

30 Técnicas Blue Team críticas para 2025

1 de Mayo de 2025



Unai Rubio
HACKER ÉTICO

30 Técnicas de Blue Team para 2025

Técnica 1: Threat Hunting en EDR modernos usando MITRE ATT&CK

El objetivo es **detectar amenazas persistentes y avanzadas (APT)** mediante el análisis proactivo de datos de telemetry recolectados desde los endpoints. Herramientas como **CrowdStrike Falcon**, **Microsoft Defender for Endpoint** y **SentinelOne** permiten no solo monitoreo en tiempo real, sino también ejecutar búsquedas estructuradas basadas en tácticas del marco **MITRE ATT&CK**.

Las acciones clave incluyen:

- Usar **consultas YARA** junto con **MITRE ATT&CK Navigator** para mapear comportamientos sospechosos a tácticas reales (como Execution, Persistence o Lateral Movement).
- Ejecutar **scripts remotos desde la consola EDR** para recolectar artefactos forenses sin interrumpir al usuario final.
- Extraer **memoria RAM en vivo** directamente desde el endpoint infectado para investigación profunda, sin necesidad de apagar el equipo y perder evidencia crítica.

📌 **Recurso útil:** La **MITRE ATT&CK Matrix**, que actúa como mapa táctico universal para entender y anticiparse a las técnicas usadas por atacantes reales.

Técnica 2: Aislar malware en ejecución sin alterar evidencia

Cuando un sistema está comprometido y el malware aún se encuentra activo, lo más crítico es **contener la amenaza sin destruir evidencia forense clave**. El primer paso es **desconectar físicamente o bloquear el acceso de red del host afectado**, evitando que el atacante tenga control remoto, pero **sin reiniciar el equipo**, ya que esto podría borrar datos volátiles como claves de cifrado o credenciales en memoria.

Para preservar toda la información posible, se realiza un **memory dump completo usando herramientas como Volatility, Velociraptor o strings**, permitiendo analizar procesos en ejecución, DLLs inyectadas o conexiones TCP activas. Además, se pueden usar utilidades como **Process Explorer (Sysinternals)** para inspeccionar comportamiento sospechoso en tiempo real.

Por último, para evitar que el malware entre en pánico al perder comunicación, se **redirigen las conexiones salientes maliciosas hacia un sinkhole seguro**, donde se monitorean los comandos que intenta recibir el atacante. Esto no solo ayuda a entender su propósito, sino que también permite rastrear posibles C2s (Command and Control) usados.

🔧 **Herramientas recomendadas:** Process Explorer (Sysinternals), NirSoft Winless, Volatility, Velociraptor.

📌 **Práctica ética y útil** tanto en entornos corporativos como forenses, respetando cadena de custodia digital.

Técnica 3: Crear honeypots internos para detectar movimientos laterales

Despliega sistemas falsos dentro de tu red corporativa para atraer y monitorear actividades sospechosas de atacantes ya dentro del perímetro. Usando herramientas como **Cowrie o T-Pot**, puedes simular servicios comunes como **SSH, SMB, RDP e incluso controladores Active Directory falsos**, engañando a los atacantes para que revelen sus tácticas sin poner en riesgo activos reales.

Cada intento de conexión, autenticación o ejecución remota se registra automáticamente, permitiendo recolectar **IOCs (Indicadores de Compromiso)** como IPs, credenciales usadas, comandos ejecutados y hasta payloads descargados. Esto no solo ayuda a **detectar lateral movement temprano**, sino también a entender las técnicas usadas por los atacantes en tu entorno específico.

🔧 **Cómo hacerlo :**

- Instalar honeypots SSH como **Cowrie** o soluciones más avanzadas como **T-Pot (multi-servicio)**.
- Configurar servicios falsos estratégicos (por ejemplo, un DC falso en una VLAN HR).
- Monitorear logs y alertar ante cualquier interacción inusual.

📌 **Recurso recomendado:** [Cowrie Honeypot \(GitHub\)](#) - ideal para principiantes en honeypots SSH.

✅ **Objetivo realista:** Detectar intrusos internos antes de que lleguen a su objetivo verdadero.

Técnica 4: Logs de alto valor en Windows y Linux

Para detectar actividades maliciosas antes de que se conviertan en incidentes graves, debes enfocarte en los logs con mayor capacidad predictiva. En sistemas **Windows** , presta especial atención a los eventos **4688 (Process Creation)** y **7045 (Service Install)** , ya que te muestran qué procesos se ejecutan y qué servicios nuevos se instalan – tácticas comunes en escaladas de privilegios o persistencia.

Busca también comandos sospechosos como **net user** , **net localgroup** , **at / schtasks** , típicamente usados por atacantes para crear usuarios o programar tareas remotas. En **Linux** , revisa siempre los archivos **.bash_history** , los logs del **auditd** y las trazas de **systemd-journald** , donde pueden esconderse pistas sobre ejecuciones no autorizadas o modificaciones críticas.

🔧 **Herramienta clave:** Usa reglas estandarizadas como **Sigma Rules** junto con plataformas de análisis como **ELK Stack** o **SIEMs compatibles** , para automatizar la detección de patrones maliciosos en estos logs, incluso a gran escala.

🚩 **Objetivo realista:** Identificar indicios tempranos de ejecución remota, escalada de privilegios o persistencia, usando fuentes de datos disponibles por defecto en cualquier entorno.

Técnica 5: Automatización básica de respuestas con SOAR

Para reducir significativamente el tiempo de respuesta ante incidentes comunes, es clave automatizar tareas repetitivas y predecibles usando plataformas **SOAR (Security Orchestration, Automation and Response)** . Escenarios ideales incluyen **bloquear IPs maliciosas en firewalls tras una alerta SIEM** , **aislar máquinas infectadas automáticamente** o incluso **generar casos en sistemas de gestión como Jira o ServiceNow** para coordinación ágil entre equipos.

Estas acciones se pueden orquestar mediante **playbooks en herramientas como Phantom, TheHive o Cortex XSOAR** , permitiendo ejecutar secuencias de respuesta complejas con un solo clic o incluso de forma totalmente automática. Esto no solo mejora la eficiencia del equipo, sino que también evita errores humanos bajo presión y mantiene consistencia en los procedimientos de respuesta.

🔧 **Herramienta clave:** Usa playbooks en **Cortex XSOAR** , **TheHive** o plataformas similares para orquestar acciones entre firewalls, EDRs, SIEMs y sistemas de tickets. Conectores APIs permiten integrar casi cualquier sistema de seguridad moderno.

🚩 **Objetivo realista:** Reducir el **MTTR (Mean Time to Respond)** a menos de 5 minutos en escenarios comunes, mejorando drásticamente la postura de seguridad operativa.

Técnica 6: Uso de EDR para detección de Living Off The Land Binaries (LOLBins)

Los atacantes cada vez usan más binarios del sistema operativo para evitar detecciones basadas en firmas. Esta técnica se enfoca en identificar el uso malicioso de herramientas legítimas como **PowerShell**, **Mshta**, **Regsvr32** o **Rundll32.exe** , que son abusadas para descargar payloads, ejecutar comandos ocultos o mantener persistencia sin dejar archivos en disco.

Mediante el uso combinado de **Sysmon (System Monitor)** y plataformas de monitorización como **Wazuh** o **Microsoft Defender ATP** , es posible capturar eventos de ejecución sospechosa, cargas de DLLs inusuales o cadenas de comandos obfusadas. Las reglas Sigma, especialmente diseñadas para detectar estas tácticas, permiten estandarizar las búsquedas en logs y alertar automáticamente ante comportamientos atípicos.

🔧 **Herramienta clave:** Configura **Sysmon** para registrar eventos de creación de procesos, cargas de imágenes y actividad de scripts, e integra esos logs en una plataforma SIEM/EDR como **Wazuh** , donde puedes aplicar correlaciones y alertas automatizadas.

🚩 **Objetivo realista:** Detectar técnicas "fileless" y living-off-the-land antes de que se conviertan en incidentes graves, mejorando la visibilidad sobre amenazas discretas.

Técnica 7: Implementar Memory Forensics para incidentes críticos

Cuando se sospecha de un ataque avanzado o malware "fileless", la memoria RAM puede contener evidencia crítica invisible desde el disco. Esta técnica se enfoca en **capturar y analizar la memoria en vivo de un sistema comprometido**, permitiendo descubrir procesos ocultos, inyecciones de código, credenciales en texto plano y conexiones TCP activas que no aparecen en los logs tradicionales.

El proceso comienza usando herramientas como **Dumplt o Belkasoft Live RAM Capturer** para generar una imagen completa de la memoria física del equipo afectado, sin alterar su estado ni interrumpir operaciones críticas. Luego, se analiza esta imagen con plataformas especializadas como **Volatility o Rekall**, capaces de desglosar estructuras internas del kernel, listar procesos no visibles y extraer payloads en ejecución.

🔧 **Herramienta clave:** Usa **Volatility (disponible en GitHub)** por su soporte amplio y comunidad activa; crea perfiles personalizados según el sistema operativo del host afectado para mejorar resultados.

📌 **Objetivo realista:** Obtener inteligencia forense profunda sobre ataques discretos, especialmente útiles en casos de malware persistente, rootkits o técnicas de evadir AVs/EDRs.

Técnica 8: Análisis de malware en entorno aislado (sandbox inteligente)

Para entender el verdadero comportamiento de un archivo sospechoso sin riesgo para tu red, debes ejecutarlo en un **entorno controlado y aislado**, como una máquina virtual con redes virtuales segmentadas. Esto permite observar cómo actúa el malware en tiempo real, desde sus conexiones salientes hasta las modificaciones que realiza en el sistema.

La forma más efectiva es usar plataformas especializadas como **ANY.RUN o Cuckoo Sandbox**, junto con máquinas virtuales aisladas físicamente (por ejemplo, VMware Workstation Pro o VirtualBox tras un firewall virtual). Estas herramientas permiten no solo **capturar IOCs (Indicadores de Compromiso)**, sino también analizar llamadas al sistema, inyecciones de memoria, cambios en el registro de Windows, creación de servicios y comunicación C2.

🔧 **Herramienta clave:** Usa **ANY.RUN** por su interfaz intuitiva y soporte de análisis dinámico en tiempo real, o **Cuckoo Sandbox** si necesitas personalización avanzada y análisis automatizado bajo Linux.

✦ **Objetivo realista:** Obtener inteligencia operativa sobre nuevos malware, comprobar falsos positivos AV y construir reglas de detección propias basadas en comportamiento real.

Técnica 9: Configuración de Data Diodes para redes sensibles

En entornos críticos como ICS (Industrial Control Systems) o sistemas SCADA , donde la seguridad física y digital están directamente conectadas, es fundamental garantizar un **flujo de datos estrictamente unidireccional** . Los **Data Diodes** son dispositivos físicos que permiten la salida de información desde una red sensible hacia otra menos segura, pero **bloquean cualquier entrada** , asegurando así que no haya conexión inversa ni posibilidad de ataque lateral.

Estas soluciones se implementan típicamente en puntos críticos de la infraestructura para tareas como **monitoreo remoto seguro, respaldo de logs o transmisión de métricas operativas** , sin comprometer la integridad del sistema controlado. Al ser dispositivos basados en hardware con **conexiones unidireccionales a nivel físico** , son virtualmente imposibles de saltar incluso por un atacante avanzado.

✂ **Proveedor clave:** Para despliegues profesionales, se recomiendan soluciones como **Owl Cyber Defense** o **Waterfall Security Solutions** , ambas certificadas para uso en entornos industriales de alto riesgo y con integración transparente en arquitecturas ya existentes.

✦ **Objetivo realista:** Proteger redes ICS/SCADA de amenazas externas mediante una barrera física imposible de sobrepasar, mejorando drásticamente su postura de seguridad sin depender únicamente de software o reglas lógicas.

Técnica 10: Detección de comandos PowerShell obfuscados

PowerShell es una de las herramientas más usadas por atacantes para ejecutar código malicioso sin tocar disco (técnicas "fileless"). Para detectar estos abusos, es fundamental monitorear scripts cifrados, cadenas en base64 o comandos obfuscados que intenten evadir AVs.

La clave está en habilitar el **ScriptBlock Logging** a través de **Sysmon** o **políticas de grupo** , lo que permite registrar el contenido real de los scripts ejecutados, incluso si están codificados. A partir de ahí, se pueden aplicar **reglas Sigma** o **reglas nativas de EDRs como Microsoft Defender ATP** para detectar patrones sospechosos como -enc, iex (Invoke-Expression), o llamadas anómalas a cmd.exe /c powershell.

🔧 **Herramienta clave:** Usa **Microsoft Defender ATP** junto con **Sysmon logging** para capturar y analizar scripts PowerShell complejos. También puedes integrar estas técnicas con SIEMs como **Splunk** o **ELK** para alertas automatizadas.

🚩 **Objetivo realista:** Detectar cargadores fileless, descifrado de payloads en memoria y ejecución remota mediante scripts PowerShell ofuscados, antes de que causen daño real.

Técnica 11: Segmentación de red basada en Zero Trust

Para evitar que un atacante se mueva libremente dentro de tu infraestructura tras una inicial brecha, es fundamental implementar una estrategia de **segmentación de red basada en Zero Trust** . Esto implica dividir la red en zonas controladas (como DMZs y VLANs por funciones), limitando accesos solo a lo estrictamente necesario para cada usuario o servicio.

La implementación avanzada utiliza soluciones como **Cisco ACI** o **VMware NSX** para aplicar **micro-segmentación dinámica** , donde cada comunicación entre servidores, usuarios o aplicaciones debe ser previamente autorizada. Este enfoque no solo detiene los movimientos laterales, sino que también reduce drásticamente el radio de afectación en caso de compromiso.

🔧 **Estándar clave:** Basa tu diseño en las recomendaciones del **NIST SP 800-207** , un marco oficial que define principios operativos claros para desplegar Zero Trust en entornos empresariales y sensibles.

🚩 **Objetivo realista:** Eliminar el "movimiento lateral fácil" dentro de la red, obligando a los atacantes a superar múltiples barreras incluso si ya están dentro del perímetro.

Técnica 12: Uso de Network Traffic Analysis (NTA) para detección temprana

La **análisis de tráfico de red (NTA)** es una de las formas más efectivas de detectar actividades maliciosas antes de que se conviertan en incidentes críticos. Mediante el monitoreo continuo del tráfico interno y externo, es posible identificar comportamientos anómalos como **conexiones a IPs desconocidas, dominios recién registrados o patrones DNS atípicos**, típicos de malware en fase de Command & Control (C2).

Herramientas como **Darktrace**, con su enfoque basado en inteligencia artificial, o plataformas más técnicas como **Corelight (basada en Zeek/Bro)** y **Suricata**, permiten analizar a gran escala estos flujos de red, aplicando reglas de detección personalizadas o usando firmas predefinidas. Estas herramientas no solo ayudan a descubrir amenazas avanzadas, sino también a cumplir con estándares de detección temprana en entornos corporativos.

🔧 **Herramienta clave:** Usa **Zeek (Bro)** o **Suricata** para inspección profunda de paquetes, y combínalo con un motor SIEM o dashboards como **ELK Stack** o **Grafana** para visualizar anomalías en tiempo real.

🚩 **Objetivo realista:** Detectar C2s tempranos, movimientos laterales internos sospechosos y exfiltraciones de datos mediante análisis pasivo del tráfico de red.

Técnica 13: Uso de Time-Based Hunting para eventos recientes

Cuando estás investigando un incidente o recibes una alerta crítica, buscar en meses de logs puede ser abrumador y poco eficiente. Por eso, el **Time-Based Hunting** se enfoca en analizar solo los eventos ocurridos en un rango corto de tiempo (por ejemplo, las últimas 6 a 48 horas), lo que mejora drásticamente la precisión y velocidad del análisis.

La idea es **correlacionar logs, procesos, conexiones y actividades sospechosas justo antes y después de una alerta conocida**, permitiendo identificar patrones relacionados como ejecuciones remotas, creación de usuarios o accesos inusuales. Esto también ayuda a descartar falsos positivos rápidamente, ya que te centras únicamente en lo relevante desde el punto de vista temporal.

🔧 **Herramienta clave:** Usa interfaces como Kibana (en entornos ELK) o dashboards integrados de tu SIEM (como Splunk o Microsoft Sentinel) para aplicar filtros temporales avanzados y cruzar datos de endpoints, red y autenticaciones en un periodo ajustado.

📌 **Objetivo realista:** Reducir tiempo de investigación, aumentar la señal frente al ruido y mejorar la capacidad de encontrar IOCs relevantes en medio de grandes volúmenes de datos.

Técnica 14: Hunting usando Indicators of Behavior (IoBs) en lugar de IoCs

Fiarse solo de los **Indicadores de Compromiso (IoCs)** como IPs, hashes o dominios maliciosos ya no es suficiente ante amenazas avanzadas y técnicas evasivas. En su lugar, esta técnica se enfoca en los **Indicadores de Comportamiento (IoBs)**, que identifican patrones de acción sospechosos generados durante una intrusión, independientemente de la firma usada. Por ejemplo, detectar el uso combinado de **PowerShell + WMI + modificación del registro de Windows** puede indicar una técnica de persistencia típica de atacantes avanzados. Usando el marco **MITRE ATT&CK**, se mapea cada secuencia de comportamiento a tácticas específicas (como Execution, Persistence o Defense Evasion), permitiendo anticiparse a los pasos siguientes del atacante.

🔧 **Metodología clave:** Usar herramientas EDR/SIEM que permitan correlacionar eventos entre endpoints y usuarios, y aplicar reglas basadas en comportamiento (behavioral analytics) para detectar cadenas de acciones sospechosas incluso si ningún elemento individual parece malicioso.

📌 **Objetivo realista:** Detectar amenazas avanzadas antes de que se materialicen completamente, basándose en cómo actúan los atacantes, no solo en lo que dejan atrás.

Técnica 15: Análisis de logs EDR/SIEM con lenguaje SQL-like

Para descubrir patrones ocultos entre grandes volúmenes de datos de seguridad, es fundamental usar lenguajes de consulta avanzados como **KQL (en Microsoft Sentinel)**, **SPL (en Splunk)** o **Lucene (en Elasticsearch)**. Estos permiten realizar búsquedas complejas, correlacionar eventos entre múltiples fuentes y detectar actividades sospechosas que un humano difícilmente encontraría revisando logs manualmente.

Por ejemplo, puedes escribir una consulta que busque **“múltiples intentos de inicio de sesión fallidos desde la misma IP en menos de 1 minuto”**, lo cual puede indicar un ataque de fuerza bruta. O incluso rastrear secuencias de comandos peligrosos ejecutados por usuarios inusuales, ayudando a detectar insiders o credenciales robadas.

🔧 **Herramienta clave:** Aprende a dominar **KQL** si usas Microsoft Sentinel, **SPL** si trabajas con Splunk, y **Lucene/Elasticsearch DSL** si estás en entornos basados en ELK Stack. Cada uno tiene su curva de aprendizaje, pero son herramientas indispensables para cualquier analista de seguridad moderno.

🚀 **Objetivo realista:** Automatizar la detección de amenazas complejas mediante consultas estructuradas, mejorando visibilidad, reduciendo falsos positivos y acelerando la investigación de incidentes.

Técnica 16: Uso de repositorios de TI para almacenar imágenes de disco forense

Durante una investigación forense o un incidente crítico, es fundamental preservar copias integrales del estado original del disco afectado. Esto permite realizar análisis posteriores, cumplir con requisitos legales o incluso volver a examinar el sistema si aparecen nuevos indicios tiempo después. Para hacerlo correctamente, se utilizan herramientas especializadas como **FTK Imager**, **LinEn** o **X-Ways Forensics**, capaces de generar imágenes forenses en formatos validables (como E01 o DD), asegurando integridad mediante hash SHA-256. Una vez creada la imagen, debe ser almacenada en repositorios seguros de TI, preferiblemente **encriptados y con firmas digitales**, para garantizar su autenticidad y protección contra modificaciones no autorizadas.


🔧 **Práctica clave:** Integra este proceso dentro del flujo estándar de respuesta ante incidentes, usando almacenamiento en servidores dedicados con control de acceso y logs de auditoría para mantener la cadena de custodia digital.


🚀 **Objetivo realista:** Garantizar disponibilidad de evidencia intacta para análisis futuro, cumplimiento legal y posibles auditorías post-incidente.

Técnica 17: Limpieza segura de sistemas después de IR (Incident Response)

Una vez que un incidente ha sido contenido y remediado, es fundamental realizar una **limpieza exhaustiva del sistema afectado** para garantizar que no queden residuos del ataque, como puertas traseras persistentes, usuarios maliciosos o certificados SSL falsos. Este paso es crítico para prevenir reinfecciones y asegurar que el entorno esté completamente restaurado a un estado seguro.

El proceso comienza revisando logs detallados de herramientas como **Sysmon**, donde se pueden identificar procesos sospechosos, conexiones inusuales o escrituras en rutas poco comunes. Además, se deben ejecutar **escaneos completos con antivirus actualizados** y comprobar manualmente cualquier anomalía detectada. Es especialmente importante validar qué **certificados SSL/TLS nuevos fueron instalados**, si existen **usuarios o grupos creados recientemente**, o si hay **servicios ocultos** que podrían estar funcionando como C2s o mecanismos de persistencia.

 **Herramienta clave:** Usa herramientas como **Autoruns**, **Process Explorer** o **Sysmon** junto con escáneres AV empresariales para revisar profundamente todo el sistema. Para automatización y consistencia, sigue checklists validadas como las del **NIST IR (Incident Response)**, que ofrecen un marco estructurado para asegurar cada punto crítico tras un incidente.

 **Objetivo realista:** Garantizar que ningún artefacto malicioso permanezca activo tras la remediación, protegiendo así la integridad del sistema y evitando futuros ataques recurrentes.

Técnica 18: Uso de playbooks estandarizados para responder a incidentes

Para agilizar la respuesta ante incidentes comunes como **phishing**, **ransomware** o **intrusiones detectadas**, es fundamental contar con **playbooks estandarizados**. Estos son guías estructuradas que definen paso a paso cómo actuar frente a cada tipo de amenaza, reduciendo tiempos de reacción y minimizando errores bajo presión.

Un playbook bien formado incluye:

- **Clasificación del evento** : ¿Qué tipo de incidente es? ¿De alto, medio o bajo impacto?
- **Acciones iniciales** : Contención, aislamiento del host, recolección de evidencia volátil.
- **Investigación** : Análisis de logs relevantes, búsqueda de IOCs, correlación con SIEM/EDR.
- **Resolución** : Eliminación del malware, limpieza del sistema, restauración segura.
- **Documentación** : Reporte detallado del incidente, lecciones aprendidas y recomendaciones técnicas.

✂ **Herramienta clave:** Usa plataformas colaborativas como **Confluence** para mantener los playbooks actualizados y accesibles, e intégralos con herramientas **SOAR (Security Orchestration, Automation and Response)** como **Cortex XSOAR** o **TheHive** , para ejecutar automáticamente ciertos pasos críticos.

🚀 **Objetivo realista:** Convertir la respuesta a incidentes en un proceso eficiente y repetible, mejorando la madurez operativa del equipo de seguridad.

Técnica 19: Uso de herramientas Open Source de ciberseguridad en producción

Las herramientas de código abierto ofrecen una alternativa poderosa, flexible y gratuita a soluciones comerciales, especialmente útiles para equipos con presupuesto limitado o iniciativas de defensa interna. Su principal ventaja es su **bajo costo** , **comunidad activa de desarrollo** y alta capacidad de **personalización** para adaptarse a necesidades específicas de cada entorno.

Ejemplos clave incluyen:

- **ELK Stack + Beats** para recolección, análisis y visualización avanzada de logs.
- **Zeek (anteriormente Bro)** para inspección profunda de tráfico de red y detección de anomalías.
- **Velociraptor** para investigación forense remota y recopilación de artefactos a gran escala.
- **Osquery** para monitoreo en tiempo real de hosts como si fueran bases de datos SQL.

🔧 **Uso recomendado:** Aunque requieren más curva de aprendizaje, estas herramientas pueden integrarse en entornos productivos con scripts personalizados, automatización y dashboards interactivos. Ideal tanto para pequeñas empresas como para ampliar capacidades en SOC corporativos.

🚩 **Objetivo realista:** Implementar controles avanzados de seguridad sin depender exclusivamente de software comercial, manteniendo calidad técnica y escalabilidad operativa.

Técnica 20: Integración de feeds de threat intelligence en el día a día

Incorporar inteligencia sobre amenazas (Threat Intelligence) no es un extra, sino una necesidad para mejorar continuamente las capacidades de detección. Usar feeds actualizados de fuentes confiables como **VirusTotal Intelligence**, **AlienVault OTX**, **Mandiant** o incluso **Shodan** , permite identificar nuevas tácticas de atacantes, hosts maliciosos, dominios C2 y firmas de malware activas en el mundo real.

Estas fuentes se pueden integrar directamente en tu plataforma de defensa mediante herramientas como **MISP (Malware Information Sharing Platform)** o **TheHive** , donde los indicadores de compromiso (IOCs) se cruzan automáticamente con tus logs de red, endpoints y SIEM.

Esto facilita la priorización de alertas, la creación de reglas proactivas y la correlación de eventos relacionados con campañas conocidas.

🔧 **Herramienta clave:** Usa **MISP** para centralizar y compartir inteligencia entre equipos, e integra estos datos con tu EDR/SIEM para actualizar reglas de detección de forma continua.

🚩 **Objetivo realista:** Mejorar la capacidad de detección usando inteligencia operativa fresca, permitiendo defenderse no solo de lo ya conocido, sino también de lo que está sucediendo ahora mismo en otros entornos.

Técnica 21: Monitoreo continuo de vulnerabilidades conocidas en tu infraestructura

No todas las vulnerabilidades representan un riesgo real para tu entorno. Esta técnica se enfoca en el **monitoreo proactivo de CVEs conocidas**, usando herramientas como **Qualys, Nessus o OpenVAS**, para identificar cuáles están presentes en tu infraestructura y qué tan probable es que sean explotadas en el mundo real.

Para priorizar eficazmente, se integran métricas como el **Exploit Prediction Scoring System (EPSS)**, que asigna probabilidades de explotación a cada CVE, permitiendo enfocar los esfuerzos en las correcciones que realmente importan. Estos datos deben estar conectados a una plataforma central como un **SIEM o CMDB**, para cruzar información de activos, criticidad y exposición real.

🔧 **Herramienta clave:** Automatiza escaneos periódicos y usa alertas basadas en **EPSS + criticidad del activo** para priorizar parcheo inteligente, no solo reactivo. Herramientas como **Tenable.cs** o **Rapid7 InsightVM** ofrecen integraciones avanzadas con feeds de threat intelligence.

🎯 **Objetivo realista:** Convertir el proceso de gestión de vulnerabilidades en algo dinámico, contextualizado y enfocado en reducir el riesgo real, no solo en cumplir checklist genéricas.

Técnica 22: Simulación de ataques para validar controles existentes

Para comprobar si tus defensas realmente funcionan, no basta con confiar en teoría o reportes pasivos. Es fundamental realizar simulaciones controladas de ataques reales usando un **Red Team autorizado**, aplicando técnicas comunes como **phishing seguido de movimiento lateral**, abuso de **LOLBins (Living Off The Land Binaries)** y **escalada de privilegios**, todo esto bajo entornos supervisados y sin poner en riesgo operaciones críticas.

Herramientas como **Caldera** (framework automatizado basado en MITRE ATT&CK) y **Atomic Red Team** (conjunto de pruebas pequeñas y reproducibles) permiten ejecutar estas simulaciones con alta precisión y sin necesidad de desarrollar exploits complejos desde cero. Además, facilitan la generación de informes sobre qué controles detectaron cada acción y cuáles fallaron silenciosamente.

✂ **Uso recomendado:** Integra estos ejercicios dentro de tu ciclo de mejora continua, usando los resultados para ajustar reglas SIEM/EDR, mejorar playbooks SOAR y entrenar a tu equipo Blue Team con escenarios reales y medibles.

📌 **Objetivo realista:** Validar la eficacia de tus controles de seguridad desde una perspectiva ofensiva, identificar brechas ocultas y aumentar la madurez de tu postura de defensa.

Técnica 23: Uso de Endpoint Detection and Response (EDR) para hunting remoto

Las soluciones **EDR modernas** no solo sirven para detección pasiva, sino también como herramientas poderosas para realizar **búsqueda proactiva de amenazas desde consolas centralizadas**, incluso sin acceso directo a los equipos afectados. Esta técnica permite a los analistas ejecutar investigación en tiempo real en endpoints distribuidos, mejorando drásticamente la capacidad de respuesta ante amenazas avanzadas.

Funcionalidades clave incluyen:

- **Remote shells** : Ejecutar comandos de forma segura y controlada en máquinas remotas para recolectar información o buscar artefactos sospechosos.
- **Live forensics** : Analizar memoria, procesos activos, conexiones de red y claves del registro en tiempo real.
- **Behavioral analytics** : Detectar secuencias de comportamiento anómalas basadas en tácticas MITRE ATT&CK, más allá de simples firmas.

✂ **Ejemplos destacados:** Plataformas como **CrowdStrike Falcon**, **SentinelOne Singularity** o **Microsoft Defender for Endpoint** ofrecen estas capacidades integradas, permitiendo escalar el threat hunting a toda la infraestructura con una sola interfaz.


📌 **Objetivo realista:** Usar el EDR no solo para alertar, sino para investigar proactivamente, reduciendo el tiempo entre compromiso y detección.


Técnica 24: Análisis de archivos adjuntos con sandboxing multifactor

Para detectar documentos maliciosos que evaden controles básicos de correo o AVs, es fundamental usar entornos de sandboxing avanzados que simulen condiciones reales de ejecución. Esta técnica se centra en el análisis profundo de archivos adjuntos comunes como **PDFs con JavaScript malicioso**, **documentos Word con OLE links** o **archivos comprimidos que contienen ejecutables embebidos**, todo esto sin arriesgar la red corporativa.

El análisis multifactor implica no solo observar el comportamiento del archivo al abrirse, sino también monitorear:

- Llamadas a APIs sospechosas o inyecciones de memoria.
- Modificaciones en el registro de Windows o creación de servicios.
- Intentos de conexión a dominios o IPs externos (C2).
- Descargas secundarias o ejecución de scripts PowerShell/WMI.

 **Herramienta clave:** Aunque ANY.RUN es una excelente opción visual y rápida, también puedes usar plataformas como **CAPE (Custom Analyst Packet Emulator)**, **Hybrid-Analysis**, **ANY.RUN API** o **Joe Sandbox**, ideales para integraciones automatizadas, análisis masivo y generación de informes con IOCs listos para tu SIEM/EDR.


 **Objetivo realista:** Detectar payloads ofuscados, cargadores maliciosos y campañas de spear phishing antes de que lleguen a los usuarios finales, mejorando drásticamente la protección contra amenazas desconocidas o zero-day.


Técnica 25: Uso de métricas SMART para evaluar el rendimiento del equipo

Para medir realmente cómo está funcionando tu equipo de seguridad (y mejorar sobre hechos, no percepciones), es clave usar **métricas SMART**: específicas, medibles, alcanzables, relevantes y con fecha límite. Esto permite transformar áreas como el threat hunting, la respuesta a incidentes o la detección proactiva en procesos maduros, optimizables y orientados a resultados.

Ejemplos prácticos incluyen:

- **MTTR (Mean Time to Respond):** Cuánto tiempo tarda tu equipo en responder desde que se genera una alerta real.
- **% de falsos positivos:** Qué porcentaje de alertas generadas terminan siendo descartadas, ayudando a ajustar reglas y filtros.
- **Cobertura de tácticas MITRE ATT&CK:** Cuántas tácticas y técnicas de ataque reales estás cubriendo con tus controles actuales.

 **Plataforma clave:** Usa herramientas como **Prometheus** para recolectar datos automáticamente desde SIEMs, EDRs y playbooks SOAR, y visualiza todo en dashboards interactivos con **Grafana**, actualizados en tiempo real para facilitar la toma de decisiones basada en datos.

 **Objetivo realista:** Transformar la gestión del equipo Blue Team en un proceso medible, comparable y enfocado en mejorar constantemente su eficacia operativa frente a amenazas reales.

Técnica 26: Capacitación continua del equipo con CTI y simulaciones

La ciberseguridad es un campo en constante evolución, y mantener a tu equipo actualizado no es opcional: es una necesidad crítica. Esta técnica se enfoca en la **capacitación activa y práctica**, combinando ejercicios técnicos como CTFs (Capture The Flag), laboratorios controlados y análisis de inteligencia sobre amenazas reales para mejorar continuamente las habilidades del equipo Blue Team.

Métodos efectivos incluyen:

- Participar en **CTFs corporativos o públicos** para simular escenarios ofensivos y defensivos bajo presión.
- Usar plataformas como **TryHackMe**, **HackTheBox** o **OverTheWire**, donde se ofrecen entornos guiados y máquinas vulnerables para practicar desde hacking básico hasta técnicas avanzadas de evasión.
- Estudiar regularmente **informes de threat intelligence** de fuentes confiables (como Mandiant, CrowdStrike o AlienVault OTX) para entender cómo operan los grupos APT y adaptar defensas a tácticas reales.

🔧 **Uso recomendado:** Integra esta práctica dentro del flujo de trabajo semanal o mensual del equipo, asignando horas dedicadas a formación técnica y evaluando progresos mediante simulaciones internas o exámenes prácticos.

📌 **Objetivo realista:** Mantener a tu equipo operativo siempre preparado para enfrentar nuevas amenazas, mejorando su capacidad analítica, técnica y de respuesta ante incidentes reales.

Técnica 27: Uso de reportes estandarizados de incidentes (IR Reports)

Una vez finalizado un incidente de seguridad, la documentación no es solo una formalidad: es una herramienta crítica para mejorar futuras respuestas y justificar acciones tomadas. Usar **reportes estandarizados de incidentes (IR Reports)** permite presentar hallazgos técnicos de forma clara, estructurada y útil tanto para equipos internos como para stakeholders externos o auditorías legales.

Un buen reporte debe incluir:

- **Información general del evento:** Tipo de ataque, sistema afectado, clasificación de severidad (por ejemplo, usando CVSS o SLAs internos).
- **Timeline detallado:** Desde el primer indicio hasta la resolución completa, mostrando cada paso clave del proceso de respuesta.
- **Hallazgos técnicos:** IOCs identificados, vulnerabilidades explotadas, procesos maliciosos, usuarios comprometidos y persistencias encontradas.
- **Acciones realizadas:** Qué se hizo durante la contención, eliminación y recuperación del sistema afectado.
- **Recomendaciones:** Acciones concretas para prevenir incidentes similares, desde parcheo urgente hasta reglas SIEM/EDR adicionales o capacitación a usuarios.

🔧 **Formato sugerido:** Usa la **plantilla NIST SP 800-61 rev4**, reconocida internacionalmente, que cubre todos los aspectos necesarios para crear informes completos, objetivos y útiles para análisis posteriores o cumplimiento regulatorio.

📌 **Objetivo realista:** Convertir cada incidente en una lección aprendida, mejorando continuamente no solo la defensa técnica, sino también la comunicación y gobernanza de seguridad dentro de la organización.


Técnica 28: Bloqueo proactivo de URLs maliciosas con proxies de seguridad


Una de las capas más efectivas de defensa en cualquier entorno corporativo es el **bloqueo proactivo de URLs maliciosas a través de proxies de seguridad gestionados**. Esta técnica se basa en interceptar y filtrar todo el tráfico web saliente antes de que llegue al usuario final, evitando que accedan a sitios conocidos por ser maliciosos o peligrosos.

Sistemas como **Zscaler Internet Access (ZIA)**, **Palo Alto Prisma Access** o **Cisco SecureX Web** permiten aplicar políticas granulares de categorización web, integración con feeds de threat intelligence y bloqueo automatizado de dominios C2, campañas de phishing o servidores de malvertising.

Los tipos de bloqueo más útiles incluyen:

- **Sitios C&C (Command & Control):** Dominios usados por malware para recibir órdenes remotas.
- **Malvertising:** Anuncios maliciosos que redirigen a exploits o payloads.
- **Phishing reports:** Dominios registrados recientemente que coinciden con plantillas típicas de phishing o aparecen en listas como Google Safe Browsing o VirusTotal.

 **Uso recomendado:** Integra estas soluciones con tu plataforma SIEM/EDR para recibir alertas en tiempo real sobre intentos de acceso bloqueados, lo que puede ayudarte a identificar hosts infectados o usuarios comprometidos.

 **Objetivo realista:** Reducir drásticamente el riesgo de infecciones iniciales causadas por navegación web, uno de los vectores de entrada más comunes en ataques reales.

Técnica 29: Diseño de arquitecturas defensivas con principios de least privilege

Una de las estrategias más efectivas para reducir la superficie de ataque es diseñar arquitecturas basadas en el principio de **"least privilege"**, es decir, garantizar que **cada usuario, servicio o aplicación tenga únicamente los permisos necesarios** para realizar sus funciones, y nada más. Este enfoque no solo limita los daños en caso de compromiso, sino que previene escaladas de privilegios y movimientos laterales dentro de la red.

Ejemplos prácticos incluyen:

- Asignar a los usuarios cuentas estándar sin permisos administrativos, incluso si su rol lo requiere temporalmente, usando técnicas como **UAC split-admin** o ejecución bajo credenciales elevadas solo cuando sea absolutamente necesario.
- Configurar correctamente el **User Account Control (UAC)** para evitar elevaciones silenciosas y asegurar que cada acción sensible requiera confirmación explícita.
- Aplicar **GPOs (Group Policy Objects) estrictos** que bloqueen ejecuciones no autorizadas, acceso remoto innecesario, instalación de software desconocido y modificaciones a configuraciones críticas del sistema operativo.

🔧 **Herramienta clave:** Usa soluciones como **Microsoft LAPS (Local Administrator Password Solution)** , políticas de protección de contraseñas avanzadas y monitoreo continuo de cambios en Active Directory para mantener el control sobre los privilegios asignados.

🚩 **Objetivo realista:** Eliminar accesos innecesarios antes de que se conviertan en vectores de ataque, fortaleciendo la postura de seguridad general y cumpliendo con estándares como **Zero Trust** y marcos regulatorios como **NIST** o **ISO 27001** .

Técnica 30: Integración de seguridad en DevOps con DevSecOps

Incorporar seguridad desde las primeras etapas del ciclo de desarrollo es clave para evitar vulnerabilidades críticas en producción. Esta técnica se enfoca en integrar controles de seguridad automatizados dentro de los **pipelines CI/CD** , garantizando que cada cambio de código sea revisado antes de llegar a producción.

Prácticas clave incluyen:

- **Static Code Analysis (SCA)** para detectar dependencias inseguras o componentes con CVEs conocidas en tiempo de build.
- Aplicar **SAST (Static Application Security Testing)** y **DAST (Dynamic Application Security Testing)** automatizado con herramientas como **SonarQube**, **Checkmarx** o **OWASP ZAP** , para identificar problemas de seguridad en el código fuente y en APIs/endpoints expuestos.
- Validar configuraciones seguras, firmar artefactos de construcción y escanear imágenes Docker antes de su despliegue, integrando seguridad sin frenar la velocidad de entrega.

🔧 **Herramienta clave:** Usa plataformas como **SonarQube** para análisis estático continuo, **Checkmarx** para revisiones profundas en repositorios empresariales, y **OWASP ZAP** o **Nuclei** para pruebas dinámicas automatizadas sobre servicios web recién desplegados.

🚀 **Objetivo realista:** Eliminar el "shift-left gap" entre desarrollo y seguridad, asegurando que solo código seguro llegue a producción, reduciendo riesgo y costos asociados a correcciones posteriores.

AVISO ETICO ,LEGAL Y RESPONSABLE :

Este documento está diseñado únicamente para uso profesional en entornos de defensa cibernética. Las técnicas aquí descritas deben aplicarse bajo autorización explícita, cumpliendo con políticas internas de seguridad y normativas legales vigentes. Queda prohibido su uso para actividades maliciosas o sin supervisión técnica adecuada. El conocimiento es poder, pero el poder debe usarse con responsabilidad.

Espero que esta guía te sirva tanto como a mí en mi día a día. Si te gustó, compártela con quien pueda aprovecharla. ¡Formemos más Blue Teams bien armados, técnicos sólidos y con ganas de defender de verdad!

Unai Rubio

Ciberseguridad | Blue Team | Threat Hunting | EDR/SIEM/EDR

Autor de *"30 Técnicas Blue Team para 2025"*

es Bilbao, España

© 2025 Unai Rubio — Todos los derechos reservados.

Este documento está pensado únicamente para uso profesional y defensivo.

No debe emplearse fuera de entornos autorizados ni con fines maliciosos.
