Módulo 2. Análisis de Protocolos (Wireshark + Zeek)

Introducción

El análisis de protocolos constituye una competencia central dentro de los enfoques contemporáneos de Network Detection and Response (NDR), ya que permite transformar el tráfico de red en evidencia estructurada y accionable. En este contexto, herramientas como Wireshark y Zeek cumplen funciones complementarias: la primera orientada a la inspección detallada de paquetes, y la segunda al registro sistemático de metadatos y eventos derivados del flujo de comunicaciones. Wireshark opera como analizador de paquetes capaz de presentar los datos capturados con un alto nivel de detalle protocolar (Sharpe, Warnicke & Lamping, 2023), mientras que Zeek se define como un framework de análisis de red centrado en la generación de registros estructurados que describen el comportamiento de los protocolos observados (The Zeek Project, 2023). Integradas en entornos como Security Onion, estas capacidades permiten correlacionar logs, identificar anomalías y sustentar procesos de caza de amenazas (Security Onion Solutions, 2024).

Unidad 1. Análisis – Wireshark, Zeek, logs y scripting

El análisis de protocolos en el marco de la Detección y Respuesta en Redes (NDR) se fundamenta en la capacidad de observar, registrar e interpretar las comunicaciones que atraviesan una infraestructura. En este sentido, Wireshark y Zeek constituyen herramientas complementarias: la primera orientada a la inspección granular de paquetes individuales, y la segunda concebida como un framework de análisis de red que transforma el tráfico en registros estructurados de alto nivel (The Zeek Project, 2023). La articulación entre captura detallada y modelado sistemático de eventos permite construir una visión integral del comportamiento protocolar y sustentar procesos de caza de amenazas basados en evidencia técnica.

Modelos de logs (conn, dns, http, ssl)

El análisis estructurado de tráfico de red en entornos de *Network Detection and Response (NDR)* descansa en la

capacidad de transformar flujos de paquetes en representaciones semánticas interpretables. Zeek fue diseñado precisamente con este objetivo: no centrarse únicamente en la captura de paquetes, sino en la generación de eventos y registros estructurados que describen el comportamiento de los protocolos observados (The Zeek Project, 2023). Esta arquitectura lo distingue de herramientas de inspección puramente forense como Wireshark, cuya finalidad principal es permitir la visualización detallada de cada paquete capturado (Sharpe, Warnicke & Lamping, 2023).

Mientras Wireshark organiza la información en torno a paquetes individuales y sus campos protocolarios, Zeek produce modelos de logs que representan abstracciones de mayor nivel. Estos modelos permiten sintetizar múltiples intercambios de paquetes en entidades lógicas coherentes, facilitando análisis longitudinales, estadísticos y conductuales. Entre los registros más relevantes para la práctica NDR se encuentran conn.log, dns.log, http.log y ssl.log, definidos en los scripts base del framework (The Zeek Project, 2023).

conn.log: modelo de conexión como unidad analítica —

El archivo **conn.log**, definido en el script base/protocols/conn/main.zeek, constituye el eje central del modelo de observación de Zeek. Este registro resume cada conexión detectada, independientemente del protocolo de aplicación transportado. En lugar de enumerar cada paquete TCP o UDP, Zeek agrupa la interacción en una entidad denominada “conexión”,

caracterizada por campos estructurados tales como dirección IP de origen, dirección IP de destino, puertos, protocolo de transporte, duración, cantidad de bytes transferidos y estado final de la conexión (The Zeek Project, 2023).

Este modelo resulta metodológicamente relevante porque permite analizar el comportamiento de la red en términos de flujos, no de paquetes aislados. Desde la perspectiva del modelo OSI, una conexión TCP implica múltiples intercambios de segmentos; Wireshark permite inspeccionar cada uno de ellos en detalle, observando flags, números de secuencia y confirmaciones (Sharpe, Warnicke & Lamping, 2023). Sin embargo, para propósitos de detección, la abstracción de alto nivel proporcionada por conn.log facilita identificar patrones agregados como:

- Conexiones extremadamente breves y repetitivas.
- Transferencias asimétricas de datos.
- Estados de finalización anómalos.
- Incrementos inusuales en la frecuencia de flujos hacia un mismo destino.

La capacidad de observar el estado de la conexión (por ejemplo, si finalizó correctamente o fue interrumpida) adquiere particular importancia en investigaciones relacionadas con escaneos de puertos o intentos de explotación. Zeek registra explícitamente estos estados, permitiendo construir hipótesis sobre comportamientos sistemáticos (The Zeek Project, 2023).

En términos comparativos, mientras Wireshark permite reconstruir manualmente la secuencia de establecimiento y cierre de conexión —incluyendo el análisis del three-way handshake TCP—, Zeek consolida dicha información en campos sintéticos que agilizan la correlación a gran escala. Esta diferencia metodológica constituye una de las bases del enfoque NDR.

dns.log: registro estructurado de resolución de nombres

El protocolo DNS opera como mecanismo fundamental de resolución en redes IP, por lo que su análisis resulta crítico tanto para la detección de infraestructuras de comando y control como para la identificación de dominios generados algorítmicamente. Zeek implementa el registro dns.log a través del script base/protocols/dns/main.zeek, el cual intercepta eventos de consulta y respuesta DNS y los transforma en entradas estructuradas (The Zeek Project, 2023). Cada registro consolida información relevante como el nombre de dominio consultado, el tipo de registro solicitado, el código de respuesta, las direcciones IP devueltas y los identificadores de transacción.

Este modelo estructurado permite analizar tendencias de resolución sin necesidad de revisar cada paquete DNS de manera individual. En contraste, Wireshark posibilita examinar con detalle los campos del encabezado DNS, incluidos flags, identificadores y registros adicionales (Sharpe, Warnicke & Lamping, 2023). Sin embargo, en entornos de gran volumen, el análisis manual de paquetes individuales resulta poco escalable.

Desde la óptica del threat hunting, el modelo dns.log facilita la identificación de fenómenos como una elevada proporción de respuestas NXDOMAIN, consultas hacia dominios poco frecuentes, patrones repetitivos de subdominios o resoluciones dirigidas a direcciones IP inusuales. La documentación oficial de Zeek subraya que el diseño de estos logs busca proporcionar una representación coherente y extensible del comportamiento protocolar, favoreciendo análisis sistemáticos y correlacionables en el tiempo (The Zeek Project, 2023).

http.log: semántica de aplicación estructurada

El protocolo HTTP, ubicado en la capa de aplicación del modelo OSI, expone múltiples campos que pueden revelar patrones conductuales. Mitkov (2021), al analizar el tráfico HTTP con Wireshark, demuestra que la disección de cabeceras permite identificar tanto funcionamiento legítimo como vulnerabilidades explotables. Campos como método, URI, agente de usuario y códigos de estado contienen información relevante para la detección de anomalías.

Zeek formaliza esta observación en el registro http.log, definido en base/protocols/http/main.zeek (The Zeek Project, 2023). Este archivo consolida los elementos centrales de cada transacción HTTP en campos estructurados. De esta manera, múltiples paquetes que conforman una solicitud y su correspondiente respuesta quedan representados en una única entrada.

La ventaja analítica de este modelo radica en la posibilidad de realizar consultas sobre: métodos HTTP inusuales, secuencias repetitivas de solicitudes, combinaciones específicas de código de estado y agentes de usuario inconsistentes.

Mientras Wireshark permite examinar cada encabezado manualmente, Zeek facilita búsquedas masivas y correlaciones temporales. Esta capacidad es especialmente relevante en contextos NDR donde el volumen de tráfico puede alcanzar millones de eventos diarios.

ssl.log: parámetros criptográficos como indicadores

El protocolo TLS —sucesor de SSL— introduce mecanismos de cifrado y autenticación basados en criptografía asimétrica y negociación de suites criptográficas. Mitkov (2021) explica que el análisis del handshake TLS permite comprender la estructura de intercambio de claves y certificados, aspecto que puede observarse detalladamente en Wireshark.

Zeek implementa el registro ssl.log (actualmente denominado tls.log en versiones recientes) para capturar metadatos del handshake, tales como versión del protocolo, suite criptográfica seleccionada y características del certificado presentado (The Zeek Project, 2023). Esta información resulta crucial para identificar configuraciones débiles o comportamientos anómalos.

La abstracción que ofrece Zeek no reemplaza el análisis forense de paquetes, sino que lo complementa. Cuando se detecta una versión obsoleta de TLS en el log estructurado, el analista puede recurrir a la captura detallada en Wireshark para examinar el intercambio exacto del ClientHello y ServerHello (Sharpe, Warnicke & Lamping, 2023).

Integración conceptual de los modelos —

Los cuatro modelos —conn, dns, http y ssl— conforman un sistema jerárquico de observación. conn.log proporciona la base de flujo; dns.log revela la resolución previa; http.log describe la interacción de aplicación; ssl.log aporta información criptográfica. Esta estratificación reproduce, de manera operativa, la lógica del modelo OSI descrita por Mitkov (2021), trasladando la separación de capas al ámbito del registro estructurado.

Desde la perspectiva NDR, la comprensión profunda de estos modelos permite formular hipótesis basadas en comportamiento y no únicamente en firmas estáticas. La documentación oficial de Zeek subraya que el framework fue diseñado precisamente para habilitar este tipo de análisis conductual extensible (The Zeek Project, 2023).

Búsquedas típicas de caza

La caza de amenazas (threat hunting) en entornos de NDR no se basa únicamente en la detección reactiva de alertas, sino en la formulación de hipótesis sobre comportamientos potencialmente anómalos que luego se contrastan contra la telemetría disponible. En el contexto de Zeek, esa telemetría está representada por los modelos de logs estructurados previamente analizados. El valor de estos registros radica en

que permiten interrogar el tráfico histórico desde una perspectiva conductual y no exclusivamente basada en firmas.

El enfoque de hunting parte de la premisa de que la actividad maliciosa suele manifestarse como desviaciones respecto del comportamiento habitual. El eBook de Endpoint Threat Hunting subraya que el análisis proactivo requiere identificar patrones que no necesariamente activan mecanismos automáticos de alerta, pero que reflejan anomalías estadísticas o contextuales (eBook Endpoint Threat Hunting, s.f.). Trasladado al plano de red, esto implica examinar características agregadas de conexiones, resoluciones DNS o transacciones HTTP.

En conn.log, la unidad analítica es la conexión como flujo lógico. Desde esta perspectiva, el analista puede formular hipótesis tales como: “¿existen conexiones extremadamente breves y repetitivas hacia múltiples destinos?” o “¿se observan transferencias de datos con asimetría marcada entre bytes enviados y recibidos?”. Zeek registra explícitamente duración, volumen de datos y estado final de cada conexión (The Zeek

Project, 2023), lo que permite detectar patrones compatibles con escaneos automatizados, intentos fallidos de autenticación o posibles exfiltraciones.

Wireshark, en cambio, exigiría reconstruir manualmente múltiples sesiones TCP para llegar a conclusiones equivalentes (Sharpe, Warnicke & Lamping, 2023). En un entorno de alto volumen, ese enfoque no resulta escalable. La abstracción de Zeek posibilita consultas agregadas que permiten identificar rápidamente desviaciones en el comportamiento global de la red.

En el ámbito de dns.log, las búsquedas suelen orientarse a identificar resoluciones anómalas. Dado que Zeek registra nombre consultado, tipo de registro y código de respuesta (The Zeek Project, 2023), el analista puede evaluar la proporción de respuestas negativas (NXDOMAIN), la recurrencia de subdominios variables o la aparición de dominios inéditos en el histórico de la organización. Estas búsquedas no requieren inspeccionar el contenido completo de cada paquete DNS, sino analizar el comportamiento agregado de resolución.

La lógica es similar en http.log. Zeek consolida método, URI, código de estado y agente de usuario en un único registro estructurado (The Zeek Project, 2023). Desde el hunting, esto

permite detectar combinaciones inusuales, como agentes de usuario atípicos que acceden repetidamente a rutas específicas o patrones repetitivos de solicitud con códigos de error sistemáticos. Mitkov (2021) demuestra que el análisis detallado de cabeceras HTTP a nivel de paquete permite identificar comportamientos anómalos; Zeek traslada esa misma capacidad a un plano estructurado y escalable.

En ssl.log o tls.log, las búsquedas suelen centrarse en parámetros criptográficos. La documentación del Book of Zeek indica que versión del protocolo y suite criptográfica negociada quedan registradas explícitamente (The Zeek Project, 2023). Esto permite detectar el uso de versiones obsoletas o combinaciones criptográficas que no se ajustan a la política organizacional. Tales desviaciones pueden indicar configuraciones inseguras o incluso intentos de evasión.

Un aspecto metodológico central es la correlación entre logs. Una resolución DNS hacia un dominio inusual puede vincularse con una conexión en conn.log y, posteriormente, con una transacción HTTP o una sesión TLS específica. Esta correlación reproduce la lógica estratificada del modelo OSI descrito por Mitkov (2021), pero aplicada a nivel de registro estructurado.

En síntesis, las búsquedas típicas de caza en Zeek no consisten en examinar paquetes individuales, sino en interrogar patrones agregados de comportamiento protocolar. Wireshark mantiene su relevancia como herramienta de validación forense puntual; sin embargo, el hunting sistemático se apoya principalmente en la capacidad de consulta y correlación que ofrecen los modelos de logs estructurados.

Scripts de enriquecimiento

Uno de los rasgos distintivos de Zeek respecto de otros analizadores de tráfico es su carácter programable. No se trata únicamente de un generador de logs predefinidos, sino de un framework cuyo comportamiento puede extenderse mediante scripting. La documentación oficial enfatiza que Zeek fue diseñado como una plataforma orientada a eventos, donde los protocolos generan eventos internos que pueden ser interceptados y procesados por scripts personalizados (The Zeek Project, 2023). Esta arquitectura transforma el análisis de red en un proceso formalizable y reproducible.

Los scripts base que definen registros como conn.log, dns.log o http.log ilustran cómo Zeek intercepta eventos protocolarios y los traduce en estructuras de datos exportables (The Zeek Project, 2023). A partir de esta base, el analista puede modificar o ampliar los registros existentes, añadir nuevos campos o generar alertas específicas cuando se cumplen determinadas condiciones. Este mecanismo permite trasladar hipótesis de detección al plano del código, evitando depender exclusivamente de análisis manuales posteriores.

El enriquecimiento puede consistir, por ejemplo, en incorporar etiquetas internas a determinadas conexiones, clasificar dominios según criterios organizacionales o agregar metadatos derivados del contexto operativo. En lugar de limitarse a almacenar dirección IP y puerto, el log puede ampliarse con categorías de riesgo o marcadores de interés definidos por la organización. Esta práctica resulta coherente con el enfoque de hunting descrito previamente, donde la detección se basa en comportamiento y contexto más que en firmas estáticas (eBook Endpoint Threat Hunting, s.f.).

Desde una perspectiva metodológica, el scripting en Zeek permite reducir la brecha entre observación y detección. En

lugar de consultar repetidamente los mismos patrones en búsquedas manuales, el analista puede codificar dichas reglas como lógica persistente. Esta formalización tiene implicancias relevantes en términos de escalabilidad y consistencia analítica. La documentación del framework destaca precisamente su capacidad de adaptación a entornos cambiantes mediante extensiones específicas (The Zeek Project, 2023).

Security Onion, al integrar Zeek dentro de un ecosistema más amplio de monitoreo, aprovecha esta extensibilidad para correlacionar eventos de red con otras fuentes de telemetría (Security Onion Solutions, 2024). El enriquecimiento no se limita entonces al propio log de red, sino que puede vincularse con datos adicionales del entorno. Esto refuerza la idea de que el scripting en Zeek no es una funcionalidad accesoria, sino un componente estructural del modelo NDR.

En contraste, Wireshark no fue concebido como un framework de enriquecimiento continuo, sino como una herramienta de inspección detallada de paquetes (Sharpe, Warnicke & Lamping, 2023). Si bien permite filtros avanzados y análisis profundos, su finalidad principal es la observación forense puntual. Zeek, en cambio, permite trasladar patrones de análisis al plano programático, consolidando un modelo de detección estructurada y sostenible en el tiempo.

En síntesis, los scripts de enriquecimiento constituyen el mecanismo mediante el cual Zeek trasciende el simple registro de tráfico y se convierte en un instrumento activo de detección. La capacidad de capturar eventos, modificar estructuras de log y añadir lógica personalizada convierte al framework en una herramienta central dentro de arquitecturas modernas de NDR.

Exportación y rotación

El análisis estructurado de protocolos en entornos NDR no se agota en la generación de logs; requiere también una gestión adecuada del ciclo de vida de esos registros. La exportación, almacenamiento y rotación constituyen dimensiones operativas esenciales para garantizar tanto la continuidad del monitoreo como la preservación de evidencia. Zeek genera múltiples archivos de log segmentados por tipo de protocolo y por intervalo temporal, lo que facilita su organización sistemática (The Zeek Project, 2023).

La rotación periódica de logs cumple una doble función. En primer lugar, previene la saturación de almacenamiento, especialmente en redes de alto volumen donde el tráfico puede producir millones de eventos diarios. En segundo lugar, permite delimitar ventanas temporales de análisis, aspecto clave en investigaciones forenses. Security Onion documenta mecanismos integrados para la gestión y rotación automática de logs generados por Zeek dentro de su arquitectura de monitoreo (Security Onion Solutions, 2024). Esta automatización garantiza que los registros se archiven de forma consistente sin interrumpir la recolección continua de datos.

Desde el punto de vista metodológico, la segmentación temporal facilita reconstrucciones cronológicas precisas. Cuando se investiga un incidente, el analista puede acotar el análisis a intervalos específicos, correlacionando entradas de conn.log, dns.log, http.log y ssl.log correspondientes al mismo período. La estructura homogénea de los logs de Zeek —basada en campos delimitados y formatos consistentes— favorece esta correlación (The Zeek Project, 2023).

La exportación de registros en formatos estructurados posibilita su integración con sistemas de análisis adicionales o plataformas de visualización. Esta interoperabilidad resulta coherente con el enfoque NDR, donde la información de red suele correlacionarse con otras fuentes de telemetría. La documentación de Security Onion destaca precisamente la integración de Zeek en un ecosistema más amplio de monitoreo y análisis (Security Onion Solutions, 2024).

Wireshark, por su parte, permite exportar capturas completas en diversos formatos preservando la integridad del tráfico observado (Sharpe, Warnicke & Lamping, 2023). Esta capacidad resulta especialmente relevante cuando se requiere conservar evidencia para análisis posteriores o compartir archivos con otros analistas. No obstante, a diferencia de Zeek, Wireshark no está diseñado como sistema continuo de generación de logs estructurados, sino como herramienta de captura y examen detallado.

La diferencia conceptual es significativa: Zeek produce registros persistentes orientados al análisis longitudinal, mientras que Wireshark produce archivos de captura que representan instancias específicas de tráfico. En el contexto NDR, ambos mecanismos son complementarios. La correcta rotación y exportación de logs garantiza que el análisis protocolar pueda sostenerse en el tiempo y que la evidencia

generada sea preservada bajo criterios de consistencia y trazabilidad.

En definitiva, la gestión adecuada de logs no constituye un aspecto meramente operativo, sino una condición estructural para que el análisis de protocolos sea sostenible, verificable y forensemente sólido.

CONTINUAR

Unidad 2. Casos de uso

DNS sospechoso y rare domains.

El protocolo DNS constituye uno de los vectores más utilizados para el establecimiento y mantenimiento de canales de comando y control. De acuerdo con el marco MITRE ATT&CK, dentro de la táctica de *Command and Control* (TA0011), los adversarios emplean protocolos legítimos para comunicarse con infraestructuras remotas, precisamente porque estos protocolos son esenciales para el funcionamiento normal de las redes y, por lo tanto, difícilmente bloqueados (MITRE ATT&CK, s.f.). DNS se destaca en este contexto por su ubicuidad y por el bajo nivel de inspección profunda que suele recibir en comparación con otros protocolos.

Desde el punto de vista del *Network Detection and Response*, el análisis DNS no debe centrarse exclusivamente en indicadores estáticos, sino en patrones conductuales. Raggi (2021) sostiene que el *threat hunting* parte de una hipótesis fundamentada acerca de una posible actividad maliciosa y progresa mediante la correlación de datos y el análisis

contextual. En este marco metodológico, el concepto de *rare domain* no implica automáticamente malicia, sino que identifica dominios con baja frecuencia o baja prevalencia histórica dentro del entorno observado. La rareza se convierte así en un criterio estadístico inicial que habilita una investigación más profunda.

Zeek facilita este enfoque mediante la generación estructurada de dns.log, donde se registran consultas, tipos de registro, códigos de respuesta y otros metadatos relevantes (The Zeek Project, 2023). Esta estructuración permite analizar comportamientos agregados sin necesidad de inspeccionar cada paquete individual. Por ejemplo, puede detectarse una alta cardinalidad de subdominios bajo un mismo dominio raíz, fenómeno común en técnicas de generación algorítmica de dominios (*Domain Generation Algorithms*, DGA). Asimismo, la repetición de respuestas NXDOMAIN puede indicar intentos sistemáticos de resolución hacia dominios inexistentes generados dinámicamente.

El análisis de rareza debe complementarse con una línea base del comportamiento normal de la organización. Aragonés Lozano (2024) enfatiza que la detección eficaz requiere modelar patrones benignos repetitivos para

poder identificar desviaciones significativas. En el contexto DNS, esto implica conocer cuáles dominios son habituales, cuáles servicios externos son utilizados regularmente y cuál es la distribución normal de consultas por host. Solo a partir de esta caracterización es posible distinguir entre un comportamiento inusual legítimo y una actividad potencialmente maliciosa.

Otro indicador relevante es la periodicidad. Elastic Security Labs describe cómo muchas familias de malware implementan mecanismos de beaconing utilizando protocolos comunes, incluyendo DNS, para comunicarse con servidores de control (Elastic Security Labs, s.f.). Estos mecanismos pueden presentar intervalos regulares o patrones temporales con pequeñas variaciones (jitter) para evadir detecciones simples basadas en frecuencia fija. Por ello, el análisis DNS sospechoso debe considerar no solo la frecuencia absoluta de consultas, sino también la regularidad temporal y la consistencia estructural de los nombres consultados.

Wireshark complementa este proceso permitiendo inspeccionar detalladamente los paquetes DNS cuando un dominio específico requiere validación profunda. La herramienta posibilita examinar flags, identificadores de

transacción y contenido completo del mensaje DNS, lo que resulta útil para confirmar anomalías detectadas previamente a nivel de log estructurado (Sharpe, Warnicke & Lamping, 2023). No obstante, su uso es típicamente posterior a la identificación inicial realizada mediante registros agregados, ya que el análisis paquete por paquete no es escalable en entornos de alto volumen.

En términos operativos, un caso típico puede describirse del siguiente modo: un host interno comienza a realizar consultas periódicas a subdominios con alta entropía bajo un dominio poco frecuente en la red corporativa. Zeek registra la actividad en dns.log; el analista identifica la rareza estadística y la regularidad temporal; posteriormente utiliza Wireshark para confirmar la estructura de los paquetes y descartar errores de implementación legítimos. Si la actividad se correlaciona con conexiones salientes registradas en conn.log, la hipótesis de un canal de comando y control adquiere mayor solidez.

En consecuencia, el análisis de DNS sospechoso dentro de un enfoque NDR combina modelado estructurado, análisis estadístico de rareza, evaluación temporal y verificación forense puntual. Este enfoque permite detectar infraestructuras adversarias previamente

desconocidas sin depender exclusivamente de listas de indicadores preexistentes, reduciendo así el tiempo de permanencia del atacante dentro de la red (Raggi, 2021).

HTTP/TLS anómalos

El análisis de tráfico HTTP y TLS plantea un desafío particular dentro del paradigma NDR, ya que gran parte de las comunicaciones contemporáneas se encuentran cifradas. Sin embargo, el cifrado no elimina la posibilidad de detección; desplaza el foco analítico desde el contenido hacia los metadatos y los patrones de comportamiento. MITRE ATT&CK señala que los adversarios suelen utilizar protocolos web estándar para establecer y mantener canales de comando y control, precisamente porque se integran naturalmente dentro del tráfico legítimo de una organización (MITRE ATT&CK, s.f.).

En el caso de HTTP, Zeek genera registros estructurados a través de http.log, consolidando método, URI, código de estado, agente de usuario y otros metadatos en una única entrada (The Zeek Project, 2023). Esta modelización permite identificar anomalías sin necesidad de inspeccionar cada paquete individual. Por ejemplo, la aparición de agentes de usuario inconsistentes con el parque tecnológico de la

organización puede constituir un indicador de actividad automatizada. Del mismo modo, secuencias repetitivas de solicitudes hacia una misma ruta con intervalos regulares pueden sugerir mecanismos de beaconing.

Elastic Security Labs describe cómo muchos implantes maliciosos emplean HTTP o HTTPS como canal de comunicación, generando patrones temporales relativamente estables, aunque a veces introducen variaciones deliberadas para evadir detecciones basadas exclusivamente en periodicidad exacta (Elastic Security Labs, s.f.). En este contexto, el análisis debe considerar no solo la frecuencia de solicitudes, sino también la distribución temporal y la coherencia estructural de las transacciones.

Cuando el tráfico se encuentra protegido mediante TLS, el contenido de aplicación deja de ser visible. No obstante, Zeek registra información relevante del handshake en ssl.log o tls.log, incluyendo versión del protocolo y suite criptográfica negociada (The Zeek Project, 2023). Estos metadatos permiten identificar configuraciones obsoletas o combinaciones atípicas

que podrían asociarse a herramientas automatizadas o bibliotecas específicas.

El concepto de TLS fingerprinting adquiere especial relevancia en este punto. El proyecto JA3 propone un método estandarizado para generar huellas digitales de clientes TLS a partir de parámetros observables en el mensaje ClientHello (Salesforce, s.f.). Estas huellas permiten identificar implementaciones específicas de bibliotecas criptográficas, independientemente del dominio al que se conecten. En un entorno NDR, la aparición de un fingerprint TLS no habitual en la organización puede indicar la ejecución de software no autorizado o potencialmente malicioso.

Desde el enfoque de threat hunting, el análisis de anomalías HTTP/TLS debe articularse con hipótesis explícitas. Raggi (2021) sostiene que el cazador formula una suposición fundamentada y luego la contrasta con múltiples fuentes de datos. Por ejemplo, una hipótesis podría ser: “Existe un host interno que mantiene comunicación persistente cifrada con infraestructura externa no habitual”. Esta hipótesis puede evaluarse correlacionando registros de conn.log, http.log y tls.log.

La modelización estadística del comportamiento normal resulta nuevamente indispensable. Aragonés Lozano (2024)

destaca que la detección eficaz requiere distinguir patrones benignos repetitivos de desviaciones significativas mediante técnicas de análisis de datos. En el caso de HTTP/TLS, esto implica conocer cuáles dominios y servicios web son utilizados regularmente, cuáles versiones de TLS predominan en la infraestructura y cuáles fingerprints TLS corresponden a aplicaciones corporativas legítimas.

Wireshark cumple un rol complementario en esta etapa. Aunque Zeek proporciona metadatos estructurados, Wireshark permite examinar detalladamente el handshake TLS, verificar la composición exacta del ClientHello y confirmar la validez de la huella digital observada (Sharpe, Warnicke & Lamping, 2023). De este modo, el análisis puede transitar desde una detección basada en comportamiento agregado hacia una validación técnica de bajo nivel.

Un escenario operativo ilustrativo puede describirse así: un equipo interno inicia conexiones TLS periódicas hacia un dominio externo poco frecuente. El fingerprint JA3 asociado no coincide con los clientes TLS habituales de la organización. Zeek registra la versión de protocolo y la suite criptográfica; el análisis temporal revela intervalos consistentes de comunicación. La correlación con conn.log muestra sesiones de duración regular con transferencia limitada de datos, lo que sugiere un posible canal de comando y control cifrado. La

inspección posterior en Wireshark confirma la estructura del handshake y descarta errores de configuración legítimos.

En consecuencia, el análisis de HTTP/TLS anómalos dentro del enfoque NDR se basa en la explotación sistemática de metadatos, fingerprints y patrones temporales. El cifrado no elimina la detectabilidad, sino que exige desplazar la atención hacia características estructurales y comportamentales que, correctamente modeladas, permiten identificar actividad adversaria incluso cuando el contenido permanece inaccesible.

Beaconing y long-connections

El concepto de beaconing ocupa un lugar central en la detección de actividades de comando y control. Dentro del marco MITRE ATT&CK, la táctica de Command and Control (TA0011) contempla múltiples técnicas mediante las cuales un sistema comprometido establece comunicaciones periódicas con infraestructura adversaria para recibir instrucciones o exfiltrar información (MITRE ATT&CK, s.f.). Estas comunicaciones suelen caracterizarse por su regularidad

temporal y por un volumen de datos relativamente bajo, diseñado para evitar umbrales tradicionales de alerta.

Elastic Security Labs describe el beaconing como un patrón de comunicación repetitivo, donde un host interno establece conexiones salientes con intervalos consistentes o cuasi consistentes hacia un mismo destino (Elastic Security Labs, s.f.). La identificación de este comportamiento no depende del contenido transmitido, sino de la estructura temporal del tráfico. En entornos donde el cifrado TLS es predominante, el análisis temporal se convierte en uno de los principales mecanismos de detección.

Zeek proporciona herramientas particularmente adecuadas para este tipo de análisis. El registro conn.log incluye campos como duración, bytes transferidos, dirección de origen y destino, y estado de la conexión (The Zeek Project, 2023). Estos datos permiten construir series temporales que describen la frecuencia y regularidad de las comunicaciones. Cuando un host presenta conexiones salientes hacia un mismo destino

con intervalos uniformes y tamaños de transferencia similares, se configura un patrón compatible con beaconing.

No obstante, los atacantes pueden introducir variabilidad deliberada en los intervalos —jitter— para evadir detecciones basadas en periodicidad exacta. Elastic señala que esta variación no elimina la regularidad estructural, sino que la difumina dentro de un rango acotado (Elastic Security Labs, s.f.). Por ello, el análisis debe incorporar métricas de dispersión y no limitarse a intervalos perfectamente constantes.

La identificación de long-connections constituye un fenómeno complementario. Mientras que el beaconing clásico se basa en conexiones breves y repetitivas, algunas herramientas de control remoto establecen conexiones persistentes de larga duración para mantener un canal abierto con el servidor adversario. En este caso, el análisis se centra en detectar sesiones inusualmente extensas o flujos con transferencia constante durante períodos prolongados. Zeek permite observar duración total y volumen de bytes, lo que facilita identificar conexiones que se apartan del comportamiento normal de la red (The Zeek Project, 2023).

Desde la perspectiva metodológica del threat hunting, Raggi (2021) subraya que el proceso comienza con una hipótesis concreta. En este contexto, una hipótesis podría formularse como: “Existe un host que mantiene comunicaciones persistentes o periódicas hacia infraestructura externa no habitual”. La validación de esta hipótesis requiere correlacionar registros de conn.log con dns.log y, eventualmente, con http.log o tls.log, dependiendo del protocolo utilizado.

Aragonés Lozano (2024) enfatiza que el análisis eficaz de grandes volúmenes de datos exige modelar patrones normales para poder detectar desviaciones. En el caso del beaconing, esto implica conocer la frecuencia típica de conexiones hacia servicios legítimos —por ejemplo, actualizaciones de software o sincronización en la nube— y distinguirlas de patrones anómalos. Sin una línea base estadística, una conexión periódica podría interpretarse erróneamente como maliciosa cuando, en realidad, corresponde a un servicio legítimo.

Wireshark desempeña un papel complementario en la fase de validación. Una vez identificado un patrón sospechoso a nivel de log estructurado, la inspección detallada de paquetes

permite confirmar características como la consistencia de los encabezados, la presencia de datos cifrados o la estructura del handshake TLS (Sharpe, Warnicke & Lamping, 2023). Sin embargo, el descubrimiento inicial del patrón suele depender del análisis agregado y temporal proporcionado por Zeek.

Un escenario ilustrativo puede describirse del siguiente modo: un equipo interno establece conexiones HTTPS cada cinco minutos hacia un dominio recientemente registrado. Las sesiones presentan duraciones breves y volúmenes de datos similares. El análisis temporal revela baja variabilidad en los intervalos. La correlación con dns.log indica que el dominio es poco frecuente en el entorno. La combinación de periodicidad, rareza y coherencia estructural fortalece la hipótesis de un canal de comando y control.

En síntesis, la detección de beaconing y long-connections se fundamenta en el análisis temporal y estructural del tráfico, más que en la inspección de contenido. En entornos dominados por cifrado, esta aproximación conductual se convierte en un pilar del enfoque NDR, permitiendo identificar comunicaciones persistentes incluso cuando los datos transmitidos permanecen inaccesibles.

Pivot con endpoints

El análisis de red, por sí solo, proporciona una visión estructural del tráfico, pero no siempre permite comprender plenamente el contexto operativo del host involucrado. En el enfoque contemporáneo de Network Detection and Response, el pivot desde la telemetría de red hacia los endpoints constituye un paso decisivo para confirmar o refutar hipótesis formuladas durante el proceso de threat hunting. Este desplazamiento analítico responde a la necesidad de correlacionar comportamiento de red con actividad interna del sistema.

De acuerdo con el marco MITRE ATT&CK, las tácticas de comando y control forman parte de una cadena más amplia de acciones que incluyen ejecución, persistencia y movimiento lateral (MITRE ATT&CK, s.f.). Detectar una comunicación sospechosa en red no basta; es necesario determinar qué proceso la originó, qué usuario estaba involucrado y qué otras acciones se ejecutaron en el host comprometido. El pivot hacia el endpoint permite integrar estas dimensiones.

El proceso de threat hunting, tal como lo describe Raggi (2021), parte de una hipótesis y progresa mediante la correlación de múltiples fuentes de datos. Una vez identificada una anomalía en conn.log, dns.log o tls.log, el siguiente paso lógico consiste en investigar el host de origen. Este pivot no implica abandonar el análisis de red, sino ampliarlo. La pregunta deja de ser únicamente "¿qué tráfico se generó?" y pasa a ser "¿qué actividad interna generó ese tráfico?".

Desde la perspectiva arquitectónica, Aragonés Lozano (2024) destaca que los cazadores de amenazas deben analizar grandes volúmenes de datos heterogéneos en intervalos reducidos de tiempo. Esta heterogeneidad incluye registros de red, eventos de sistema, autenticaciones y ejecución de procesos. La integración de estas fuentes es fundamental para alcanzar una consciencia situacional adecuada, ya que el análisis aislado de una sola capa puede conducir a conclusiones parciales.

En términos prácticos, el pivot puede desarrollarse del siguiente modo: un análisis de beaconing detecta conexiones periódicas desde un host interno hacia un dominio raro. El siguiente paso consiste en examinar en el endpoint qué

proceso estableció la conexión. Si se identifica un binario desconocido o ejecutado desde una ubicación inusual, la hipótesis de compromiso se fortalece. Este procedimiento refleja la lógica del hunting iterativo descrito por Raggi (2021), donde cada hallazgo conduce a nuevas preguntas y nuevas verificaciones.

Zeek facilita el inicio del pivot al proporcionar identificadores consistentes de conexión, como dirección IP, puerto y timestamp (The Zeek Project, 2023). Estos elementos permiten correlacionar eventos de red con registros de host. Si la organización dispone de un entorno integrado —como el descrito en la documentación de Security Onion—, la correlación puede automatizarse parcialmente, reduciendo el tiempo de investigación (Security Onion Solutions, 2024).

Wireshark, en este contexto, mantiene su función como herramienta de validación puntual. Si el pivot revela actividad sospechosa en el endpoint, la inspección detallada de los paquetes asociados puede aportar evidencia adicional sobre la naturaleza del tráfico (Sharpe, Warnicke & Lamping, 2023). Sin embargo, el objetivo principal del pivot no es reconstruir cada byte transmitido, sino comprender el encadenamiento causal entre evento de red y actividad interna.

Un aspecto crítico es evitar falsos positivos derivados de comportamiento legítimo. Servicios corporativos, actualizaciones automáticas o sincronizaciones en la nube pueden generar patrones periódicos similares a los de beaconing. Solo el análisis del endpoint permite distinguir entre un proceso legítimo firmado digitalmente y un ejecutable malicioso desplegado recientemente. De este modo, el pivot reduce la ambigüedad inherente al análisis exclusivamente de red.

En términos estratégicos, el pivot con endpoints refleja la transición desde una detección basada en anomalías hacia una investigación contextualizada. La red proporciona indicios; el endpoint ofrece evidencia directa sobre ejecución y persistencia. Esta integración reduce el tiempo de permanencia del adversario y fortalece la capacidad de respuesta temprana.

En síntesis, el pivot con endpoints constituye el cierre natural del ciclo de casos de uso analizados en este bloque. El análisis DNS sospechoso, la identificación de anomalías HTTP/TLS y la detección de beaconing adquieren pleno significado cuando se integran con la telemetría de host. Solo mediante esta correlación multicapa es posible transformar una anomalía

estadística en una conclusión fundamentada sobre compromiso real.

CONTINUAR

Preguntas de repaso

El uso de TLS impide completamente la detección de actividad de comando y control en entornos NDR, ya que el contenido del tráfico se encuentra cifrado.

- ◯ Verdadero
- ◯ Falso

SUBMIT

La identificación de beaconing se basa exclusivamente en intervalos perfectamente constantes entre conexiones repetidas.

- ◯ Verdadero
- ◯ Falso

SUBMIT

El pivot desde la telemetría de red hacia los endpoints es un paso necesario para confirmar si una anomalía detectada en DNS o TLS corresponde efectivamente a un compromiso del host.

- ◯ Verdadero
- ◯ Falso

SUBMIT

Conclusión

El análisis de protocolos mediante Wireshark y Zeek constituye un eje metodológico central en arquitecturas modernas de Network Detection and Response. Mientras Wireshark aporta inspección profunda de paquetes a nivel forense, Zeek permite modelar el comportamiento de la red a través de logs estructurados y análisis temporal. La detección de DNS sospechoso, anomalías HTTP/TLS, beaconing y la posterior correlación con endpoints demuestra que la eficacia no

depende únicamente del contenido visible, sino del análisis conductual y contextual. Integrar modelado estadístico, fingerprinting y pivot multicapa fortalece la capacidad de identificar actividad adversaria incluso en entornos ampliamente cifrados.

CONTINUAR

Referencias

Aragonés Lozano, M. (2024). *Threat hunting basado en técnicas de inteligencia artificial.* Universitat Politècnica de València.

Elastic. (s.f.). *Network beaconing identification*. Elastic Integrations.

Elastic Security Labs. (s.f.). *Identifying beaconing malware*. Elastic.

MITRE ATT&CK. (s.f.). *Command and control (TA0011) – Enterprise matrix*. MITRE Corporation.

Mitkov, I. (2021). *Análisis de paquetes con Wireshark: Estudio de vulnerabilidades.*

Raggi, C. (2021). *Threat hunting: La práctica de detectar amenazas ocultas en nuestra red.*

Salesforce. (s.f.). *JA3: TLS client fingerprinting*.

Security Onion Solutions. (2024). *Zeek – Security Onion documentation* (Version 2.4).

Sharpe, R., Warnicke, E., & Lamping, U. (2023). *Wireshark user's guide*.

The Zeek Project. (2023). *The Book of Zeek* (Version 8.1.1).

The Zeek Project. (2023). *Base protocols: conn package*.

The Zeek Project. (2023). *Base protocols: dns package*.

The Zeek Project. (2023). *Base protocols: http package*.

The Zeek Project. (2023). *Base protocols: ssl package*.

CONTINUAR