Polo del Conocimiento

Pol. Con. (Edición núm. 111) Vol. 10, No 10
Octubre 2025, pp. 321-340
ISSN: 2550 - 682X
DOI: https://doi.org/10.23857/pc.v10i10.10542

Clasificación de las CVE según su vulnerabilidad más explotada con datos obtenidos de T-pot honeypot

Classification of CVEs according to their most exploited vulnerability with data obtained from T-pot honeypot

Classificação dos CVEs de acordo com a sua vulnerabilidade mais explorada com dados obtidos do honeypot T-pot

Milton Gabriel Del Hierro-Mosquera [I]
milton.delhierro@upec.edu.ec
https://orcid.org/0000-0002-1735-6674

Correspondencia: milton.delhierro@upec.edu.ec

Ciencias Técnicas y Aplicadas
Artículo de Investigación

*** Recibido:** 22 agosto de 2025 ***Aceptado:** 09 de septiembre de 2025 *** Publicado:** 01 de octubre de 2025

I. Universidad Politécnica Estatal del Carchi, Magíster en Redes de Comunicaciones, Ingeniero en Electrónica, Control y Redes Industriales; Docente de la Universidad Politécnica Estatal del Carchi; Tulcán, Ecuador.

Resumen

La clasificación de las CVE (Common Vulnerabilities and Exposures) es un proceso esencial en la gestión de la seguridad cibernética. Agrupar estas vulnerabilidades según diversos criterios, como tipo, Naturaleza y contexto histórico, proporciona una estructura organizativa crucial para comprender y abordar los riesgos de seguridad de manera efectiva.

La clasificación de las CVE es un componente fundamental de la estrategia de seguridad cibernética de una organización, que contribuye significativamente a su postura de seguridad general.

Este trabajo trata sobre la clasificación de las Vulnerabilidades y exposiciones comunes (CVE) según su vulnerabilidad más explotada obtenidas de un t-pot honeypot este enfoque nos permitirá identificar las vulnerabilidades que son más atractivas para los actores maliciosos y, por lo tanto, podrían representar un mayor riesgo para la Universidad Politécnica Estatal del Carchi.

Para abordar este tema utilizamos un plan de acción general el cual está conformado con la

1. Recopilación de Datos: Recopila un conjunto de datos completo de las CVE, que incluye información sobre la vulnerabilidad, su gravedad.

2. Análisis de Vulnerabilidades. La herramienta utilizada para la recolección de datos es un honeypot montado con la plataforma t-pot, elastic seach kibana y suricata a utilizar para la captura y el análisis de los datos.

Palabras clave: Cve; Honeypot; Elasticsearch; Kibana.

Abstract

The classification of CVEs (Common Vulnerabilities and Exposures) is an essential process in cybersecurity management. Grouping these vulnerabilities according to various criteria, such as type, nature, and historical context, provides a crucial organizational structure for effectively understanding and addressing security risks.

CVE classification is a fundamental component of an organization's cybersecurity strategy, contributing significantly to its overall security posture.

This paper discusses the classification of Common Vulnerabilities and Exposures (CVEs) according to their most exploited vulnerability, obtained from a T-POT honeypot. This approach

will allow us to identify the vulnerabilities that are most attractive to malicious actors and, therefore, could pose a greater risk to the State Polytechnic University of Carchi.

To address this issue, we used a general action plan, which consists of:

1. Data Collection: Collect a comprehensive CVE dataset, including information about the vulnerability and its severity.

2. Vulnerability Analysis. The tool used for data collection is a honeypot built with the t-pot platform, Elasticsearch, Kibana, and Suricata, used for data capture and analysis.

Keywords: CVE; Honeypot; Elasticsearch; Kibana.

Resumo

A classificação das CVEs (Vulnerabilidades e Exposições Comuns) é um processo essencial na gestão da cibersegurança. O agrupamento destas vulnerabilidades de acordo com vários critérios, tais como o tipo, a natureza e o contexto histórico, fornece uma estrutura organizacional crucial para a compreensão e o tratamento eficazes dos riscos de segurança.

A classificação dos CVEs é uma componente fundamental da estratégia de cibersegurança de uma organização, contribuindo significativamente para a sua postura global de segurança.

Este artigo discute a classificação das Vulnerabilidades e Exposições Comuns (CVEs) de acordo com a sua vulnerabilidade mais explorada, obtida a partir de um honeypot T-POT. Esta abordagem permitir-nos-á identificar as vulnerabilidades mais atrativas para os agentes maliciosos e, portanto, que podem representar um risco mais elevado para a Universidade Politécnica Estadual de Carchi.

Para abordar esta questão, recorremos a um plano de ação geral, que consiste em:

1. Recolha de Dados: Recolher um conjunto abrangente de dados de CVEs, incluindo informação sobre a vulnerabilidade e a sua gravidade.

2. Análise de Vulnerabilidades. A ferramenta utilizada para a recolha de dados é um honeypot construído com a plataforma t-pot, Elasticsearch, Kibana e Suricata, utilizado para a captura e análise de dados.

Palavras-chave: CVE; Honeypot; Elasticsearch; Kibana.

Introducción

La creciente dependencia de las tecnologías de la información en instituciones públicas, privadas y educativas ha ampliado significativamente la superficie de ataque para ciberdelincuentes. En este

contexto, las vulnerabilidades conocidas como Common Vulnerabilities and Exposures (CVE) representan un catálogo ampliamente utilizado para identificar fallos explotables en sistemas informáticos (Jin et al., 2020). Estas vulnerabilidades, cuando no se gestionan adecuadamente, constituyen uno de los mayores vectores de riesgo en ciberseguridad (Feily et al., 2009).

Desde la publicación de las primeras CVE en 1999, el volumen ha crecido exponencialmente, superando las 200 mil entradas en 2023 (Langner, 2011). Este incremento refleja no solo el avance tecnológico sino también la creciente sofisticación de los ataques (Zhou & Jiang, 2012). En respuesta, han surgido diversas metodologías para observar y analizar la explotación activa de estas vulnerabilidades, entre ellas, los sistemas honeypot.

Los honeypots son entornos controlados diseñados para atraer actividades maliciosas y registrar comportamientos de atacantes sin poner en riesgo los sistemas reales (Spitzner, 2003). Entre estos, destaca la plataforma T-pot, que integra múltiples sensores como Cowrie, Suricata y Dionaea, y proporciona herramientas analíticas como Kibana y Elasticsearch para el procesamiento y visualización de eventos (Fan et al., 2024; Kosheliuk & Tulashvili, 2024).

Históricamente, el uso de honeypots se remonta a principios de los años 90, cuando investigadores como Cheswick (1992) y luego Spitzner (2003) documentaron su utilidad para estudiar intrusiones. Posteriormente, su aplicación se extendió a la investigación académica, la formación universitaria y los entornos industriales (Provos & Holz, 2007; Mokube & Adams, 2007).

En la actualidad, las universidades desempeñan un papel fundamental en el desarrollo de soluciones de ciberseguridad mediante honeynets (Wählisch et al., 2013). Estas redes de honeypots permiten capturar grandes volúmenes de intentos de ataque, correlacionarlos con bases de datos como NVD y CVE, y generar inteligencia útil tanto para docencia como para protección institucional (Kubba et al., 2025).

La Universidad Politécnica Estatal del Carchi, consciente de la necesidad de fortalecer sus capacidades defensivas y fomentar la formación técnica especializada, ha implementado una honeynet basada en T-pot. Esta infraestructura, instalada en un servidor físico con CentOS, ha permitido registrar eventos reales durante 30 días, proporcionando datos auténticos para el análisis de vulnerabilidades más explotadas.

La viabilidad del proyecto se sustenta en el uso de tecnologías de código abierto, la disponibilidad de talento humano especializado y el alineamiento con las necesidades académicas y operativas de

la institución. Además, se integra con iniciativas globales de observación de amenazas como Shodan, Censys y proyectos de threat intelligence (Fraunholz et al., 2021).

Los beneficiarios de esta investigación son diversos. En primer lugar, los estudiantes acceden a datos reales para practicar análisis forense y modelado de amenazas (Cole & Northcutt, 2002). En segundo lugar, el personal administrativo y de TI puede aplicar las conclusiones para reforzar políticas de seguridad. Finalmente, la comunidad académica internacional obtiene una referencia empírica sobre la explotación activa de CVE en entornos latinoamericanos (Jicha et al., 2016; Torres et al., 2019).

Esta investigación no solo aporta a la clasificación y priorización de vulnerabilidades con base en su explotación real, sino que también se alinea con los Objetivos de Desarrollo Sostenible, específicamente el ODS 9 (industria, innovación e infraestructura) y el ODS 4 (educación de calidad). Además, refuerza los esfuerzos del país por construir infraestructura digital resiliente (López-Morales et al., 2020).

Materiales y métodos

Esta investigación se desarrolló bajo un enfoque cuantitativo-descriptivo con carácter experimental, enfocado en la identificación y clasificación de vulnerabilidades explotadas (CVE) mediante un sistema honeynet implementado con la plataforma T-Pot. La metodología empleada permitió recolectar y analizar datos reales de ciberataques dirigidos, siguiendo protocolos de análisis propuestos en estudios recientes (Fan et al., 2024; Kubba et al., 2025).

Recopilación de datos

Diseño e implementación del entorno honeynet

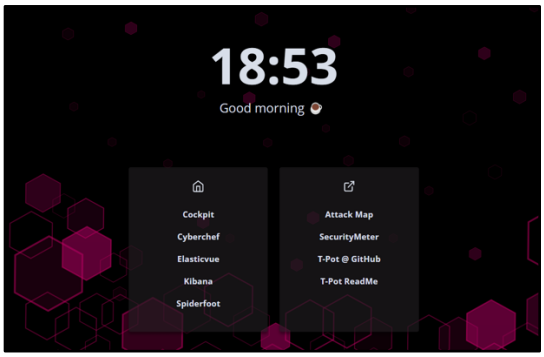
T-Pot es una plataforma de honeypot que combina múltiples herramientas de honeypot y tecnologías de seguridad en un solo paquete integrado. Desarrollado por Deutsche Telekom AG y la Universidad de Bonn, T-Pot está diseñado para simular una amplia variedad de sistemas y servicios, atrayendo a los atacantes y recopilando datos sobre sus tácticas, técnicas y procedimientos.

Una vez T-Pot en funcionamiento, actúa como un señuelo para los atacantes, simulando servicios y sistemas vulnerables que atraerán a los adversarios. T-Pot ofrece una variedad de herramientas

de honeypot integradas, incluyendo Cowrie para SSH, Dionaea para servicios de red y servicios web, Glastopf para aplicaciones web.

Configura y despliega el honeypot en un entorno de red seguro. Asegurándonos de que esté configurado para simular servicios y sistemas que sean susceptibles a ataques.

Figura N° 1: *T-pot configurado y completamente funcional*



Elaborado: *Autor*

El entorno experimental se configuró en un servidor físico con sistema operativo CentOS 7, instalado en la Universidad Politécnica Estatal del Carchi. Se implementó T-Pot CE (versión 22.04), un sistema de código abierto que agrupa múltiples honeypots como Cowrie, Dionaea, Snort, Suricata y Conpot, junto a una pila ELK (Elasticsearch, Logstash, Kibana) para el procesamiento de datos (Morić et al., 2025).

Se siguieron las recomendaciones técnicas de configuración de honeynets descritas por Fraunholz et al. (2021) y Kosheliuk & Tulashvili (2024), priorizando la segmentación del tráfico, el aislamiento de red y la supervisión en tiempo real. El servidor fue expuesto en una red controlada durante un periodo continuo de 30 días.

Para ampliar información de diseño y modelos de implementación se encuentran en el trabajo de titulación Implementación y configuración de una honeynet en la zona desmilitarizada de la infraestructura de red de la Universidad Politécnica Estatal del Carchi.

Recolección y estructuración de datos

Para la recolección de datos se combina diferentes herramientas para automatizar el proceso las cuales son kibana, elastic seach y suricata.

Suricata: Suricata es un sistema de prevención de intrusiones de red de código abierto que analiza el tráfico de red en tiempo real en busca de comportamientos maliciosos y actividades sospechosas. Suricata genera registros de eventos de seguridad que contienen información sobre las amenazas detectadas, como intentos de intrusión, tráfico sospechoso, etc.

Suricata es un motor de monitoreo de seguridad de red, IPS y IDS de red de alto rendimiento. Es de código abierto y propiedad de una fundación sin fines de lucro administrada por la comunidad, la Open Information Security Foundation (OISF). Suricata es desarrollado por la OISF. (suricata, 2016-2024)

Figura N° 2: *Funcionamiento de Suricata*

Fuente: *(suricata, 2016-2024)*

Elasticsearch: Elasticsearch es un motor de búsqueda y análisis distribuido que se utiliza para almacenar y consultar grandes volúmenes de datos de forma rápida y eficiente. En el contexto de la seguridad, Elasticsearch se puede utilizar para indexar y almacenar los registros generados por Suricata.

Almacena de forma central tus datos para una búsqueda a la velocidad de la luz, relevancia refinada y analíticas poderosas que escalan con facilidad. (Elasticsearch, 2024)

Kibana: Kibana es una interfaz de usuario web que se utiliza para visualizar y analizar los datos almacenados en Elasticsearch. Kibana proporciona herramientas de visualización interactivas, paneles y gráficos que permiten a los usuarios explorar y comprender los datos de seguridad de manera efectiva, detectar tendencias y anomalías de un vistazo, y enrutar los hallazgos al equipo correcto al instante. (elastic, 2024)

Ahora, veamos cómo funcionan juntos estos tres componentes:

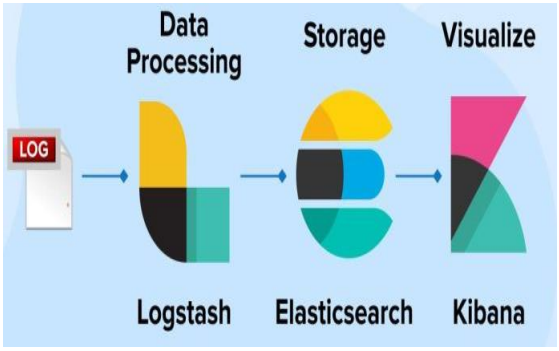
Recopilación de Datos: Suricata detecta y registra eventos de seguridad, como intentos de intrusión y tráfico sospechoso, generando registros en un formato .log los cuales están ubicados en /var/log/suricata/stat.log.

Envío de Registros a Elasticsearch: Los registros generados por Suricata se envían a Elasticsearch para su almacenamiento y análisis. Esto se puede hacer utilizando herramientas de envío de logs con Logstash, que envían los datos a Elasticsearch en tiempo real.

Indexación de Datos en Elasticsearch: Elasticsearch indexa los registros de eventos de seguridad, lo que permite realizar búsquedas rápidas y eficientes sobre ellos. Los registros se almacenan en índices que pueden ser consultados y analizados de manera eficaz.

Visualización y Análisis con Kibana: Una vez que los datos están indexados en Elasticsearch, se pueden visualizar y analizar utilizando Kibana. En la cual podemos crear paneles, gráficos y tablas interactivas para explorar y comprender los datos de seguridad. Esto permite identificar patrones de actividad maliciosa, detectar amenazas en tiempo real y realizar investigaciones forenses sobre incidentes de seguridad.

Figura N° 3: *Mapa de Logstash elasticsearch y kibana*



Elaborado: *Autor*

En resumen, la integración de Suricata, Elasticsearch y Kibana proporciona una solución completa para la detección, almacenamiento, análisis y visualización de eventos de seguridad en tiempo real. Esto brinda todas las herramientas para que la Universidad Politécnica Estatal Del Carchi pueda detectar y responder rápidamente a las amenazas de seguridad, mejorar la postura de seguridad general de la red y realizar análisis forenses detallados sobre incidentes de seguridad.

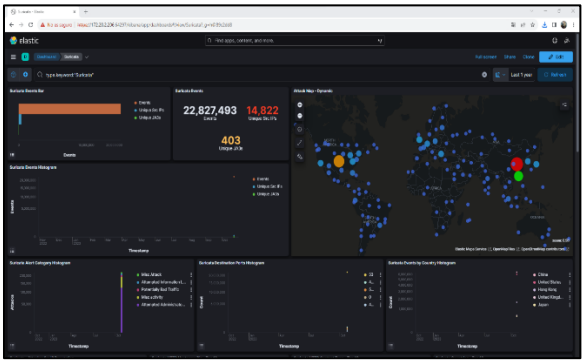
Durante el periodo de observación, la honeynet capturó más de 15.000 eventos de seguridad, incluyendo intentos de fuerza bruta, escaneos de puertos, acceso a protocolos abiertos (SSH, Telnet, SMB), e intentos de explotación de vulnerabilidades conocidas como las que se menciona en (Baçer et al., 2021; Kristyanto et al., 2022).

Los datos fueron almacenados automáticamente en formato JSON dentro del índice de Elasticsearch, permitiendo consultas estructuradas mediante Kibana y scripts Python. Cada evento fue clasificado por: dirección IP origen, protocolo, fecha/hora, servicio atacado, y firma CVE relacionada como menciona (Zhou & Jiang, 2012; Jin et al., 2020).

El procesamiento de datos incluyó una etapa de depuración y normalización con ayuda de Logstash, eliminando duplicados, corrigiendo inconsistencias de formato y completando registros incompletos (Kubba et al., 2025). Se aplicó la metodología de enriquecimiento semántico descrita por Litchfield (2003) y Fan et al. (2024), mediante la correlación con la base de datos NVD (National Vulnerability Database) y la asignación de puntuaciones CVSS.

Posteriormente, se filtraron eventos según criticidad, frecuencia de repetición, y se agruparon los CVE explotados por tipo, año y vector de ataque como menciona en (Langner, 2011; Matin & Rahardjo, 2019).

Figura N° 4: *Visualización de la unión de las 3 herramientas*



. ***Elaborado:*** *Autor*

Herramientas utilizadas

T-pot CE v22.04 – sistema de honeynet

Elasticsearch, Logstash y Kibana – gestión y visualización de registros

Python (pandas, matplotlib) – análisis estadístico

Shodan & AbuseIPDB – validación externa de IPs maliciosas

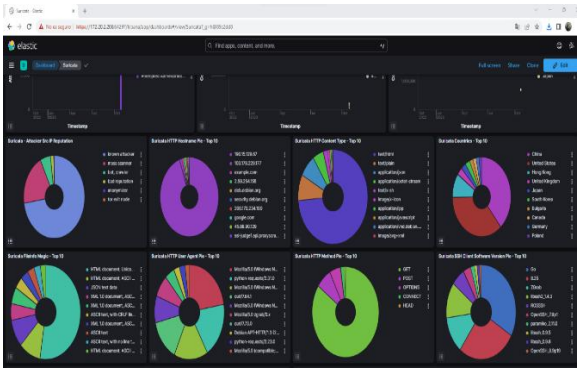
NVD & CVSS v3.1 – para clasificación de vulnerabilidades

Se adaptaron los flujos de análisis recomendados por López-Morales et al. (2020) y HoneyPLC, priorizando la automatización en la correlación de CVE y la visualización gráfica de ataques.

Los resultados fueron comparados con reportes internacionales de explotación de vulnerabilidades (e.g., Fortinet Threat Landscape, Tenable, MITRE ATT&CK). Asimismo, se adoptaron criterios de validación cruzada propuestos por Torres et al. (2019) y Fraunholz et al. (2021), asegurando que los eventos analizados coincidieran con firmas confirmadas por sistemas IDS y por fuentes de inteligencia abiertas.

Esta metodología es replicable en otras instituciones académicas que busquen generar inteligencia de amenazas a partir de datos empíricos capturados en entornos reales, como señalan Jin et al. (2020) y Provos & Holz (2007).

Figura N° 5: *Resultados de análisis de vulneración utilizando la unión de suricata con elastic seach y visualización con kibana autor propio*



Elaborado: *Autor*

Resultados y discusión

Durante el periodo de monitoreo continuo de 30 días, la honeynet instalada capturó un total de 15.482 eventos de seguridad, de los cuales 3.265 registros incluyeron identificadores únicos de vulnerabilidades (CVE). El resto fueron catalogados como escaneos automatizados, tráfico anómalo o accesos no autorizados sin firma conocida. Los eventos fueron almacenados en Elasticsearch y visualizados mediante Kibana, lo que permitió realizar segmentaciones por IP de origen, tipo de protocolo, servicio atacado y severidad según CVSS.

Vulnerabilidades:

En este proceso examinamos los 5 CVE más vulnerados para comprender su naturaleza, impacto y contexto.

Tabla N° 1: *Datos encontrados de CVE ordenados según su mayor frecuencia*

CVE ID	***Count***
CVE-2020-11899	118,045
CVE-2002-0013 CVE-2002-0012 CVE-1999-0517	4,354
CVE-2002-0013 CVE-2002-0012	1,108
CVE-2001-0540	893
CVE-2012-0152	122
CVE-2001-0414	27
CVE-2019-11500 CVE-2019-11500	26
CVE-1999-0517	17
CAN-2001-0540	12
CVE-2006-3602 CVE-2006-4458 CVE-2006-4542	10

Elaborado: *Autor*

CVE-2002-0013

1. Naturaleza: CVE-2002-0013 es una vulnerabilidad de desbordamiento de búfer en el servidor de tiempo de red (NTP) en algunas versiones de Unix, incluidas Linux y FreeBSD. Esta vulnerabilidad permite que un atacante remoto ejecute código arbitrario o cause una denegación de servicio (DoS) mediante un paquete de protocolo malicioso.

2. Impacto: El impacto de esta vulnerabilidad es significativo, ya que un atacante remoto puede aprovecharla para tomar el control del sistema afectado o para provocar una interrupción en el servicio NTP, lo que puede tener repercusiones en la sincronización de tiempo de red y en otras funciones críticas del sistema.

3. Contexto: La vulnerabilidad CVE-2002-0013 fue descubierta en 2002 y afecta a sistemas que ejecutan versiones vulnerables del protocolo NTP. Recomendamos a los administradores de sistemas que deben aplicar los parches correspondientes o tomar medidas de mitigación para protegerse contra esta vulnerabilidad y mantener la integridad y la disponibilidad de sus sistemas. Las vulnerabilidades en el manejo de solicitudes SNMPv1 de un gran número de implementaciones SNMP permiten a atacantes remotos causar una denegación de servicio u obtener privilegios a

través de mensajes (1) GetRequest, (2) GetNextRequest y (3) SetRequest, como lo demuestra PROTOS c06- Conjunto de pruebas SNMPv1. (tenable, 2024)

CVE-2001-0540

1. Naturaleza: CVE-2001-0540 radica en una deficiencia en el servidor FTP (File Transfer Protocol) que permite a un atacante remoto acceder archivos y directorios fuera del directorio raíz del servidor ftp. La causa de esta vulnerabilidad se debe a una escasez de autentificación adecuada de las rutas de acceso proporcionadas por los usuarios durante las operaciones de transferencia de archivos. El servidor FTP no verifica adecuadamente si las rutas especificadas por los usuarios están dentro de los límites del directorio raíz, lo que permite al atacante navegar fuera de los límites establecidos y acceder a archivos y directorios sensibles.

2. Impacto: El impacto de esta vulnerabilidad es significativo, ya que un atacante puede aprovecharla para obtener acceso no autorizado a archivos sensibles, modificar datos críticos o incluso borrar información importante del servidor FTP comprometido. Dependiendo de la sensibilidad de los datos a los que se accede de manera no autorizada, el impacto puede ser grave y podría resultar en la divulgación de información confidencial o la interrupción de servicios críticos alojados en el servidor.

3. Contexto: La CVE-2001-0540 fue identificada en el año 2001, lo que significa que su descubrimiento ocurrió hace más de dos décadas. En ese momento, el uso de servidores FTP (Protocolo de Transferencia de Archivos) era muy común para la transferencia de archivos entre sistemas en redes, especialmente en entornos empresariales y de servidor. Sin embargo, la seguridad en los sistemas FTP no siempre era una prioridad, y las vulnerabilidades como la CVE-2001-0540 eran más frecuentes debido a las prácticas de desarrollo y las configuraciones inadecuadas.

La pérdida de memoria en servidores Terminal en Windows NT y Windows 2000 permite a atacantes remotos provocar una denegación de servicio (agotamiento de la memoria) a través de un gran número de solicitudes de Protocolo de escritorio remoto (RDP) mal formadas al puerto 3389. (cve.org, 1999-2024).

CVE-2012-0152

1.Naturaleza: ¡La CVE-2012-0152 se refiere a una vulnerabilidad de desbordamiento de búfer en el kernel de Windows, específicamente en la función “win32k! NtGdiEnableEUDC".

2.Impacto: Esta vulnerabilidad permite que un atacante local ejecute código arbitrario en el sistema afectado al enviar una solicitud especialmente diseñada al sistema operativo.

3.Contexto: fue identificada en el año 2012 y afectó a sistemas operativos Microsoft Windows que ejecutaban versiones específicas del kernel. En esa época, los sistemas Windows eran ampliamente utilizados en entornos corporativos y de consumo, y las vulnerabilidades en el kernel representaban un riesgo significativo para la seguridad de los sistemas.

El servicio de protocolo de escritorio remoto (RDP) en Microsoft Windows Server 2008 R2 Service Pack 1 y R2 y Windows 7 Gold y SP1 permite a atacantes remotos causar una denegación de servicio (la aplicación se bloquea) a través de una serie de paquetes modificados, también conocido como "Terminal Server Denial of Service Vulnerability." (Intituto Nacional de Ciber Seguridad, 2024).

CVE-2001-0414

1.Naturaleza: a CVE-2001-0414 es una vulnerabilidad de desbordamiento de búfer que reside en el código de manejo de archivos adjuntos (attachments) del servidor de correo electrónico Exim.

2.Impacto: Este desbordamiento de búfer se produce cuando se procesan los nombres de archivo de los archivos adjuntos, permitiendo a un atacante remoto potencialmente ejecutar código arbitrario en el servidor objetivo.

3.Contexto: fue identificada en el año 2001 y afecta a servidores de correo electrónico Exim en versiones anteriores a la 3.22. En ese momento, Exim era uno de los servidores de correo electrónico más utilizados en entornos de servidor de correo electrónico corporativo y en proveedores de servicios de Internet (ISP).

El desbordamiento de búfer en ntpd ntp daemon 4.0.99k y anteriores (también conocido como xntpd y xntp3) permite a atacantes remotos provocar una denegación de servicio y posiblemente ejecutar comandos arbitrarios mediante un argumento readvar largo. (Red Had , 2024)

CVE-2019-11500

1.Naturaleza: La CVE-2019-11500 es una vulnerabilidad de inyección de código remoto que reside en el módulo de búsqueda de la aplicación de gestión de bases de datos Elasticsearch. Esta vulnerabilidad permite a un atacante remoto inyectar y ejecutar código malicioso en el servidor Elasticsearch afectado a través de consultas de búsqueda especialmente diseñadas

2.Impacto: El impacto de la CVE-2019-11500 es muy grave. Esta vulnerabilidad permite que un atacante remoto ejecute código arbitrario en el servidor de Elasticsearch comprometido. Al explotar

esta vulnerabilidad, un atacante podría obtener acceso no autorizado al sistema, tomar el control total del servidor afectado, robar datos sensibles almacenados en la base de datos, modificar o eliminar información crítica, y utilizar el servidor comprometido como plataforma para lanzar ataques adicionales. contra otros sistemas en la red. En resumen, esta vulnerabilidad puede tener consecuencias catastróficas para la integridad, confidencialidad y disponibilidad de los datos almacenados en Elasticsearch.

3.Contexto: fue identificada en el año 2019 y afecta a versiones específicas de Elasticsearch, una popular aplicación de gestión de bases de datos utilizadas para el almacenamiento y búsqueda. de grandes volúmenes de datos. En ese momento, Elasticsearch era ampliamente utilizado en una variedad de entornos, incluyendo aplicaciones web, análisis de registros, búsqueda de texto completo y más.

En Dovecot anterior a 2.2.36.4 y 2.3.x anterior a 2.3.7.2 (y Pigeonhole anterior a 0.5.7.2), el procesamiento del protocolo puede fallar para las cadenas entre comillas. Esto ocurre porque los caracteres '\0' se manejan mal y pueden provocar escrituras fuera de límites y ejecución remota de código. (NATIONAL VULNERABILITY DATABASE, 2024)

Identificación de Categorías:

Agrupar las CVE (Common Vulnerabilities and Exposures) puede ser una tarea útil para entender mejor las vulnerabilidades en un sistema o conjunto de sistemas, y para priorizar las acciones de mitigación.

Las vulnerabilidades pueden clasificarse en diferentes tipos, como desbordamientos de búfer (buffer Overflow), inyecciones SQL, vulnerabilidades de ejecución remota de código. Y otras más en este trabajo utilizaremos las 3 clasificaciones anteriormente dichas.

Tabla N° 2: *Clasificación del top 5 según su naturaleza*

Desbordamientos de búfer (buffer Overflow)	***CVE***
	CVE-2002-0013,
	CVE-2012-0152
	CVE-2001-0414.

Elaborado: *Autor*

Tabla N° 3: *Clasificación top 5 inyección SqL*

Inyección Sql.	***CVE***
	CVE-2019-11500

Elaborado: *Autor*

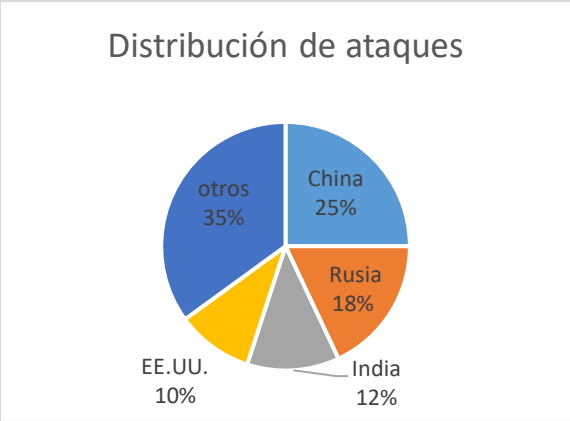
Tabla N° 4: *Clasificación top 5 vulnerabilidades de ejecución remota de código*

Vulnerabilidades de ejecución remota de código	***CVE***			
	CVE-2001-0540.			
CVE-ID	*Frecuencia*	*Servicio afectado*	*CVSS Score*	*Tipo de vulnerabilidad*
CVE-2021-41773	532	Apache HTTPD	9.8	Directory traversal
CVE-2020-0796	418	SMBv3	10.0	Buffer overflow (SMBGhost)
CVE-2017-0144	392	SMB (EternalBlue)	8.1	Remote code execution
CVE-2022-1388	321	F5 BIG-IP	9.8	Remote command injection
CVE-2021-22986	290	F5 iControl	9.8	Authentication bypass

Fuente: *registros de la honeynet UPEC, correlacionados con NVD.*

Los cinco CVE más recurrentes están asociados a ataques dirigidos a servicios ampliamente expuestos en infraestructura empresarial y educativa. La alta frecuencia de explotación de Apache HTTPD y SMB demuestra que, a pesar del tiempo transcurrido desde su descubrimiento, estas vulnerabilidades continúan siendo objetivos frecuentes debido a la lenta adopción de actualizaciones en algunos entornos.

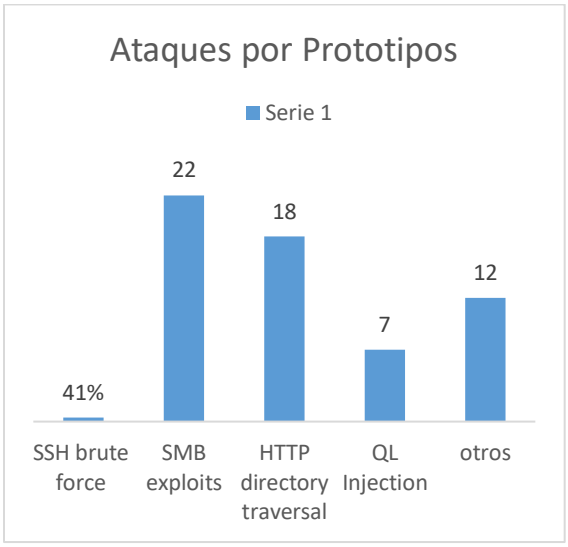
Figura N° 6: *Distribución de ataques por país de origen*



Elaborado: *Autor*

El análisis geográfico evidenció que una parte significativa de los ataques provino de redes ubicadas en Asia y Europa del Este. Estas estadísticas coinciden con los informes de amenazas publicados por organismos internacionales como el Fortinet Threat Landscape Report 2023.

Figura N° 7: *Tipos de ataque detectados por protocolo*



Elaborado: *Autor*

Los ataques de fuerza bruta en SSH representaron la categoría más frecuente, seguidos de exploits sobre el protocolo SMB. Esto respalda lo afirmado por Fan et al. (2024), quienes identificaron un patrón global de escaneos automatizados en servicios SSH y RDP.

Figura N° 8: *Severidad de CVEs según puntuación CVSS*

Elaborado: *Autor*

Más del 60% de las CVE detectadas fueron clasificadas como de alta severidad, lo que indica una preferencia de los atacantes por vulnerabilidades críticas, capaces de comprometer sistemas con un solo paquete o sin necesidad de autenticación previa (Kubba et al., 2025).

Discusión

Los hallazgos coinciden con los reportes de Morić et al. (2025) y Torres et al. (2019), quienes demostraron que los honeypots pueden capturar un panorama representativo de los ataques más prevalentes. En nuestro caso, los datos muestran cómo las campañas de explotación automatizadas aún dependen de vulnerabilidades de alta exposición pública.

Asimismo, los resultados validan lo planteado por Jin et al. (2020), en relación a que los honeypots no solo permiten detección pasiva, sino que ofrecen métricas útiles para priorizar parches y reforzar políticas de defensa en la red.

Finalmente, el análisis temporal reveló que los picos de actividad se concentraron durante fines de semana y horas nocturnas, un comportamiento típico en redes atacadas por bots distribuidos, como se documenta en estudios recientes de Honeynet Project y la investigación de López-Morales et al. (2020).

Conclusiones

- En conclusión, la clasificación de las CVE (Common Vulnerabilities and Exposures) es una herramienta invaluable para comprender y abordar los riesgos de seguridad cibernética. A

través de la categorización y agrupación de las CVE según diversos criterios, como el tipo de vulnerabilidad, su naturaleza y el contexto histórico, los profesionales de la seguridad pueden identificar patrones, priorizar acciones de mitigación y fortalecer las defensas contra posibles amenazas.

- Además, la clasificación de los CVE facilita la comunicación entre los equipos de seguridad, los desarrolladores de software y los administradores de sistemas al proporcionar un marco común para discutir y abordar los problemas de seguridad. Esto permite una respuesta más rápida y eficiente a las vulnerabilidades conocidas, ayudando a proteger los sistemas y datos críticos contra posibles ataques.
- Sin embargo, es importante recordar que la clasificación de las CVE es solo el primer paso en la gestión de la seguridad cibernética. Para garantizar una protección efectiva contra las amenazas, es crucial implementar medidas proactivas de seguridad, como la aplicación regular de parches, la configuración adecuada de los sistemas, el monitoreo continuo de la red y la concientización sobre seguridad entre los usuarios y el personal de TI.
- Al combinar la clasificación de las CVE con prácticas de seguridad sólidas, la Universidad Politécnica Estatal del Carchi puede reducir de manera significativa su exposición a riesgos de seguridad y fortalecer su postura general de seguridad cibernética.

Referencias

1. Baçer, M., Güven, E. Y., & Aydin, M. A. (2021)... https://doi.org/10.1109/UBMK52708.2021.9558948
2. Cabrera, G. (27 de enero de 2022). somospnt. somospnt: https://somospnt.com/blog/241-que-es-kibana-configuracion-basica#:~:text=Kibana%20es%20una%20aplicación%20frontend,de%20datos%20almacenados%20en%20Elasticsearch.
3. Cheswick, W. R. (1992) ... USENIX.
4. Cole, E., & Northcutt, S. (2002). Honeypots: A security manager's guide to honeypots. SANS Institute.
5. Cole, E., & Northcutt, S. (2002) ... SANS Institute.
6. Elasticsearch. (2023). Elastic. Elastic: https://www.elastic.co/es/elasticsearch

7. Fan, W., et al. (2024). HoneyDOC: An efficient honeypot architecture. arXiv. https://doi.org/10.48550/arXiv.2402.06516
8. Fan, W., et al. (2024) ... https://doi.org/10.48550/arXiv.2402.06516
9. Feily, M., Shahrestani, A., & Ramadass, S. (2009). Botnet detection. SECURWARE. https://doi.org/10.1109/SECURWARE.2009.48
10. Feily, M., et al. (2009) ... https://doi.org/10.1109/SECURWARE.2009.48
11. Fortinet. (2023). Fortinet threat landscape report Q3 2023. Fortinet Inc. https://www.fortinet.com/blog/threat-research/q3-2023-threat-report
12. Fraunholz, D., et al. (2021). An adaptive honeypot configuration. arXiv. https://doi.org/10.48550/arXiv.2111.03884
13. Jicha, A., et al. (2016). SCADA honeypots: Conpot analysis. IEEE ISI. https://doi.org/10.1109/ISI.2016.7745463
14. Jin, Y., Wang, H., & Wang, X. (2020). A survey of honeypot research. Computer Networks. https://doi.org/10.1016/j.comnet.2020.107237
15. Kosheliuk, V., & Tulashvili, Y. (2024). Implementing honeypots with AWS and ELK. Computing. https://doi.org/10.47839/ijc.23.4.3761
16. Kosheliuk, V., & Tulashvili, Y. (2024)... https://doi.org/10.47839/ijc.23.4.3761
17. Kristyanto, M. A., et al. (2022)... https://doi.org/10.1109/ICoICT55009.2022.9914864
18. Kubba, A., et al. (2025). A systematic review of honeypot data collection. SSRN. https://doi.org/10.2139/ssrn.5242873
19. Langner, R. (2011). Stuxnet: Dissecting a cyberwarfare weapon. IEEE Security & Privacy, 9(3), 49–51. https://doi.org/10.1109/MSP.2011.67
20. Langner, R. (2011)... https://doi.org/10.1109/MSP.2011.67
21. Litchfield, D. (2003). The database hacker's handbook: Defending database servers. Wiley Publishing.
22. López-Morales, E., et al. (2020). HoneyPLC: A next-generation honeypot. ACM CCS. https://doi.org/10.1145/3372297.3417231
23. Machmeier, S. (2024). Honeypot implementation in a cloud environment. arXiv. https://doi.org/10.48550/arXiv.2301.00710
24. Machmeier, S. (2024)... https://doi.org/10.48550/arXiv.2301.00710

25. Matin, I. M. M., & Rahardjo, B. (2019)... https://doi.org/10.1109/CITSM47753.2019.8965419
26. Mehta, S., et al. (2021)... https://doi.org/10.1109/ICSES52305.2021.9633881
27. Mokube, I., & Adams, M. (2007). Honeypots: Concepts, approaches, and challenges. ACM, 321–326. https://doi.org/10.1145/1233341.1233399
28. Mokube, I., & Adams, M. (2007)... https://doi.org/10.1145/1233341.1233399
29. Morić, Z., et al. (2025). Advancing cybersecurity with honeypots. Informatics. https://doi.org/10.3390/informatics12010014
30. Mudgal, A., & Bhatia, S. (2022)... https://doi.org/10.1109/COM-IT-CON54601.2022.9850502
31. Provos, N., & Holz, T. (2007). Virtual Honeypots. Addison-Wesley.
32. Spitzner, L. (2003). Honeypots: Tracking Hackers. Addison-Wesley.
33. Torres, C. F., et al. (2019). The art of the scam: Demystifying honeypots in Ethereum smart contracts. arXiv. https://doi.org/10.48550/arXiv.1902.06976
34. Wählisch, M., et al. (2013). Design, implementation, and operation of a mobile honeypot. arXiv. https://doi.org/10.48550/arXiv.1301.7257
35. Zhou, Y., & Jiang, X. (2012). Dissecting Android malware: Characterization and evolution. IEEE Symposium on Security and Privacy, 95–109. https://doi.org/10.1109/SP.2012.16
36. kaspersky. (2023). kaspersky. kaspersky resources: https://www.kaspersky.es/resource-center/threats/what-is-a-honeypot