Herramienta para escaneo de información de la arquitectura en máquinas Linux

Miguel Ángel Pereira Pérez

Universidad Católica San Antonio de Murcia, España

Abstract. The abstract should briefly summarize the contents of the paper in 150–250 words.

1 Estado del arte

El desarrollo de herramientas automatizadas para el descubrimiento y análisis de información en sistemas Linux representa un área de investigación activa en el campo de la ciberseguridad. Este estado del arte examina las técnicas, metodologías y herramientas existentes para la recopilación de información de arquitecturas Linux, proporcionando el contexto necesario para justificar el desarrollo de una nueva herramienta especializada en este dominio.

1.1 Fundamentos del Information Gathering en Sistemas Linux

El *information gathering* constituye la base de cualquier evaluación de seguridad sobre sistemas Linux. Esta etapa inicial del proceso de análisis permite recolectar información crítica acerca de la infraestructura objetivo, desde el descubrimiento de sistemas activos en la red hasta detalles internos sobre su configuración, servicios, usuarios y posibles debilidades. Su correcta ejecución es indispensable tanto en auditorías ofensivas (pentesting) como en defensas proactivas (hardening y monitoreo), ya que proporciona el conocimiento necesario para comprender cómo está estructurada y protegida la superficie de ataque.

En el caso particular de los sistemas basados en Unix/Linux, la visibilidad de la información, el modelo de permisos y la estructura modular de los servicios hacen que esta fase adquiera una relevancia aún mayor. El dominio de estas técnicas permite no solo identificar vectores de entrada, sino también detectar desviaciones de seguridad o configuraciones erróneas que podrían ser aprovechadas por un atacante.

A continuación, se detallan las metodologías establecidas en la práctica profesional y las características técnicas propias de los entornos Unix/Linux que influyen directamente en las estrategias de recopilación de información.

Metodologías Establecidas de Descubrimiento

Las metodologías contemporáneas para el descubrimiento de información en sistemas Linux se estructuran tradicionalmente en fases secuenciales que van desde el reconocimiento inicial hasta la post-explotación. Esta aproximación sistemática incluye las fases de Reconocimiento, Enumeración, Análisis de Vulnerabilidades, Explotación y Post-Explotación, donde cada etapa proporciona información crítica para las siguientes. El reconocimiento inicial se centra en la identificación de sistemas activos, servicios en ejecución y características del sistema operativo, mientras que la enumeración profundiza en la obtención de información específica sobre usuarios, recursos compartidos y configuraciones del sistema.

La fase de post-explotación, una vez obtenido acceso a un sistema, se enfoca en la búsqueda sistemática de información local que permita tanto documentar el compromiso como facilitar movimientos laterales en la infraestructura. Esta información incluye listados de usuarios y contraseñas del archivo */etc/passwd* y */etc/shadow*, configuraciones de servicios, software instalado y certificados digitales que puedan proporcionar acceso a otros sistemas. La recopilación de esta información es fundamental para entender la arquitectura completa de la infraestructura y identificar vectores de ataque adicionales.

Características Específicas de los Sistemas Unix/Linux

Los sistemas basados en Unix presentan características únicas que los hacen especialmente interesantes desde el punto de vista del descubrimiento de información. Una característica fundamental es que las aplicaciones críticas como servidores web, bases de datos y servicios de correo electrónico habitualmente operan bajo cuentas de usuario dedicadas, lo que facilita la identificación de servicios basándose en los nombres de usuario presentes en el sistema. Esta práctica permite a un atacante predecir nombres de usuario locales en función del software identificado, como usuarios apache, mysql, oracle o tomcat.

Además, los sistemas Unix frecuentemente permanecen en funcionamiento durante largos períodos sin reinicios, lo que puede resultar en la acumulación de vulnerabilidades no parcheadas a lo largo del tiempo. Esta característica, combinada con la ausencia habitual de políticas de bloqueo de cuentas tras intentos fallidos de autenticación, hace que estos sistemas sean particularmente susceptibles a ataques de fuerza bruta y técnicas de enumeración. La gestión de contraseñas en estos sistemas también presenta aspectos únicos, con la evolución desde el modelo clásico que almacenaba contraseñas en */etc/passwd* hasta el modelo actual que utiliza */etc/shadow* con algoritmos de hashing más robustos.

1.2 Herramientas y Técnicas Actuales

Antes de entrar de comentar las distintas herramientas y técnicas actuales para llevar a cabo una auditoría de seguridad en un entorno empresarial es

conveniente diferenciar entre Herramientas de Seguridad Activa y Herramientas de Seguridad Pasiva.

Con respecto a las primeras, las herramientas pasivas obtienen información sin interactuar directamente con el objetivo. Se basan en datos públicos o en la observación del tráfico sin enviar paquetes al sistema bajo análisis. Esto tiene varias ventajas: es menos probable que activen alarmas en los sistemas de defensa, no generan logs evidentes en los objetivos y permiten realizar un mapeo preliminar antes de iniciar acciones más intrusivas.

Las herramientas activas interactúan directamente con el objetivo enviando paquetes, estableciendo conexiones o realizando peticiones específicas para extraer información. Esto permite verificar en tiempo real el estado de los sistemas, identificar versiones precisas, puertos abiertos y configuraciones actuales. Sin embargo, implican un riesgo elevado de ser detectadas por IDS/IPS o de dejar trazas en los logs del sistema auditado.

Existen muchas más clasificaciones que se podrían aplicar como por ejemplo atendiendo a las fases de una auditoría:

- Reconocimiento: Esta fase busca recopilar toda la información posible sobre el objetivo, sin interactuar directamente (pasivo) o mediante sondas (activo). Se suelen usar Herramientas OSINT, herramientas de escaneo de red, etc.
- Análisis de vulnerabilidades: Una vez identificados los sistemas y servicios, se analizan versiones, configuraciones y vulnerabilidades conocidas. En esta fase sería destacable hacer mención a la herramienta NMAP.
- Explotación: Esta fase trata de aprovechar las vulnerabilidades detectadas para obtener acceso no autorizado o control. Se usan frameworks de Explotación como Metaploit Framework, Hydra...
- Post-explotación: Tras comprometer un sistema, se busca escalar privilegios, persistencia y moverse lateralmente en la red. Se usan técnicas de Pivoting / Tunneling Tools: SSH tunnels, proxychains, metasploit route.

Herramientas de Seguridad Activa

- **Nmap**: más allá del escaneo de puertos, Nmap detecta servicios, versiones y sistemas operativos mediante técnicas de fingerprinting como TCP/IP stack analysis. Su motor NSE permite incluso scripts para descubrir vulnerabilidades específicas (ej. http-vuln-cve2014-3704 para Drupalgeddon).
- **Masscan**: centrado en velocidad extrema, es ideal para redes extensas o Internet-wide scanning, pero menos detallado que Nmap (solo reporta puertos abiertos).
- **Netcat**: útil para establecer conexiones directas, verificar manualmente banners o transferir archivos. Es clave para confirmar manualmente hallazgos de escaneos automatizados.
- **Nikto**: realiza un fuzzing sencillo sobre servidores HTTP/HTTPS, identificando configuraciones inseguras, archivos ocultos y versiones vulnerables.

- **Hydra**: empleada en ataques de fuerza bruta sobre múltiples protocolos. Su velocidad configurable y soporte para listas de credenciales la hacen esencial para auditar la robustez de contraseñas.
- **Sqlmap**: automatiza la identificación y explotación de inyecciones SQL, validando si es posible extraer datos sensibles, escalar privilegios o incluso escribir archivos en el sistema.
- **Metasploit Framework**: además de ejecutar exploits, cuenta con módulos auxiliares para escanear redes, enumerar usuarios SMB, o comprobar vulnerabilidades sin necesariamente explotar.

Ventajas:

- Validación directa del estado actual del objetivo.
- Permite confirmar vulnerabilidades activas que OSINT o escaneos pasivos no pueden demostrar.
- Es fundamental para auditorías de cumplimiento normativo o test de intrusión controlados.

Limitaciones:

- Genera tráfico detectable que puede disparar alarmas o bloquear la IP del auditor.
- Puede alterar el sistema objetivo (por ejemplo, dejar registros en logs o incluso provocar cierres inesperados por exploración excesiva de servicios).
- Requiere un manejo ético y controlado, con consentimiento explícito del cliente en entornos reales.

Herramientas de Seguridad Pasiva

- **Spiderfoot**: automatiza el OSINT mediante más de 200 módulos. Es capaz de realizar correlaciones avanzadas, como detectar si un dominio objetivo está relacionado con direcciones IP maliciosas en listas negras, o si aparecen credenciales expuestas en bases de datos públicas.
- **TheHarvester**: particularmente eficaz para recolectar correos electrónicos y subdominios desde motores de búsqueda, certificados SSL públicos y claves PGP. Esto permite identificar la huella digital de una organización antes de enviar cualquier paquete.
- **Maltego CE**: permite explorar gráficamente las relaciones entre dominios, IPs, organizaciones y usuarios, obteniendo una visión estructurada de potenciales vectores de ataque o conexiones entre entidades.
- **Shodan (CLI o web)**: proporciona banners de servicios accesibles públicamente, detectando versiones de software expuestas, configuraciones incorrectas y potenciales CVEs sin necesidad de tocar directamente la red objetivo.
- **dnsenum, dnsrecon**: realizan consultas sobre registros DNS para identificar posibles transferencias de zona, registros MX o SRV que pueden revelar la estructura interna de una organización.

Ventajas:

- Bajo riesgo de detección.
- Permiten elaborar perfiles detallados del objetivo (personas, correos, tecnología utilizada) sin alertar al equipo de seguridad.
- Son fundamentales en las etapas iniciales del ciclo de pentesting (reconocimiento pasivo).

Limitaciones:

- La información está limitada a lo que ya es público.
- No valida si un servicio está realmente activo o accesible desde el punto de vista del auditor.
- Puede no reflejar el estado actual (por ejemplo, un servicio detectado vía Shodan puede haber sido deshabilitado).

Herramientas de Escaneo de Red

Las herramientas actuales para el descubrimiento de información se pueden categorizar según su función específica dentro del proceso de reconocimiento. Nmap se ha establecido como el estándar de facto para el escaneo de puertos y la identificación de servicios, proporcionando capacidades avanzadas a través de su sistema de scripts NSE que permite realizar pruebas específicas como la identificación de servidores FTP con acceso anónimo o sistemas Windows vulnerables a través de SMB. Esta herramienta es fundamental para la fase inicial de descubrimiento, permitiendo mapear la superficie de ataque disponible en la red objetivo.

Netcat, conocida como la "navaja suiza de TCP/IP", proporciona funcionalidades versátiles para el fingerprinting de sistemas, escaneo de puertos, transferencia de archivos y establecimiento de shells remotas. Su simplicidad y flexibilidad la convierten en una herramienta esencial para la verificación manual de servicios y la realización de conexiones de bajo nivel. Estas herramientas, sin embargo, requieren un conocimiento técnico significativo y operación manual para obtener resultados completos y contextualizados.

Frameworks de Explotación

Metasploit Framework representa la plataforma de explotación más utilizada en la actualidad, proporcionando una arquitectura modular que incluye exploits, payloads, encoders y módulos NOP. Su relevancia para el "information gathering" radica en sus capacidades de post-explotación, donde módulos especializados permiten la extracción automatizada de información crítica una vez obtenido acceso a un sistema. El framework incluye tres tipos principales de payloads: singles (autocontenidos), stagers (establecimiento de conexiones) y stages (código final ejecutado), cada uno adaptado a diferentes escenarios de explotación.

Meterpreter, como componente avanzado de Metasploit, proporciona capacidades específicas para la recopilación de información en sistemas comprometidos, incluyendo comandos para la navegación del sistema de archivos, obtención de información del sistema, manipulación de procesos y extracción de credenciales. Sin embargo, estas herramientas están diseñadas principalmente para escenarios de penetration testing activo y requieren acceso previo al sistema objetivo, limitando su aplicabilidad en escenarios de análisis defensivo o auditoría de configuración.

Herramientas de Reconocimiento OSINT

Spiderfoot es una plataforma de reconocimiento automatizado orientada a la recopilación de inteligencia de fuentes abiertas (OSINT), capaz de descubrir relaciones entre direcciones IP, dominios, correos electrónicos, nombres de usuario, claves públicas PGP y otros identificadores. Su arquitectura modular permite el uso de más de un centenar de módulos para la exploración de servicios, registros DNS, brechas de seguridad conocidas, información en redes sociales, bases de datos públicas de incidentes y más. Esta herramienta es especialmente relevante para la fase de reconocimiento pasivo y para la identificación de posibles vectores de ataque antes de interactuar activamente con los sistemas objetivo. Además, su interfaz web y su capacidad de generar informes detallados facilitan la correlación de datos obtenidos a partir de múltiples fuentes, mejorando el contexto del descubrimiento inicial. Al igual que otras herramientas OSINT como TheHarvester o Maltego, Spiderfoot complementa los análisis activos al proporcionar una base de información previa que guía y optimiza las fases subsiguientes del proceso de descubrimiento.

1.3 Análisis de Protocolos y Servicios

El análisis detallado de los protocolos y servicios expuestos por un sistema Linux es un componente esencial dentro del proceso de information gathering, ya que permite identificar vectores de ataque específicos asociados a cada tipo de tecnología empleada. Esta etapa se apoya en técnicas de enumeración orientadas a cada protocolo —como FTP, SSH, SNMP, SMB o LDAP— con el objetivo de descubrir configuraciones inadecuadas, accesos no controlados o errores de implementación que pueden comprometer la seguridad del sistema.

A través del uso de herramientas especializadas y técnicas diseñadas para explotar comportamientos protocolarios predecibles, es posible obtener información sensible sin necesidad de explotación directa, lo que convierte a esta fase en una de las más valiosas tanto en análisis preventivos como ofensivos. Asimismo, el conocimiento profundo de las vulnerabilidades históricas y recurrentes en protocolos ampliamente adoptados permite anticipar riesgos y aplicar contramedidas efectivas.

En las siguientes secciones se detallan las principales estrategias de enumeración por protocolo y se analizan algunas de las vulnerabilidades más relevantes asociadas a servicios de red críticos en entornos Linux.

Técnicas de Enumeración por Protocolo

El estado actual del arte incluye técnicas especializadas para la enumeración de información a través de diferentes protocolos de red. Para servicios FTP, las técnicas se centran en la verificación de acceso anónimo, lo cual puede revelar información sensible almacenada en servidores mal configurados. La metodología estándar implica intentar autenticación con el usuario anonymous y contraseñas en formato de correo electrónico, seguido de la exploración del contenido accesible.

Para servicios SSH, las técnicas incluyen la verificación de claves privadas conocidas y la enumeración de usuarios válidos aprovechando diferencias en los tiempos de respuesta de ciertas implementaciones de OpenSSH. Los servicios SNMP presentan oportunidades particulares para la recopilación de información, especialmente cuando utilizan comunidades por defecto como public o private, permitiendo acceso a la base de datos MIB que contiene información detallada sobre configuración de red, tablas de rutas e información del sistema operativo.

Vulnerabilidades en Servicios de Red

Los servicios SMB han sido históricamente una fuente significativa de vulnerabilidades, desde las sesiones nulas en versiones antiguas de Windows hasta vulnerabilidades críticas como EternalBlue. Las sesiones nulas permiten la enumeración de usuarios y grupos, recursos compartidos y SIDs sin autenticación, utilizando técnicas como RID Cycling para identificar sistemáticamente cuentas de usuario. Los servicios NFS en versiones 2 y 3 presentan vulnerabilidades fundamentales en su modelo de autenticación basado únicamente en UID y GID, permitiendo evasión de controles de acceso mediante la manipulación de identificadores de usuario.

Los servicios LDAP pueden proporcionar acceso a información sensible del directorio activo, especialmente cuando permiten acceso anónimo a bases de datos que deberían estar protegidas. Esta información puede incluir estructuras organizacionales, relaciones entre usuarios y sistemas, y configuraciones de seguridad que faciliten ataques posteriores.

1.4 Técnicas de Post-Explotación y Escalada de Privilegios

Una vez comprometido un sistema, el atacante se encuentra en posición de ampliar su control mediante técnicas de post-explotación, cuyo objetivo es extraer información sensible y preparar nuevas fases de ataque, como el movimiento lateral o el acceso persistente. En entornos Linux, estas técnicas se centran principalmente en la búsqueda de credenciales, certificados, secretos y configuraciones expuestas que puedan reutilizarse en otros nodos de la red.

Además, la escalada de privilegios es una etapa crítica que permite convertir un acceso limitado en un control total del sistema. Para ello, se explotan

debilidades locales como configuraciones erróneas, permisos inseguros o vulnerabilidades del kernel y software instalado. El conocimiento profundo de la arquitectura de permisos de Linux, junto con herramientas especializadas para el análisis de servicios y usuarios, permite detectar oportunidades de elevación que, de otro modo, podrían pasar desapercibidas.

Este apartado examina las principales técnicas utilizadas en fases avanzadas de compromiso, destacando su relevancia tanto para operaciones ofensivas como para reforzar defensas desde una perspectiva preventiva.

Búsqueda de Credenciales y Certificados

Una vez obtenido acceso inicial a un sistema Linux, las técnicas actuales se centran en la búsqueda sistemática de credenciales almacenadas en diversas ubicaciones. Los archivos */etc/passwd* y */etc/shadow* contienen información crítica sobre usuarios locales y sus contraseñas hasheadas, siendo el primer archivo accesible públicamente mientras que el segundo requiere privilegios de administrador. La extracción de estos hashes permite su posterior crackeado utilizando herramientas especializadas como John the Ripper.

Además de las credenciales del sistema, las técnicas modernas incluyen la búsqueda de contraseñas almacenadas en archivos de configuración, scripts de administración y directorios de usuario. Los certificados digitales y claves privadas encontrados en directorios de usuario pueden proporcionar acceso a otros sistemas de la infraestructura, mientras que el código fuente de aplicaciones frecuentemente contiene cadenas de conexión con credenciales embebidas.

Escalada de Privilegios en Linux

Las técnicas de escalada de privilegios en sistemas Linux se basan en la identificación de vulnerabilidades locales, servicios mal configurados y permisos inadecuados en el sistema de archivos. La identificación de servicios ejecutándose con privilegios elevados pero accesibles por usuarios con menores privilegios representa una vía común de escalada. La búsqueda de contraseñas almacenadas en texto claro en archivos de configuración o scripts representa otra técnica fundamental.

La evaluación de permisos en directorios críticos del sistema puede revelar oportunidades para la modificación de ejecutables o bibliotecas que posteriormente serán ejecutados con privilegios elevados. Las vulnerabilidades locales del kernel o de servicios específicos proporcionan vías adicionales para la escalada de privilegios, requiriendo la identificación precisa del software instalado y sus versiones.

1.5 Limitaciones de las Soluciones Actuales

A pesar del amplio ecosistema de herramientas disponibles para el descubrimiento de información en sistemas Linux, el enfoque predominante presenta limitaciones significativas en términos de eficiencia, cohesión y escalabilidad. La

mayoría de las soluciones actuales están orientadas a tareas específicas, carecen de interoperabilidad real y requieren un alto grado de intervención humana para su correcta operación. Esta realidad plantea barreras operativas importantes tanto para analistas de seguridad como para administradores de sistemas que buscan realizar evaluaciones periódicas o mantener un control continuo sobre su infraestructura.

En este apartado se analizan dos limitaciones clave que justifican la necesidad de una nueva herramienta integral: la fragmentación del ecosistema de utilidades existentes y la falta de mecanismos de contextualización automática de los hallazgos. Ambos factores dificultan la obtención de una visión unificada de la postura de seguridad y reducen la capacidad de respuesta ante posibles desviaciones o vulnerabilidades.

Fragmentación de Herramientas

El panorama actual se caracteriza por una significativa fragmentación de herramientas, cada una especializada en aspectos específicos del proceso de descubrimiento de información. Esta especialización, aunque proporciona capacidades técnicas avanzadas, requiere que los analistas dominen múltiples herramientas con diferentes interfaces, formatos de salida y metodologías de operación. La integración de resultados de diferentes herramientas frecuentemente requiere procesamiento manual y correlación de datos, introduciendo posibilidades de error y aumentando el tiempo necesario para completar evaluaciones comprehensivas.

La mayoría de herramientas actuales están diseñadas para operación manual o semi-automatizada, requiriendo intervención humana continua para la toma de decisiones sobre qué técnicas aplicar y cómo interpretar los resultados obtenidos. Esta limitación es particularmente significativa en entornos donde se requiere análisis sistemático de múltiples sistemas o evaluaciones periódicas de configuración de seguridad.

Ausencia de Contextualización Automática

Las herramientas existentes generalmente proporcionan datos en bruto sin contextualización automática sobre las implicaciones de seguridad de la información descubierta. Por ejemplo, mientras Nmap puede identificar servicios en ejecución, no proporciona automáticamente evaluación del riesgo asociado con configuraciones específicas o recomendaciones de endurecimiento. Esta limitación requiere que los analistas posean conocimiento técnico profundo para interpretar correctamente los resultados y determinar las acciones apropiadas.

La falta de integración con sistemas de monitorización y gestión de configuración representa otra limitación significativa. Las herramientas actuales operan típicamente como soluciones puntuales que no se integran naturalmente con la infraestructura de monitorización existente, limitando su utilidad para la supervisión continua de configuraciones de seguridad.

1.6 Oportunidades de Investigación

La evolución constante del panorama de amenazas y la creciente complejidad de las infraestructuras empresariales Linux plantean importantes desafíos para la gestión de la seguridad. En este contexto, existe una clara oportunidad de investigación en el desarrollo de herramientas que no solo repliquen técnicas tradicionales, sino que ofrezcan capacidades inteligentes, adaptativas y fácilmente integrables en entornos reales. La herramienta propuesta en este trabajo pretende abordar directamente estas necesidades, aportando valor tanto a equipos de ciberseguridad ofensiva como defensiva.

Automatización Inteligente

La automatización del proceso de descubrimiento de información es esencial para responder a los requisitos actuales de velocidad, precisión y cobertura en auditorías de seguridad. Las herramientas existentes, aunque potentes, suelen requerir intervención manual, lo que introduce ineficiencias y dependencias del conocimiento experto. En este sentido, una herramienta que automatice la selección y ejecución de técnicas de escaneo, adaptándose a las características del entorno, representa un salto cualitativo.

Además, la incorporación de capacidades de aprendizaje automático (machine learning) permitiría identificar patrones anómalos en configuraciones del sistema, ayudando a detectar desviaciones respecto a una línea base segura. Por ejemplo, un modelo entrenado sobre configuraciones estándar podría identificar servicios habilitados innecesariamente, cambios no autorizados en permisos o la aparición de nuevos procesos sospechosos.

Esta capacidad también se extiende al análisis de los resultados de escaneo: correlacionar hallazgos técnicos (versiones, puertos abiertos, servicios) con bases de datos de vulnerabilidades y guías de endurecimiento (hardening) puede permitir la generación automática de recomendaciones priorizadas, reduciendo la carga analítica sobre el operador humano.

Integración con Ecosistemas de Monitorización

Las arquitecturas modernas requieren soluciones que no actúen de forma aislada, sino que se integren nativamente con las plataformas de monitorización y respuesta existentes. En este sentido, la herramienta diseñada debe incluir soporte para exportar datos en formatos estructurados (ej. JSON, YAML o directamente índices para ElasticSearch), facilitando su ingestión por soluciones SIEM como Wazuh, Splunk o Elastic Security.

La capacidad de emitir alertas automáticas cuando se detecten cambios críticos en la configuración (ej. nuevos servicios abiertos, usuarios inesperados, modificaciones en ficheros sensibles) transformaría el descubrimiento pasivo en una supervisión activa continua. Esto permitiría extender su uso más allá del pentesting puntual y convertirlo en una herramienta de "higiene continua de seguridad" dentro de procesos de defensa activa o blue teaming.

Además, la interoperabilidad con plataformas de gestión de vulnerabilidades, como OpenVAS o Tenable.io, permitiría enriquecer sus reportes con análisis de riesgo basados en CVSS y priorización según la criticidad del entorno. Esto facilitaría la toma de decisiones estratégicas sobre qué vulnerabilidades abordar primero, optimizando la asignación de recursos.

1.7 Contexto Actual: Justificación de la Herramienta

En el entorno actual de ciberseguridad, caracterizado por ataques automatizados, brechas relacionadas con errores de configuración y entornos híbridos en rápida evolución (cloud, on-prem, contenedores), las herramientas deben ser ágiles, adaptables y estar profundamente integradas con el resto de la infraestructura defensiva.

La herramienta propuesta, que combina descubrimiento de red, análisis local de sistemas Linux, evaluación de permisos y configuración, y generación de reportes contextualizados, puede convertirse en una solución clave para:

- Administradores que necesiten detectar desviaciones o configuraciones peligrosas sin realizar análisis manuales.
- Analistas de seguridad que requieran correlacionar configuraciones reales con inteligencia de amenazas.
- Pentesters y Red Teamers que busquen maximizar la eficacia de la fase de reconocimiento sin depender de múltiples herramientas fragmentadas.

Al integrar capacidades de escaneo con análisis de protección local (permisos, usuarios, servicios activos) y un mecanismo de reporte estructurado, esta herramienta cubre un vacío funcional importante entre las utilidades clásicas de escaneo y los sistemas de monitorización tradicional.

2 Motivación

3 Hipótesis

4 Tesis

5 Demostración

Acknowledgments. A bold run-in heading in small font size at the end of the paper is used for general acknowledgments, for example: This study was funded by X (grant number Y).

Disclosure of Interests. It is now necessary to declare any competing interests or to specifically state that the authors have no competing interests. Please place the statement with a bold run-in heading in small font size beneath the (optional) acknowledgments[1], for example: The authors have no competing interests to declare that are relevant to the content of this article. Or: Author A has received research grants from Company W. Author B has received a speaker honorarium from Company X and owns stock in Company Y. Author C is a member of committee Z.

References

1. Author, F.: Article title. Journal **2**(5), 99–110 (2016)
2. Author, F., Author, S.: Title of a proceedings paper. In: Editor, F., Editor, S. (eds.) CONFERENCE 2016, LNCS, vol. 9999, pp. 1–13. Springer, Heidelberg (2016). https://doi.org/10.10007/1234567890
3. Author, F., Author, S., Author, T.: Book title. 2nd edn. Publisher, Location (1999)
4. Author, A.-B.: Contribution title. In: 9th International Proceedings on Proceedings, pp. 1–2. Publisher, Location (2010)
5. LNCS Homepage, http://www.springer.com/lncs, last accessed 2023/10/25

[1] If EquinOCS, our proceedings submission system, is used, then the disclaimer can be provided directly in the system.