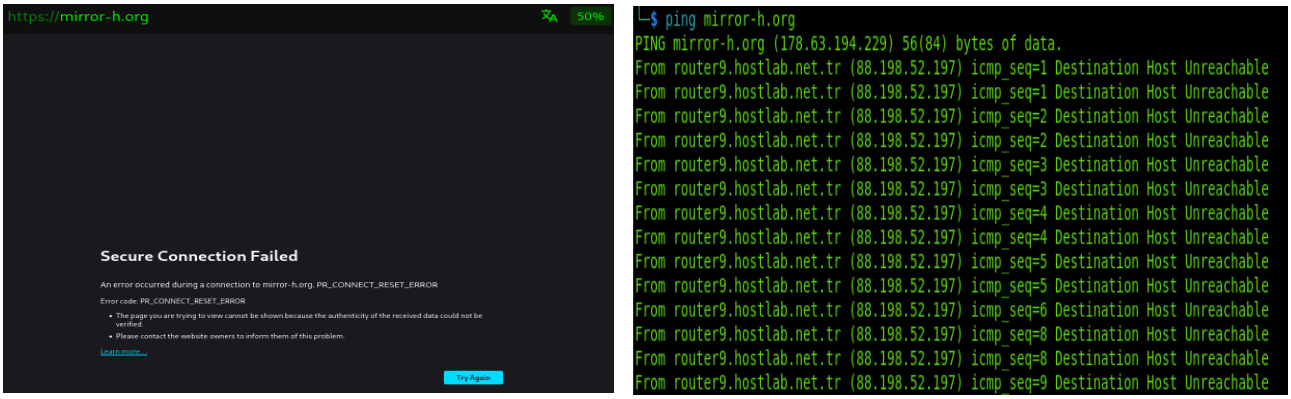


Fig 11: Evidencias indisponibilidad sitio mirror-h.org

Por otro lado, a partir del mes de febrero, el sitio zone-xsec.com implementó un sistema de protección contra ataques de denegación de servicio (DDoS), el cual impide el acceso automatizado mediante herramientas tradicionales como curl. Esta protección representa un obstáculo importante para la automatización de nuestro bot y de la recolección de datos.

De las Figuras 12 a la 14 se evidencia el flujo de las peticiones HTTP generadas al visitar el sitio zone-xsec.com. En dicho análisis se observa que la primera solicitud realizada nos lleva a un sistema de protección Anti-DDoS, el cual impone un control de acceso mediante cookies de autorización. Debido a que las solicitudes fueron ejecutadas mediante el comando cURL, estas no incluyen la lógica de navegación ni la ejecución de JavaScript necesarios para superar dicha verificación.

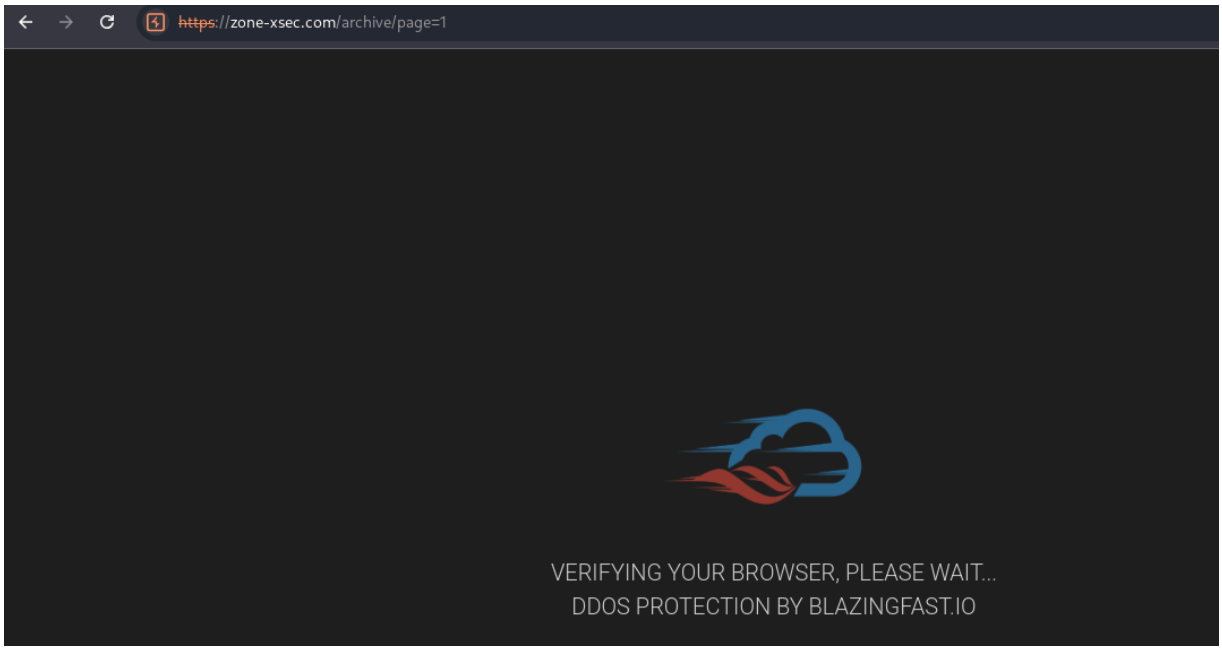


Fig 12: Front-End del sitio zone-xsec.com donde se visualiza su control anti DDoS

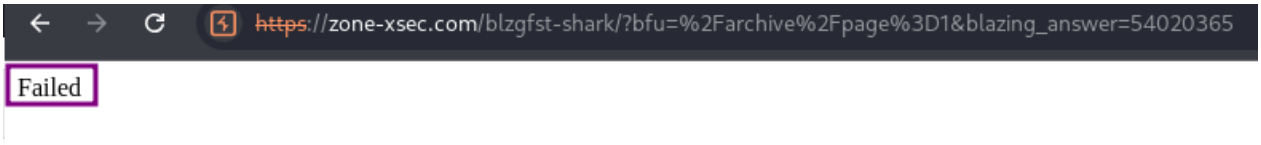


Fig 13: Prohibición de entrada al sitio zone-xsec.com

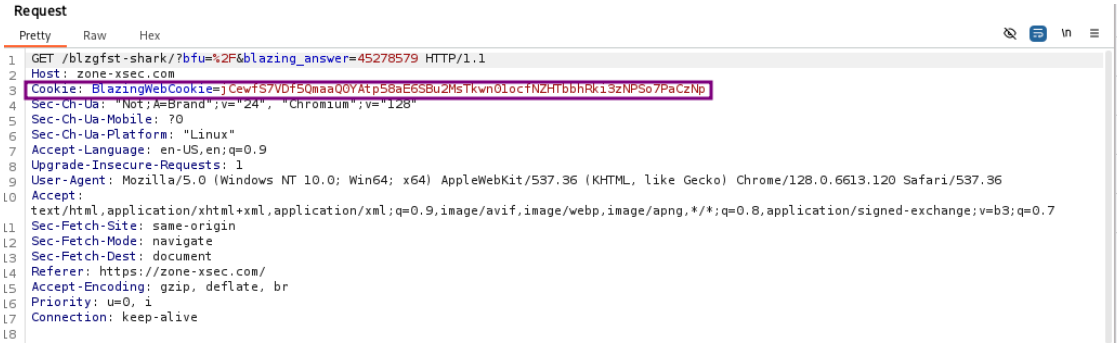


Fig 14: Cookie implementada por el sitio zone-xsec.com

Como mejora, se plantea la incorporación de herramientas especializadas en web scraping como Playwright, que permiten simular el comportamiento real de un navegador, incluyendo la ejecución de scripts y el manejo dinámico de cookies. Esto permitiría automatizar la obtención de cookies válidas, habilitando el acceso al sitio y la recopilación de datos, aún frente a las restricciones impuestas por mecanismos de protección contra ataques distribuidos de denegación de servicio (DDoS).

7. Conclusiones y recomendaciones

A lo largo del desarrollo del proyecto, se logró implementar exitosamente un módulo automatizado para la recolección de registros desde los sitios mirror-h.org y zone-xsec.com y notificación al webhook de discord. Este sistema demostró ser estable y funcional durante un período de observación de al menos dos meses, permitiendo la obtención continua de nuevos registros sin intervención manual.

Durante el transcurso del monitoreo se presentaron dos desafíos relevantes: el primero, asociado a la disponibilidad intermitente del sitio mirror-h.org, el cual en varias ocasiones dejó de responder y presentó caídas del servidor. El segundo, relacionado con la implementación de mecanismos de protección anti-bots por parte del sitio zone-xsec.com, que impiden el acceso a usuarios automatizados mediante herramientas y comandos como cURL. A pesar de estas limitaciones, se propone como mejora la incorporación de herramientas modernas de web scraping que simulan una navegación real del usuario, tales como Selenium o Playwright. Estas

tecnologías, ampliamente utilizadas en el estado del arte, permiten manejar cookies dinámicamente y evadir de forma controlada las restricciones impuestas por sistemas anti-DDoS o anti-bots, manteniendo así la operatividad del módulo.

Como recomendación final, se sugiere migrar completamente las solicitudes basadas en cURL hacia entornos controlados de automatización con navegadores, lo cual proporcionará mayor robustez, flexibilidad y resistencia frente a cambios en las políticas de acceso de los sitios monitoreados.

Referencias

[1] Candra Kurniawan, Agung Triayudi, "Reconstruction and Detection of Gambling Web Defacement Attack Using Wazuh and Velociraptor", 2024 International Conference on Information Technology Research and Innovation (ICITRI), pp.257-262, 2024.

[2] Xuan Dau Hoang and Ngoc Tuong Nguyen. 2019. A Multi-layer Model for Website Defacement Detection. In Proceedings of the 10th International Symposium on Information and Communication Technology (SoICT '19). Association for Computing Machinery, New York, NY, USA, 508–513. https://doi.org/10.1145/3368926.3369730

[3] Romagna, M., & van den Hout, N. J. (2017). Hacktivism and Website Defacement: Motivations, Capabilities and Potential Threats. Virus Bulletin Conference.

[4] Albalawi, M., Aloufi, R., Alamrani, N., Albalawi, N., Aljaedi, A., & Alharbi, A. R. (2022). Website Defacement Detection and Monitoring Methods: A Review. Electronics, 11(3573). https://doi.org/10.3390/electronics11213573

[5] C. Kurniawan and A. Triayudi, "Reconstruction and Detection of Gambling Web Defacement Attack Using Wazuh and Velociraptor," 2024 International Conference on Information Technology Research and Innovation (ICITRI), Jakarta, Indonesia, 2024, pp. 257-262, doi: 10.1109/ICITRI62858.2024.10699215.

[6] COLCERT AD-0709-021. Prevención de ataques tipo Defacement. Disponible en: https://www.colcert.gov.co/800/w3-article-396553.html

[7] {Sitio web: https://github.com/fastfire/deepdarkCTI/blob/main/defacement.md}

[8] {Sitio web: https://www.mirror-h.org/}

[9] {Sitio web: https://zone-xsec.com/}

[10] {Sitio web: https://www.zone-h.org/}

Anexos

Anexo 1. Poster

Departamento de Ingeniería Electrónica y de Telecomunicaciones

Sistema de Monitoreo OSINT (Open Source Intelligence) para la Detección y Notificación Temprana de Sitios Web Comprometidos.

UNIVERSIDAD DE ANTIOQUIA
Facultad de Ingeniería

PRACTICANTE: Jose Alejandro Serna Castañeda
ASESORES: Jaime Alberto Vergara - Francisco Muñoz

PROGRAMA: Ingeniería de Telecomunicaciones
Semestre de la práctica: 2025-1

Introducción

Los ataques a sitios web representan una amenaza constante para empresas de todos los sectores. Estos incidente generan pérdidas económicas y también afectan la reputación y la confianza de los clientes. Entre los diferentes tipos de ataques, el defacement es una técnica común utilizada por atacantes para modificar el contenido de los sitios web afectados, dejando mensajes visibles y exponiendo vulnerabilidades de seguridad. La detección temprana de estos ataques es crucial para mitigar sus efectos y evitar mayores impactos.

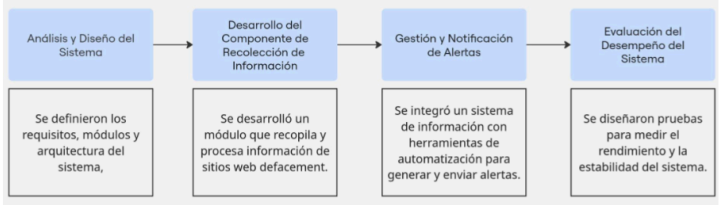
Como respuesta a esta problemática, se desarrolló un sistema automatizado capaz de monitorear sitios web especializados en reportar casos de defacement. El sistema analiza continuamente la información publicada y, en caso de detectar incidentes relacionados con sitios web de Colombia, emite una alerta.



Objetivos

- Desarrollar un sistema automático de detección y notificación de sitios web comprometidos, dirigido a empresas colombianas y clientes de ALIGO DEFENSORES INFORMÁTICOS SAS, que genere alertas tempranas.
- Definir los componentes y la estructura del sistema a implementar.
- Desarrollar un componente que detecte sitios web comprometidos en sitios de defacement.
- Implementar un sistema de información que se integre a la solución.
- Analizar y evaluar el desempeño del sistema.

Metodología

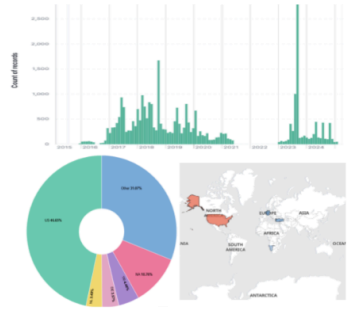


Conclusiones

- Se logró implementar exitosamente un módulo automatizado para la recolección de registros desde los sitios mirror-h.org y zone-xsec.com y notificación al webhook de discord. Este sistema demostró ser estable y funcional durante un período de observación de al menos dos meses, permitiendo la obtención continua de nuevos registros sin intervención manual.
- Se enfrentaron dos desafíos principales: la intermitencia del sitio mirror-h.org, con caídas frecuentes del servidor, y la protección anti-bots de zone-xsec.com, que bloquea accesos automatizados como los realizados con cURL.

Resultados

- La recolección de datos depende directamente de la disponibilidad de los sitios consultados. En el caso de mirror-h.org se evidenciaron caídas frecuentes, mientras que zone-xsec.com implementó barreras de acceso automatizado. Estas limitaciones afectan la continuidad del monitoreo y destacan la necesidad de adaptar las herramientas a entornos más restrictivos.



Historico de datos recopilados y paises con mas ataques reportados.

DATOS DE CONTACTO DEL AUTOR:
jose.serna@aligo.com.co josea.serna@udea.edu.co https://www.linkedin.com/in/joseaserna/