Análisis y descubrimiento de vulnerabilidades en múltiples superficies de ataque en entornos de seguridad informática

Carlos Daniel Quiros Carvajal

Informe final de práctica empresarial para optar por el título de Ingeniero de sistemas

Asesor
Deisy Loaiza Berrio, Ingeniera de sistemas

Universidad de Antioquia
Facultad de ingeniería
Ingeniería de sistemas
Medellín
2024

Dedicatoria

A mi madre, fuente inagotable de apoyo y sabiduría, cuyo amor y aliento han sido mi mayor impulso en este viaje académico.

A mis profesores, quienes con paciencia y conocimiento han iluminado el camino de mi formación como Ingeniero de Sistemas.

A mis amigos y compañeros de clase, por compartir risas, desafíos y éxitos a lo largo de esta travesía universitaria.

Este trabajo está dedicado a todos aquellos que creyeron en mí y que han sido parte fundamental en mi búsqueda constante de conocimiento. Optar por el título de Ingeniero de Sistemas no solo representa un logro personal, sino un compromiso continuo con la excelencia y la innovación en el mundo de la tecnología.

¡A seguir construyendo juntos el futuro digital!

Agradecimientos

En el cierre de este significativo capítulo académico que representa la obtención del título de Ingeniero de Sistemas, deseo expresar mi profundo agradecimiento a quienes han sido pilares fundamentales en este viaje.

A mi familia, mi mayor fuente de apoyo y motivación, gracias por creer en mí y alentarme a alcanzar mis metas. Su constante respaldo ha sido esencial para superar los desafíos y celebrar los triunfos.

A mis respetados profesores, quienes con su conocimiento experto y dedicación han guiado mi formación académica. Cada lección impartida ha sido un valioso aporte a mi crecimiento como profesional de la ingeniería de sistemas.

A mis amigos y compañeros de estudios, compartimos risas, desafíos y aprendizajes. Gracias por ser parte integral de esta travesía, por el apoyo mutuo y por los momentos que han enriquecido mi experiencia universitaria.

A la empresa "Fluid Attacks", quiero expresar mi más sincero agradecimiento por brindarme la oportunidad de realizar mis prácticas profesionales en un entorno tan dinámico y desafiante. La experiencia adquirida en sus proyectos de ciberseguridad ha sido invaluable para mi desarrollo profesional y para comprender la aplicación práctica de los conocimientos teóricos adquiridos en la universidad.

A todos aquellos que, de una manera u otra, han contribuido a mi crecimiento y éxito, les doy las gracias. Este logro no solo es mío, sino de una comunidad que ha creído en mí y ha sido parte esencial de mi trayectoria académica.

Tabla de contenido

LISTA DE TABLAS

LISTA DE FIGURAS

SIGLAS, ACRÓNIMOS Y ABREVIATURAS

ToE.	Target of Evaluation
URL	Uniform Resource Locator
SQL.	Structured Query Language
XSS	Cross-site scripting
API	Application Programming Interfaces

RESUMEN

Este proyecto abordó la creciente preocupación por la seguridad informática en la sociedad moderna, con un enfoque específico en la identificación de vulnerabilidades en diversas superficies de ataque. Fue desarrollado por un estudiante de Ingeniería de Sistemas y practicante en la empresa de ciberseguridad "Fluid Attacks", quien enfrentó el desafío de analizar 28 proyectos, cada uno con su propio conjunto de código y aplicativo web. La metodología integral que se utilizó incluyó la investigación exhaustiva, el análisis de código en varios lenguajes de programación y pruebas de penetración en aplicativos web. Se documentaron las vulnerabilidades encontradas con detalles precisos, respaldando las conclusiones con evidencias visuales como capturas de pantalla o vídeos. El propósito era contribuir significativamente a la seguridad informática, fortaleciendo la protección de sistemas y salvaguardando la información en un entorno digital expuesto a amenazas. Se logró evaluar correctamente la severidad de las vulnerabilidades y proporcionar recomendaciones claras para su mitigación, cumpliendo con los estándares y normativas pertinentes.

Palabras clave *— Seguridad Informática, Vulnerabilidades, Ingeniería de Sistemas, Fluid Attacks, Análisis de Código, Pruebas de Penetración, Mitigación de Riesgos, Ciberseguridad, Protección de Datos, Normativas de Seguridad.*

ABSTRACT

This project addressed the growing concern for computer security in modern society, with a specific focus on identifying vulnerabilities in various attack surfaces. It was developed by a Systems Engineering student and intern at the cybersecurity company "Fluid Attacks", who faced the challenge of analyzing 28 projects, each with its own set of code and web application. The comprehensive methodology used included extensive research, code analysis in various programming languages and penetration testing of web applications. The vulnerabilities found were documented in precise detail, supporting the findings with visual evidence such as screenshots or videos. The purpose was to contribute significantly to computer security, strengthening system protection and safeguarding information in a digital environment exposed to threats. We were able to correctly assess the severity of vulnerabilities and provide clear recommendations for their mitigation, complying with the relevant standards and regulations.

Keywords *— Cybersecurity, Vulnerabilities, Systems Engineering, Fluid Attacks, Code Analysis, Penetration Testing, Risk Mitigation, Data Protection, Cybersecurity Regulation.*

I. INTRODUCCIÓN

En el contexto de la creciente dependencia de la sociedad moderna en la tecnología, se observaron numerosos avances y beneficios, pero también surgieron nuevos desafíos en términos de seguridad informática. En este escenario, el proyecto tuvo como objetivo abordar una problemática clave en el campo de la seguridad informática: identificar vulnerabilidades existentes en diferentes superficies de ataque.

Como estudiante de ingeniería de sistemas y practicante en una empresa especializada en seguridad informática, se ha comprendido la importancia crítica de garantizar la integridad, confidencialidad y disponibilidad de la información en entornos digitales. En el rol de Security Tester, se llevaban a cabo análisis exhaustivos de código en diversos lenguajes de programación y de aplicativos (webs y móviles), con el fin de detectar y reportar vulnerabilidades que pudieran ser explotadas por actores maliciosos.

En este proyecto, se enfrentó a un desafío particular: el análisis de 28 proyectos, cada uno de ellos con su propio conjunto de código y aplicativo web. Esta diversidad implicaba que, en ocasiones, los aplicativos web no estaban disponibles para su evaluación. No obstante, se aprovecharon todas las oportunidades para examinar tanto el código fuente como los aplicativos web, en busca de vulnerabilidades que pudieran poner en riesgo la seguridad de los sistemas y los datos.

El objetivo principal de este proyecto fue desarrollar un enfoque integral y riguroso para la identificación de vulnerabilidades en múltiples superficies de ataque. A través de un análisis exhaustivo, se documentaron las vulnerabilidades encontradas, indicando la línea exacta en caso de hallarse en el código, o proporcionando la URL y el parámetro vulnerable si se trataba de un aplicativo web. Además, se añadieron evidencias visuales, como imágenes o videos, que respaldan las conclusiones y facilitan la comprensión de las vulnerabilidades detectadas.

En resumen, el proyecto académico tuvo como objetivo brindar una contribución significativa a la seguridad informática mediante la identificación y documentación de vulnerabilidades en diversas superficies de ataque. El trabajo realizado busca fortalecer la protección de los sistemas y salvaguardar la información sensible en un entorno digital cada vez más complejo y expuesto a amenazas.

II. OBJETIVOS

A. Objetivo general

Identificar vulnerabilidades existentes en diferentes superficies de ataque para mejorar la seguridad informática de 28 proyectos de empresas externas.

B. Objetivos específicos

- Recopilar información sobre los vectores de ataque más comunes en código y aplicaciones web.

- Realizar análisis exhaustivos de código en diversos lenguajes de programación para identificar posibles vulnerabilidades.

- Realizar pruebas de penetración en los aplicativos web disponibles para identificar vulnerabilidades y brechas de seguridad.

- Documentar y reportar las vulnerabilidades encontradas, proporcionando información detallada sobre su impacto y severidad.

- Proponer recomendaciones y medidas de seguridad que permitan la mitigación de las vulnerabilidades encontradas y el fortalecimiento de la seguridad informática de los sistemas y aplicaciones web.

III. MARCO TEÓRICO

En la construcción del marco teórico para el análisis de vulnerabilidades, se basó en diversas páginas y entes reguladores reconocidos en el campo de la seguridad informática. Estos recursos ofrecen pautas, mejores prácticas y estándares para la identificación, mitigación y prevención de vulnerabilidades en sistemas y aplicaciones web. A continuación, se describirán brevemente algunos de los más relevantes:

CIS[1] (Center for Internet Security): Una organización sin fines de lucro que proporciona benchmarks y guías de configuración segura para sistemas y aplicativos.

HIPAA[2] (Health Insurance Portability and Accountability Act): Una legislación estadounidense que establece estándares de seguridad y privacidad para la información de salud.

ISO/IEC 27001 y ISO/IEC 27002[3] Normas internacionales para la gestión de la seguridad de la información y los controles de seguridad.

MITRE ATT&CK®[4] Un conocido modelo de matriz de adversarios que describe tácticas, técnicas y procedimientos utilizados por actores maliciosos en ciberataques.

NIST[5] (National Institute of Standards and Technology): Un organismo estadounidense que ha publicado una amplia gama de estándares y guías de seguridad informática, incluyendo el NIST Framework for Improving Critical Infrastructure Cybersecurity, NIST 800-53, NIST 800-171 y NIST 800-115.

OWASP[7] (Open Web Application Security Project): Una organización líder en seguridad de aplicaciones web, que ha desarrollado guías y listas de las 10 principales vulnerabilidades (OWASP Top 10) y otros proyectos específicos como el OWASP ASVS (Application Security

Verification Standard), OWASP API Security Top 10 y OWASP MASVS (Mobile Application Security Verification Standard).

PCI DSS[6] (Payment Card Industry Data Security Standard): Un estándar de seguridad de la industria de tarjetas de pago que establece requisitos para proteger la información de las tarjetas de crédito.

OWASP Top 10 Privacy Risks[7] Un proyecto que se enfoca en las principales preocupaciones de privacidad en las aplicaciones web y proporciona recomendaciones para abordarlas.

SQL Injection Is Still Alive: A Study on SQL Injection Signature Evasion Techniques[8] En este artículo, se describe la naturaleza de los ataques de inyección SQL, se analizan técnicas actuales de evasión de detección y se propone una combinación de soluciones para mitigar el riesgo de estos ataques.

Profiling Database Application to Detect SQL Injection Attacks[9] En este artículo, se presenta un marco novedoso basado en técnicas de detección de anomalías para contrarrestar amenazas a las bases de datos internas de una organización provenientes de aplicaciones de bases de datos.

MACE: Detecting Privilege Escalation Vulnerabilities in Web Applications[10] En este artículo, Se aborda la detección de escalada de privilegios no autorizados en aplicaciones web, debido a autorizaciones incorrectas en el código del servidor

Identifying cross site scripting vulnerabilities in Web applications[11] Este artículo presenta un enfoque que combina análisis estático y dinámico para evaluar la vulnerabilidad XSS en aplicaciones web.

Preventing Input Validation Vulnerabilities in Web Applications through Automated Type Analysis[12] este artículo aborda las vulnerabilidades comunes en aplicaciones web, como el cross-site scripting y la inyección SQL, qué son explotadas por atacantes.

IV. METODOLOGÍA

La metodología propuesta para la búsqueda de vulnerabilidades se basó en una serie de actividades secuenciales, diseñadas para alcanzar los objetivos planteados. A continuación, se presenta la estructura de esta metodología:

1. Investigación y recopilación de información:
    - Realizar una revisión exhaustiva de las fuentes de información relacionadas con superficies de ataque, estándares de seguridad y buenas prácticas.
    - Analizar y sintetizar la información recopilada para identificar los tipos de vulnerabilidades más relevantes.
2. Análisis de código:
    - Realizar análisis estáticos y dinámicos de código en diferentes lenguajes de programación de forma manual y también utilizando herramientas especializadas.
    - Identificar vulnerabilidades comunes, como inyecciones SQL, cross-site scripting (XSS), entre otros, y documentar su ubicación y severidad.
3. Pruebas de penetración en aplicativos web:
    - Realizar pruebas de penetración utilizando técnicas y herramientas especializadas para identificar vulnerabilidades en los aplicativos web disponibles.
    - Evaluar la seguridad de las autenticaciones, autorizaciones, manejo de sesiones, entre otros aspectos críticos.
4. Documentación y reporte de vulnerabilidades:
    - Analizar las vulnerabilidades encontradas y proponer recomendaciones prácticas para su solución o mitigación.
    - Generar informes claros y detallados, que incluyan evidencias visuales, como capturas de pantalla o vídeos, para respaldar los hallazgos.
5. Propuesta de recomendaciones y medidas de seguridad:

- Elaborar informes detallados sobre las vulnerabilidades encontradas, siguiendo los estándares y formatos establecidos por la empresa y los entes reguladores mencionados en el marco teórico.
- Incluir información precisa sobre la ubicación de las vulnerabilidades en el código fuente o en el aplicativo web (URL y parámetros).
- Agregar evidencias visuales de las vulnerabilidades encontradas, como capturas de pantalla, videos o cualquier otro medio que respalde las conclusiones.

V. RESULTADOS

El análisis de vulnerabilidades fue desarrollado por el practicante en la empresa Fluid Attacks en 28 proyectos, donde trabajó con los objetivos y metodologías definidos en este documento.

Los resultados de los análisis de vulnerabilidades y las correcciones de las brechas de seguridad encontradas no se comparten en este documento. No obstante, se detalla el proceso general y los datos más relevantes. Proporcionar detalles más específicos sobre las vulnerabilidades encontradas en el análisis podría vulnerar las políticas de mantener la confidencialidad y proteger datos sensibles de las empresas dueñas de los proyectos.

Para realizar el análisis de vulnerabilidades del código fuente y las aplicaciones web, con el propósito de examinar aspectos críticos en la infraestructura de las plataformas de las Tecnologías de las empresas y detectar posibles falencias, se utilizaron las siguientes herramientas:

- **BeEF**: Browser Exploitation Framework, una herramienta de pruebas de penetración centrada en el navegador web.
- **Burp Suite Professional:** Conjunto de herramientas para automatizar, encontrar y ayudar a descubrir y explotar vulnerabilidades web.
- **Dbeaver**: Herramienta multiplataforma para la gestión de bases de datos
- **enumerate-iam**: Intenta forzar todas las llamadas API permitidas por la política IAM. Las llamadas realizadas por esta herramienta son todas no destructivas (solo se realizan llamadas get y list).
- **ffuf:** Fuzzer web rápido
- **Gitleaks:** Herramienta de código abierto para detectar secretos y datos sensibles en repositorios Git
- **hashcat:** Herramienta de pirateo rápida, eficaz y versátil que ayuda a realizar ataques de fuerza bruta sin conexión.
- **Nmap:** Utilidad de detección de redes y auditoría de seguridad

Cada herramienta seleccionada para el análisis desempeñó un papel clave en la identificación de vulnerabilidades. Desde el Browser Exploitation Framework (BeEF), que se centró en exploits a nivel de navegador, hasta el conjunto de herramientas integral Burp Suite Professional, que automatizó la detección y explotación de vulnerabilidades web, cada elección reflejó la sofisticación y el alcance del análisis llevado a cabo.

La inclusión de Dbeaver para la gestión de bases de datos, enumerate-iam para la evaluación de políticas IAM en la nube, ffuf como un fuzzer web rápido y eficiente, Gitleaks para la detección de secretos en repositorios Git, hashcat para la evaluación de contraseñas, y Nmap para la detección de redes, ilustra la diversidad de enfoques aplicados para abordar las múltiples capas de seguridad presentes en los proyectos evaluados.

Durante los meses dedicados al proyecto de análisis de vulnerabilidades en Fluid Attacks, se logró validar un impresionante Alcance del Análisis (ToE, por sus siglas en inglés) combinado de todos los proyectos, alcanzando la cifra significativa de 1284.36. Esta métrica única amalgama no solo las líneas de código meticulosamente verificadas en diversos lenguajes de programación, sino también la exhaustiva evaluación de inputs provenientes de aplicativos web. Este ToE integral encapsula la complejidad y amplitud de los proyectos evaluados, resaltando la profundidad del compromiso hacia la seguridad informática. La validación de este ToE no solo representa un hito cuantitativo, sino que también subraya la comprensividad del enfoque adoptado para garantizar la integridad y resistencia de los sistemas frente a amenazas cibernéticas en constante evolución.

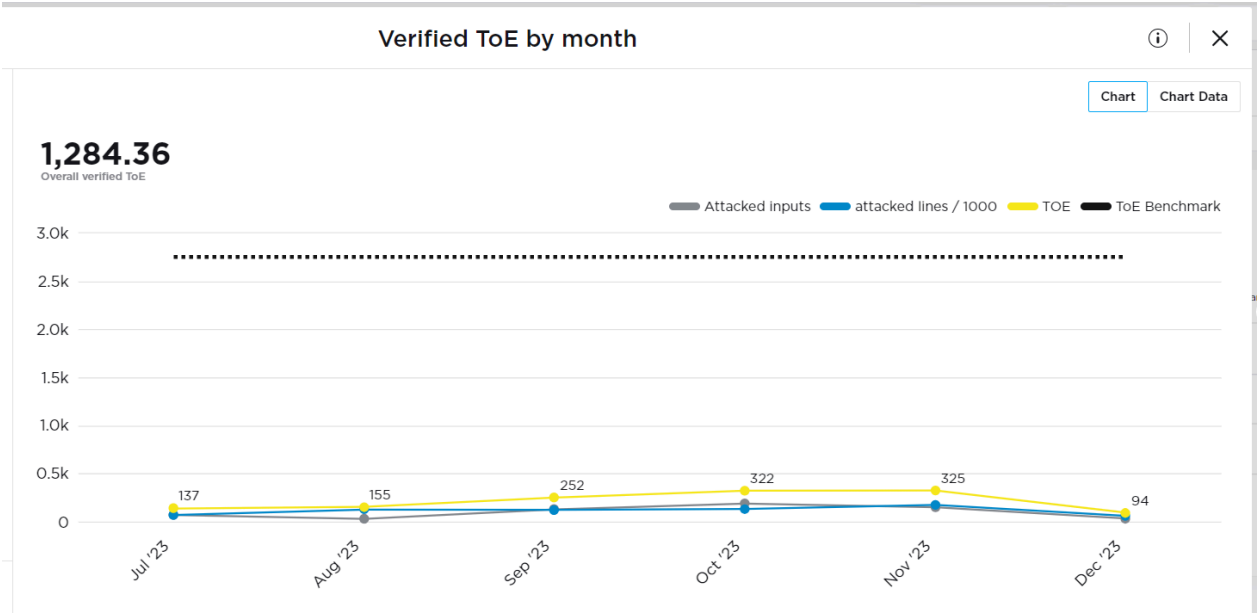


Fig. 1. Total ToE verificado

Durante el meticuloso análisis de los proyectos, se revelaron múltiples hallazgos que arrojan luz sobre la seguridad informática en las superficies de ataque evaluadas. Algunos de los descubrimientos más relevantes incluyen aquellas vulnerabilidades distribuidas en la siguiente tabla (TABLA I)

TABLA I
CATEGORÍAS DE LAS VULNERABILIDADES

Categoría	Total	Definiciones
Subversión del acceso	258	Acción de manipular o eludir los mecanismos de acceso autorizado a sistemas o datos, permitiendo entrada no autorizada
Manipulación de datos	0	Alteración no autorizada de la información almacenada, con el objetivo de modificar, destruir o acceder a datos sensibles.
Interacciones engañosas	28	Uso de tácticas fraudulentas para inducir a errores en las interacciones con sistemas, engañando a usuarios o sistemas automáticos.
Abuso de funcionalidad	604	Utilización inapropiada o maliciosa de las características y funciones legítimas de un sistema para propósitos no autorizados.
Recopilación de información	3634	Proceso de recoger datos, a menudo de manera no autorizada, con el objetivo de obtener información sensible o confidencial.
Técnicas probabilísticas	0	Empleo de métodos basados en probabilidades para explotar vulnerabilidades, aprovechando la imprevisibilidad en la toma de decisiones.
Manipulación de protocolos	61	Modificación no autorizada de los protocolos de comunicación utilizados entre sistemas para comprometer la seguridad de la transmisión de datos.
Manipulación de sistemas	0	Acción de alterar el funcionamiento normal de sistemas informáticos con el propósito de obtener acceso no autorizado o causar daño.
Inyección inesperada	326	Introducción no autorizada e inesperada de datos o comandos maliciosos en un sistema, a menudo a través de entradas de usuario, para comprometer su integridad.

En los cuales es importante resaltar que el total de vulnerabilidades encontradas entre las diferentes categorías asciende a un total de 4911

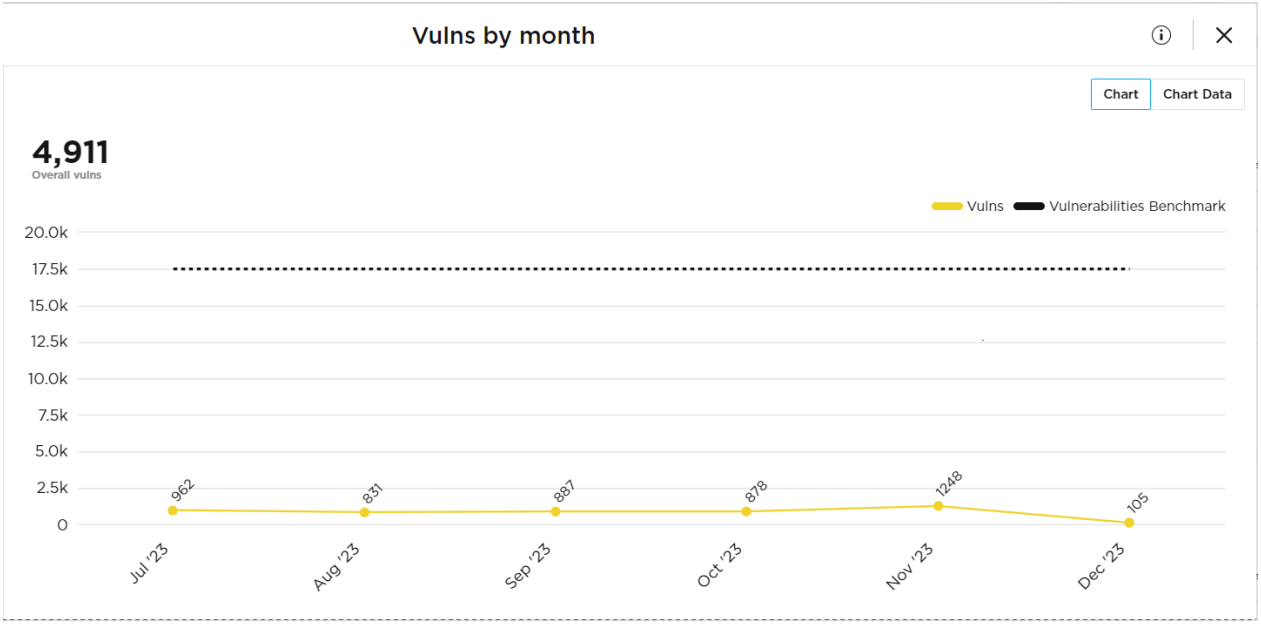
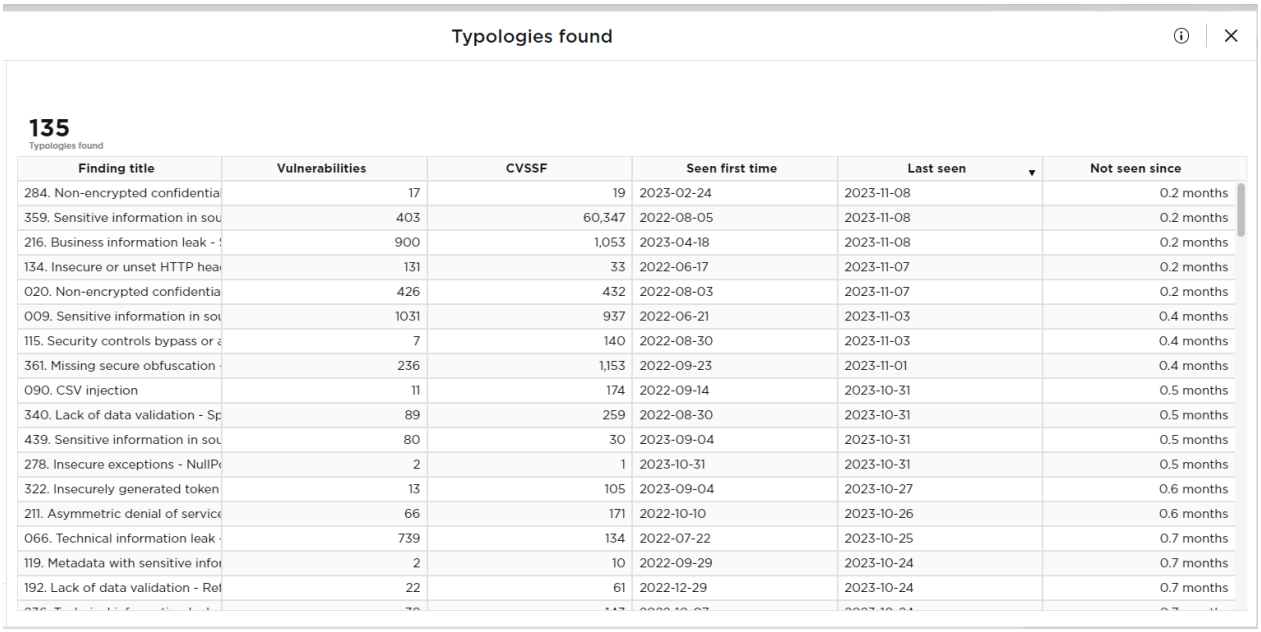


Fig. 2. Total de vulnerabilidades encontradas

El análisis meticuloso reveló un total de 4911 vulnerabilidades, distribuidas en categorías cruciales como subversión del acceso, manipulación de datos, interacciones engañosas y más. Este conjunto diverso de debilidades destaca la complejidad y la variedad de amenazas presentes en los proyectos examinados. Además, la identificación de 135 tipologías diferentes subraya la necesidad de un enfoque holístico y adaptativo para abordar los desafíos específicos de seguridad en cada proyecto.



Finding title	Vulnerabilities	CVSSF	Seen first time	Last seen	Not seen since
284. Non-encrypted confidentia	17	19	2023-02-24	2023-11-08	0.2 months
359. Sensitive information in sou	403	60,347	2022-08-05	2023-11-08	0.2 months
216. Business information leak -	900	1,053	2023-04-18	2023-11-08	0.2 months
134. Insecure or unset HTTP hea	131	33	2022-06-17	2023-11-07	0.2 months
020. Non-encrypted confidentia	426	432	2022-08-03	2023-11-07	0.2 months
009. Sensitive information in sou	1031	937	2022-06-21	2023-11-03	0.4 months
115. Security controls bypass or a	7	140	2022-08-30	2023-11-03	0.4 months
361. Missing secure obfuscation	236	1,153	2022-09-23	2023-11-01	0.4 months
090. CSV injection	11	174	2022-09-14	2023-10-31	0.5 months
340. Lack of data validation - Sp	89	259	2022-08-30	2023-10-31	0.5 months
439. Sensitive information in sou	80	30	2023-09-04	2023-10-31	0.5 months
278. Insecure exceptions - NullP	2	1	2023-10-31	2023-10-31	0.5 months
322. Insecurely generated token	13	105	2023-09-04	2023-10-27	0.6 months
211. Asymmetric denial of service	66	171	2022-10-10	2023-10-26	0.6 months
066. Technical information leak	739	134	2022-07-22	2023-10-25	0.7 months
119. Metadata with sensitive infor	2	10	2022-09-29	2023-10-24	0.7 months
192. Lack of data validation - Ref	22	61	2022-12-29	2023-10-24	0.7 months

Fig. 4. Total de las tipologías encontradas

Las implicaciones derivadas de estos hallazgos subrayan la urgencia de tomar acciones correctivas y preventivas. Aunque los detalles específicos de las vulnerabilidades no se divulgan por razones de confidencialidad, se insta a las organizaciones dueñas de los proyectos a considerar las recomendaciones generales proporcionadas en el marco teórico y, específicamente, las detalladas en los informes de vulnerabilidades.

Durante la ejecución del análisis de vulnerabilidades en los 28 proyectos de la empresa Fluid Attacks, se encontraron diversas dificultades que presentaron desafíos significativos en el proceso. Uno de los eventos más recurrentes fue la presencia de páginas web o ambientes y APIs no funcionales. Este escenario comprometió la capacidad de evaluación de vulnerabilidades en tiempo real, requiriendo un enfoque más proactivo para la identificación y corrección de posibles amenazas, cambiando el enfoque al análisis de código estático.

Otro desafío crucial que se manifestó como eventos recurrentes fue la presencia de credenciales vencidas de los usuarios de pruebas. La obsolescencia de las credenciales afectó la integridad de las pruebas de penetración y limitó la capacidad de simular escenarios realistas de

amenazas. La gestión efectiva de las credenciales se volvió imperativa para garantizar la validez y relevancia de los resultados obtenidos.

Adicionalmente, se encontraron flujos de la aplicación que no funcionaban correctamente, introduciendo eventos que complicaron la evaluación de la seguridad. Estos fallos operativos no solo afectaron la eficiencia del análisis, sino que también resaltaron la importancia de evaluar no solo la seguridad técnica, sino también la funcionalidad general de las aplicaciones en términos de resistencia a amenazas.

Finalmente, los repositorios de código que no se clonaban adecuadamente fue otro desafío significativo. Este evento impactó directamente en la capacidad de realizar análisis estáticos de código y resaltó la necesidad de abordar problemas operativos para garantizar un análisis de vulnerabilidades integral. Estas dificultades, agrupadas bajo el término "eventos," subrayan la complejidad inherente a la evaluación de la seguridad en entornos dinámicos y la necesidad de estrategias adaptativas para superar obstáculos operativos y técnicos.

VI. ANÁLISIS

El análisis de los resultados obtenidos durante la ejecución de la propuesta de prácticas en la empresa Fluid Attacks revela una panorámica profunda y desafiante de la ciberseguridad en el entorno digital actual. La complejidad de las vulnerabilidades identificadas, sumada a la magnitud de las tipologías descubiertas, plantea interrogantes significativas sobre la seguridad de los proyectos analizados.

La selección cuidadosa de herramientas de vanguardia para llevar a cabo el análisis no solo refleja la diligencia de la empresa, sino también la necesidad de enfoques variados para abordar las distintas capas de seguridad en sistemas y aplicaciones web. El uso de BeEF, Burp Suite Professional, Dbeaver, y otras herramientas especializadas demuestra la sofisticación técnica aplicada en la evaluación de vulnerabilidades, señalando una conciencia profunda de las amenazas potenciales.

La identificación de 4911 vulnerabilidades, distribuidas en las diferentes categorías, subraya la fragilidad inherente en los proyectos evaluados. Estos hallazgos plantean preguntas cruciales sobre la efectividad de las medidas de seguridad implementadas y destaca la necesidad de una revisión exhaustiva de las políticas de seguridad existentes.

La amalgama de vulnerabilidades descubiertas en los diversos proyectos evaluados ha generado un impacto total de CVSSF (Cumulative Vulnerability Severity Score) de 727,000. Esta métrica, concebida por Fluid Attacks, proporciona una representación precisa del nivel de exposición al riesgo que presentan las vulnerabilidades identificadas en el sistema. Con una escala que abarca desde 0.015625 hasta 4096, el CVSSF permite realizar un análisis agregado de las vulnerabilidades, proporcionando así una medida integral del riesgo de seguridad. Este puntaje combinado refleja la magnitud de la amenaza cibernética que podría afectar la integridad y operatividad de los sistemas, reforzando la necesidad de estrategias de mitigación y medidas preventivas para salvaguardar la seguridad informática.

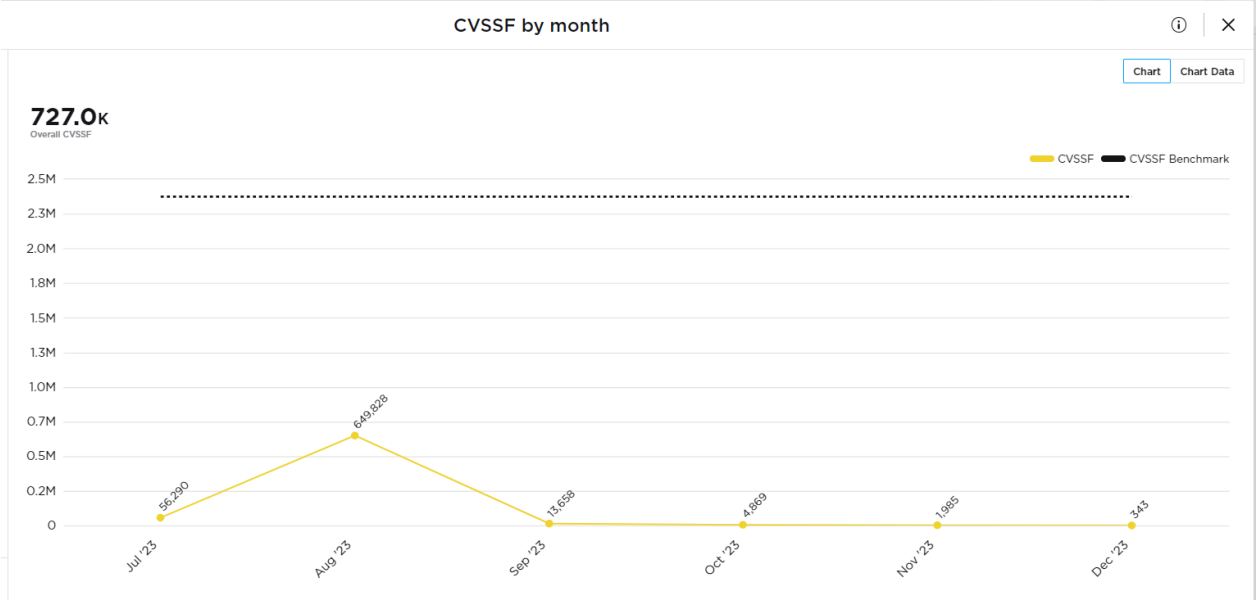


Fig. 4. Total de CVSSF reportado

La clasificación de estas vulnerabilidades en 135 tipologías diferentes revela la diversidad de los desafíos de seguridad presentes. Cada tipología, representando una amenaza única, demanda una atención específica y soluciones personalizadas. Este nivel de detalle en la clasificación refleja la meticulosidad del análisis y destaca la complejidad intrínseca de garantizar una seguridad informática sólida.

La decisión estratégica de no divulgar detalles específicos de las vulnerabilidades, respetando la confidencialidad y la protección de datos sensibles, refuerza el compromiso ético de Fluid Attacks. Sin embargo, esta elección plantea la pregunta sobre cómo equilibrar la transparencia con la protección de la información sensible en el ámbito de la ciberseguridad.

En última instancia, los resultados obtenidos resaltan la urgencia de acciones correctivas y preventivas. Las organizaciones dueñas de los proyectos deben considerar no solo las recomendaciones generales proporcionadas en el marco teórico, sino también las sugerencias específicas detalladas en los informes de vulnerabilidades. La ciberseguridad ya no es simplemente una precaución; se ha convertido en una necesidad imperativa en la era digital actual. Adoptar una postura proactiva, integrar prácticas sólidas y adaptarse continuamente son

los pilares esenciales para salvaguardar la integridad de sistemas y datos en un entorno digital en constante evolución.

VII. CONCLUSIONES

El exhaustivo análisis de 28 proyectos en la empresa Fluid Attacks ha revelado hallazgos significativos que impactan directamente en la ciberseguridad y la integridad de sistemas críticos. Las conclusiones extraídas reflejan una evaluación ponderada de los objetivos establecidos, proporcionando percepciones cruciales para el fortalecimiento de la seguridad informática.

La identificación de 4911 vulnerabilidades, distribuidas en categorías clave como subversión del acceso, manipulación de datos e interacciones engañosas, destaca la amplitud de desafíos a los que se enfrentan los proyectos evaluados. Este análisis va más allá de la cuantificación numérica, revelando vulnerabilidades específicas que demandan atención prioritaria y estrategias de mitigación efectivas.

La diversidad de 135 tipologías diferentes de vulnerabilidades dentro de estas categorías amplifica la complejidad inherente a la seguridad informática. Cada tipología representa una dimensión única de amenaza, exigiendo soluciones específicas y un enfoque adaptativo. Este nivel de detalle proporciona una guía estratégica valiosa para la implementación de medidas de seguridad personalizadas.

El conjunto de herramientas avanzadas seleccionadas, desde BeEF hasta Burp Suite Professional, demuestra la necesidad de una estrategia diversificada en la evaluación de vulnerabilidades. La combinación de estas herramientas resalta la importancia de una aproximación integral que abarque tanto los aspectos de navegador web como las vulnerabilidades específicas de aplicaciones.

La decisión ética de no divulgar detalles específicos de las vulnerabilidades, salvaguardando la confidencialidad y protegiendo datos sensibles, refleja el compromiso de Fluid Attacks con prácticas responsables en ciberseguridad. Esta elección resalta la importancia crítica de equilibrar la transparencia con la seguridad de la información, un desafío constante en el paisaje digital actual.

En síntesis, este análisis no solo logra los objetivos iniciales, sino que también destaca áreas clave para la mejora de la seguridad informática. Las 4911 vulnerabilidades y las 135 tipologías identificadas proporcionan una base sólida para estrategias de corrección y prevención. Estas conclusiones ofrecen una guía estratégica que va más allá de las métricas numéricas, enfatizando la necesidad de adoptar enfoques adaptables y proactivos para garantizar la integridad de sistemas y datos en un entorno digital dinámico.

VIII. RECOMENDACIONES

El análisis exhaustivo de vulnerabilidades en los 28 proyectos ejecutados en la empresa Fluid Attacks ha proporcionado una base sólida para futuras investigaciones y líneas de estudio en el ámbito de la ciberseguridad. Las recomendaciones ofrecidas a continuación delinean áreas clave que podrían abordarse para fortalecer aún más la seguridad informática en entornos digitales complejos:

Exploración de Nuevas Técnicas de Detección: Investigar y desarrollar nuevas técnicas de detección de vulnerabilidades que puedan abordar específicamente las amenazas emergentes y las tácticas de los actores maliciosos en evolución.

Estudio Detallado de Tipologías Específicas: Realizar análisis más detallados de las 135 tipologías de vulnerabilidades identificadas, profundizando en la comprensión de su funcionamiento y enfoques específicos de mitigación.

Integración de Inteligencia Artificial y Aprendizaje Automático: Investigar cómo las tecnologías de inteligencia artificial y aprendizaje automático pueden ser implementadas para mejorar la detección proactiva de vulnerabilidades, adaptándose a patrones de amenazas cambiantes.

Desarrollo de Herramientas Especializadas: Crear y perfeccionar herramientas especializadas para la evaluación de vulnerabilidades en entornos específicos, como aplicaciones móviles, dispositivos IoT y sistemas de nube.

Evaluación de Impacto en la Arquitectura de Red: Profundizar en la evaluación del impacto de vulnerabilidades en la arquitectura de red, considerando no solo las aplicaciones específicas sino también la infraestructura subyacente.

Estudio de Amenazas en Entornos de Desarrollo Ágil: Analizar las vulnerabilidades específicas que pueden surgir en entornos de desarrollo ágil, donde los cambios rápidos pueden introducir inadvertidamente nuevas amenazas.

Colaboración con Comunidades de Investigación: Fomentar la colaboración con comunidades de investigación en ciberseguridad para compartir conocimientos, identificar nuevas tendencias y enfrentar desafíos comunes de manera colaborativa.

Desarrollo de Estrategias de Mitigación Basadas en Riesgos: Investigar y desarrollar estrategias de mitigación basadas en riesgos que prioricen las vulnerabilidades de acuerdo con su impacto potencial en la seguridad y operatividad del sistema.

Evaluación Continua de Buenas Prácticas: Realizar evaluaciones periódicas de las buenas prácticas de seguridad informática y adaptarlas a medida que evolucionen las amenazas y las tecnologías.

Estudios sobre Políticas de Divulgación: Investigar y desarrollar políticas de divulgación efectivas que equilibren la transparencia con la protección de datos sensibles, permitiendo compartir conocimientos sin comprometer la seguridad.

Estas recomendaciones representan áreas estratégicas para futuras investigaciones, donde la innovación y la adaptabilidad son esenciales. Al abordar estas líneas de estudio, se puede avanzar significativamente en la comprensión y mitigación de amenazas cibernéticas en constante evolución.

REFERENCIAS

[1] CIS. [Consultado el 23, junio, 2023]. https://www.cisecurity.org/

[2] HIPAA FOR Professionals. HHS.gov. https://www.hhs.gov/hipaa/for-professionals/index.html .

[3] ISO/IEC 27001 Standard – Information Security Management Systems. ISO. https://www.iso.org/standard/27001 .

[4] MITRE ATT&CK® [Anónimo]. MITRE ATT&CK®.: https://attack.mitre.org/ .

[5] NATIONAL INSTITUTE of Standards and Technology. NIST. https://www.nist.gov/

[6] OFFICIAL PCI Security Standards Council Site. PCI Security Standards Council. https://www.pcisecuritystandards.org/ .

[7] OWASP FOUNDATION, the Open-Source Foundation for Application Security | OWASP Foundation. OWASP Foundation, the Open-Source Foundation for Application Security | OWASP Foundation. https://owasp.org/.

[8] A. Sadeghian, M. Zamani and S. Ibrahim, "SQL Injection Is Still Alive: A Study on SQL Injection Signature Evasion Techniques," in 2013 International Conference on Informatics and Creative Multimedia (ICICM), Kuala Lumpur, Malaysia, 2013 pp. 265-268.

[9] J. Early, A. Kamra and E. Bertino, "Profiling Database Application to Detect SQL Injection Attacks," in Performance, Computing, and Communications Conference, 2002. 21st IEEE International, New Orleans, LA, USA, 2007 pp. 449-458.

[10] Maliheh Monshizadeh, Prasad Naldurg, and V. N. Venkatakrishnan. 2014. MACE: Detecting Privilege Escalation Vulnerabilities in Web Applications.

[11] G. A. Di Lucca, A. R. Fasolino, M. Mastoianni and P. Tramontana, "Identifying cross site scripting vulnerabilities in Web applications," Proceedings. Sixth IEEE International Workshop on Web Site Evolution, Chicago, IL, USA, 2004, pp. 71-80, doi: 10.1109/WSE.2004.10013.

[12] T. Scholte, W. Robertson, D. Balzarotti and E. Kirda, "Preventing Input Validation Vulnerabilities in Web Applications through Automated Type Analysis," 2012 IEEE 36th Annual Computer Software and Applications Conference, Izmir, Turkey, 2012, pp. 233-243, doi: 10.1109/COMPSAC.2012.34.