SOCIA · CIBERSEGURIDAD

Fuerza bruta SSH contra un servidor interno

Del triaje de la alerta al cierre del caso en un SOC

Herramientas: TheHive · Cortex · Graylog/Wazuh · Malcolm · Arkime · OPNsense

Roles: analista N1 y analista N2

Grabación: 9 de agosto de 2026

Tiempo activo: 18:42

Contexto del caso:

Wazuh ha detectado una sucesión de autenticaciones SSH fallidas contra un servidor de la red interna y ha enviado la alerta a TheHive a través de Graylog. A partir de ahí, el equipo del SOC tiene que decidir tres cosas: si el ataque ha tenido éxito, hasta dónde ha llegado el atacante y qué medida de contención corresponde aplicar.

El recorrido reproduce el reparto de trabajo habitual entre dos niveles. El analista de nivel 1 (N1) recibe la alerta, abre el caso, cualifica los observables y deja por escrito lo que ha averiguado. El analista de nivel 2 (N2) recoge ese testigo, confirma con los registros y el tráfico capturado si hubo una autenticación correcta, mide el alcance del ataque, aplica la contención en el cortafuegos y cierra el expediente.

Conviene tener presente desde el principio que los pasos concretos de este tipo de casos dependen de las premisas y las instrucciones que cada organización haya definido. El reparto entre N1 y N2 que se usa aquí es uno de los posibles: en otra organización todo el recorrido podría corresponder al N1, o el corte podría estar en otro punto. Lo que sí se mantiene de un sitio a otro es el método: cualificar, comprobar, medir el alcance, contener y documentar.

Objetivos de aprendizaje

- Convertir una alerta de Wazuh en un caso de TheHive con los campos de clasificación bien puestos.
- Cualificar observables y lanzar analizadores de Cortex sobre una IP sospechosa.
- Redactar un traspaso N1→N2 que permita continuar la investigación sin repetir trabajo.
- Comprobar en Graylog si una fuerza bruta SSH terminó en un inicio de sesión aceptado.
- Medir el alcance de la actividad de una IP con Malcolm y Arkime.
- Aplicar y documentar una contención por alias de cortafuegos en OPNsense.
- Cerrar el caso con una clasificación y un resumen que sirvan de referencia después.

Índice de contenidos

FASE 1

ANALISTA N1

Triaje de la alerta y apertura del caso

TheHive es el portal donde el SOC da seguimiento a las alertas y a los casos, y va a ser el punto central de todo el trabajo: aunque la investigación pase por Graylog, Malcolm o el cortafuegos, todo lo que se averigüe vuelve aquí. Por eso el primer movimiento del analista no es buscar paquetes, sino abrir el portal y ver qué hay en la cola.

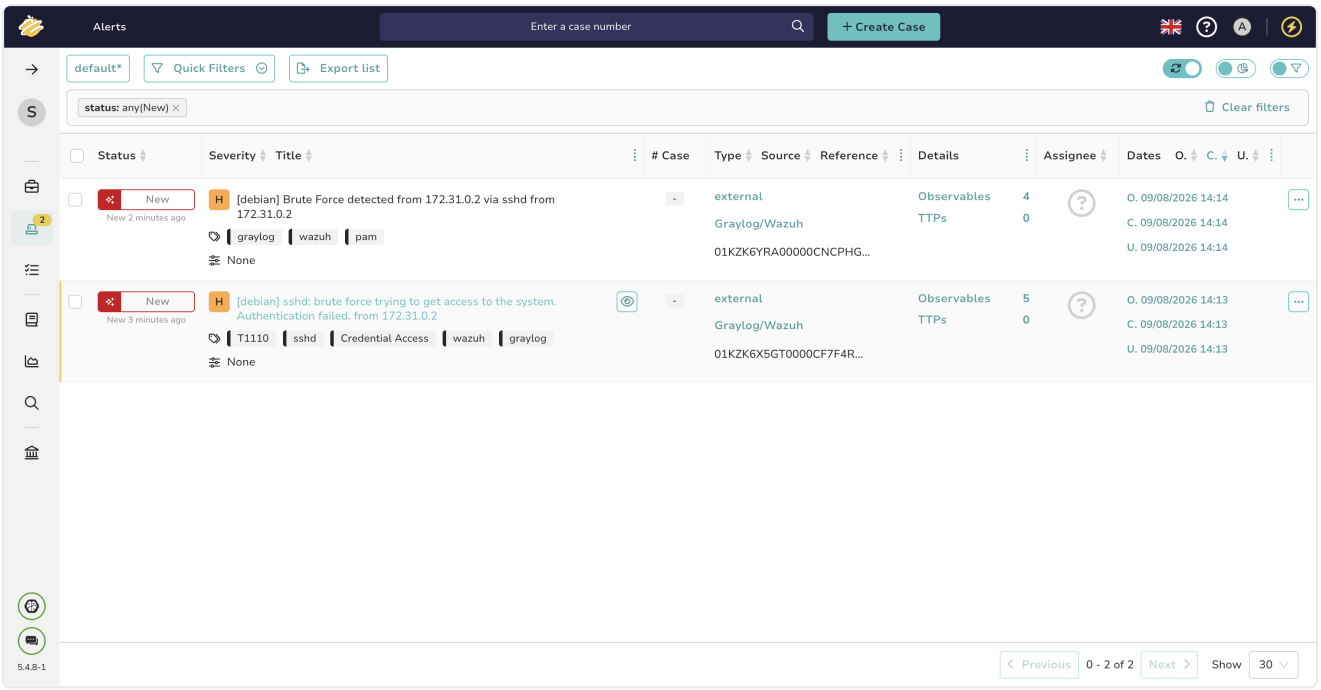
Esta primera fase responde a una pregunta muy concreta: *¿qué me está contando la alerta y merece convertirse en un caso?* Nos interesa distinguir bien dos objetos que TheHive trata de forma distinta. Una **alerta** es una notificación que llega de una fuente externa —aquí, Wazuh a través de Graylog— y que todavía no tiene a nadie asignado ni un procedimiento en marcha. Un **caso** es el expediente de trabajo: tiene responsable, estado, observables, comentarios y una historia. Trabajar directamente sobre alertas no deja rastro; por eso el primer paso real de la investigación es promover la alerta a caso.

Paso 1. Revisar la cola de alertas

Después de iniciar sesión con las credenciales del analista de nivel 1, vamos primero a la sección **Cases** y comprobamos que no hay ningún caso abierto: nadie ha empezado a trabajar todavía. Pasamos entonces a **Alerts**, que es donde aterriza lo que envían las fuentes de detección.

Aquí encontramos dos alertas que, leyendo sus títulos, se refieren al mismo hecho. Una llega como `[debian] sshd: brute force trying to get access to the system. Authentication failed. from 172.31.0.2` y trae cinco observables; la otra, `[debian] Brute Force detected from 172.31.0.2 via sshd`, trae cuatro y llega por otra vía de detección. Ambas comparten origen `Graylog/Wazuh` y severidad HIGH.

Cuando dos alertas describen el mismo incidente conviene empezar por la que aporta más información y dejar la otra para el final, en lugar de abrir dos expedientes paralelos. Nosotros vamos a trabajar sobre la primera, que es la que trae más observables, y al cerrar el caso volveremos a por la segunda.



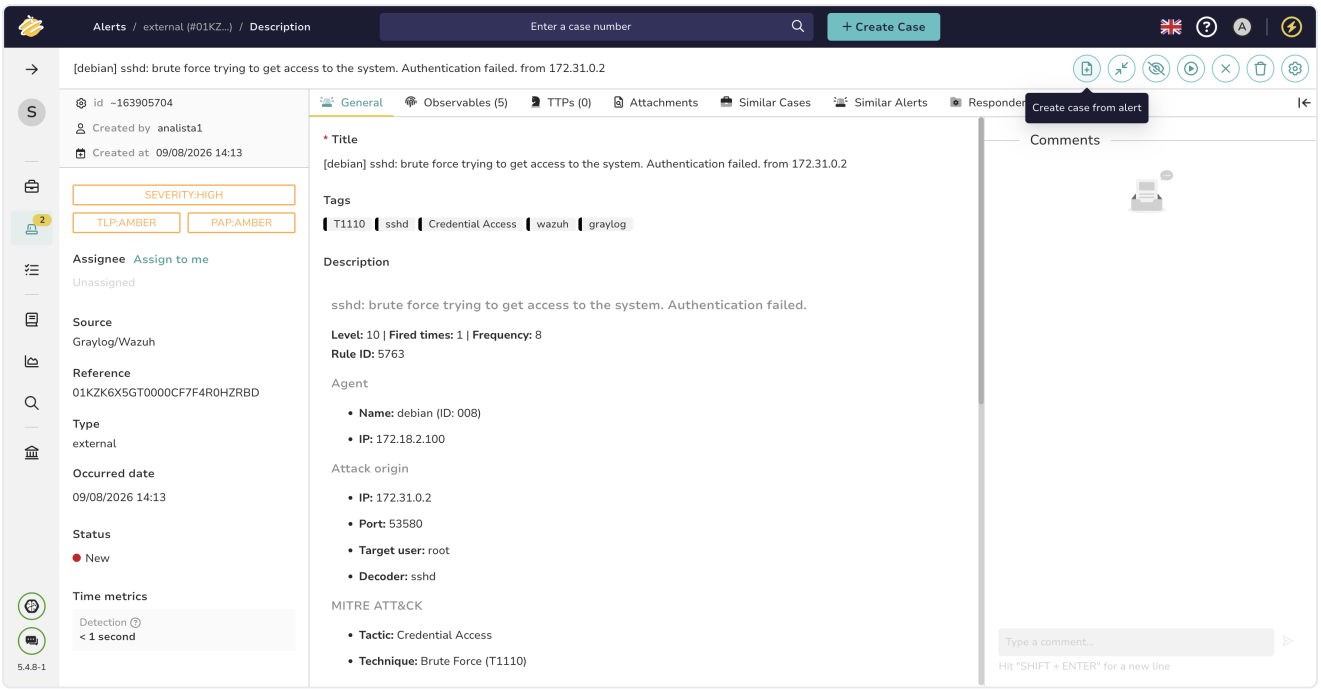
Cola de alertas de TheHive con las dos alertas del mismo incidente, ambas con origen Graylog/Wazuh y severidad HIGH.

Paso 2. Leer la alerta y quedarse con los datos del ataque

Al entrar en la alerta tenemos que hacer lo que hace cualquier analista antes de tocar nada: analizar qué datos hay ahí. La descripción que genera Wazuh es bastante completa y ya nos da casi todo lo que necesitamos para redactar el caso.

De la ficha nos quedamos con la regla que ha disparado la detección (`Rule ID: 5763`, nivel 10, con una frecuencia de 8 intentos), con el agente afectado (`debian`, IP `172.18.2.100`) y con el origen del ataque (`172.31.0.2`, puerto `53580`), además del usuario contra el que se intentaba entrar, `root`. La alerta también viene clasificada en MITRE ATT&CK como *Credential Access / Brute Force (T1110)*.

Ese es el hallazgo de partida: un intento de fuerza bruta por SSH desde `172.31.0.2`. Lo que la alerta **no** nos dice es si alguno de esos intentos acabó funcionando, y esa será justamente la pregunta que arrastraremos durante el resto del caso.



Detalle de la alerta en TheHive: regla 5763, agente debian (172.18.2.100), origen del ataque 172.31.0.2 puerto 53580 y usuario objetivo root.

Paso 3. Crear el caso a partir de la alerta

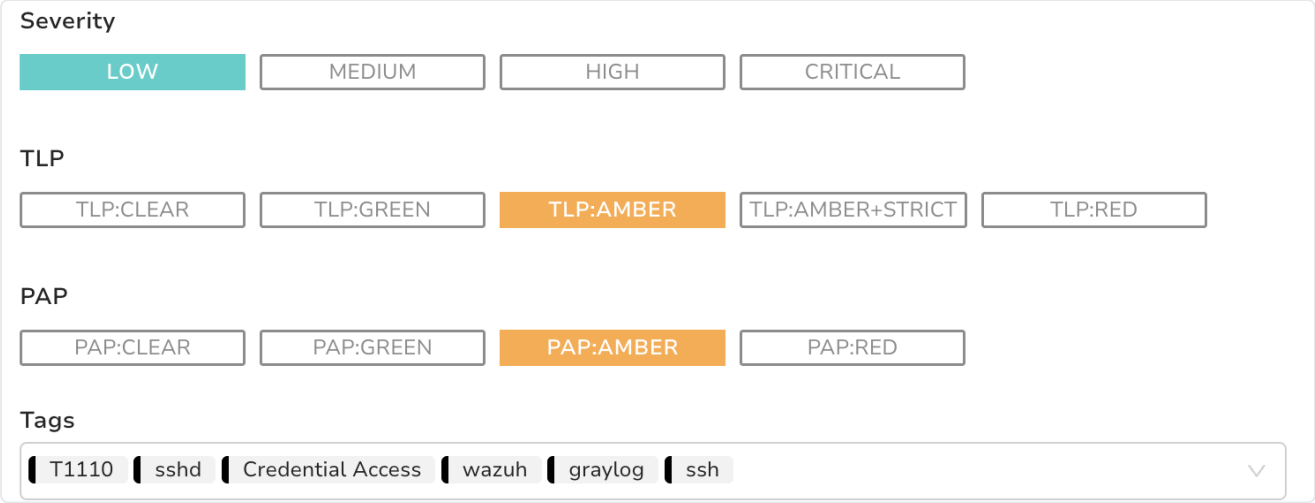
Para empezar a trabajar hay que crear un caso a partir de la alerta. El botón *Create case from alert* está en la barra de acciones de la propia alerta, y al pulsarlo TheHive nos pregunta cómo queremos crearlo. Muchas organizaciones definen **plantillas** para tratar todos los casos parecidos de la misma manera; como aquí no hay ninguna definida, partimos de un caso vacío (*Empty case*) y lo rellenamos desde cero.

El **título** conviene que se lea solo. Seguimos un patrón de tipo de ataque + máquina afectada con su IP + IP de origen, y lo dejamos como `Brute force SSH to from vulnmachine (172.18.2.100) from 172.31.0.2`. El nombre de la máquina lo tomamos de la propia descripción de la alerta; si no lo supiéramos, siempre se puede consultar después en los observables.

En **severidad** bajamos la alerta de HIGH a **LOW**. No es un capricho: mientras no sepamos si el atacante ha llegado a entrar, un intento fallido de fuerza bruta no tiene por qué ser grave, y mantener una severidad alta desde el principio distorsiona la cola de trabajo del SOC. Si más adelante se confirmara el acceso, la severidad se sube.

Los campos **TLP** y **PAP** son los que regulan qué se puede hacer con la información del caso. Ponemos los dos en **AMBER**: el TLP indica que la información se puede compartir dentro de la organización, y el PAP, que se puede compartir con terceros para que la analicen.

En **tags** la alerta ya arrastra `T1110`, `sshd`, `Credential Access`, `wazuh` y `graylog`; le añadimos `ssh`. Las etiquetas son lo que permitirá localizar y agrupar estos casos más adelante, así que merece la pena añadir las que la organización tenga acordadas. La descripción la dejamos tal y como viene generada, porque ya es suficientemente completa, y pulsamos **Confirm**.



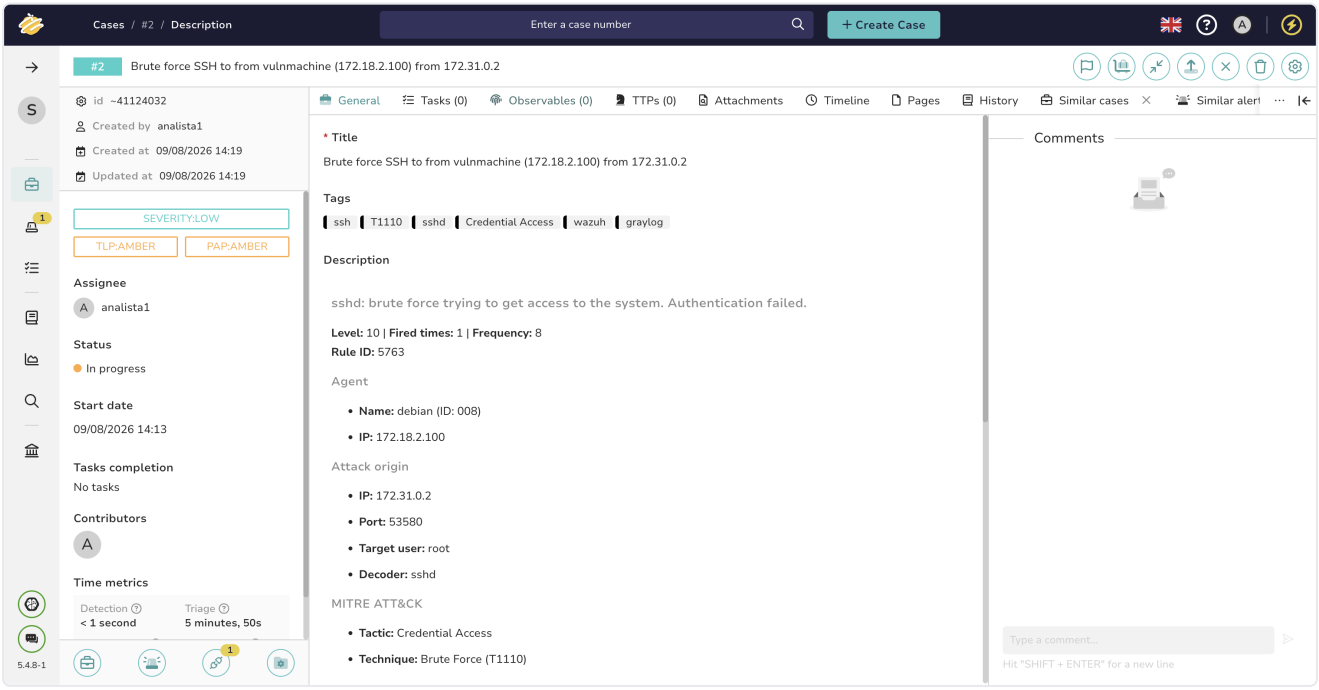
Formulario de creación del caso listo para confirmar: severidad LOW, TLP:AMBER, PAP:AMBER y la etiqueta ssh añadida a las heredadas de la alerta.

Paso 4. Poner el caso en progreso

Con el caso ya creado, el primer gesto es cambiar su estado de *New* a **In progress** y guardar. Es un detalle pequeño con una función real: indica al resto del equipo que alguien está trabajando el caso y evita que dos analistas se pisen sobre la misma alerta.

La ficha del caso queda entonces con su número (`#2`), la severidad LOW que hemos fijado, TLP y PAP en AMBER, el analista como responsable y las métricas de tiempo que TheHive va calculando por su cuenta.

> **Resultado:** El caso #2 queda abierto, asignado y en estado *In progress*, con los cinco observables importados de la alerta pendientes de cualificar.



Ficha del caso #2 recién creado, en estado In progress, con severidad LOW y TLP/PAP AMBER.

FASE 2

ANALISTA N1

Cualificación de los observables

Los **observables** son los datos concretos que el caso arrastra y sobre los que se puede investigar: direcciones IP, nombres de máquina, puertos, usuarios. Al crear el caso desde la alerta, TheHive ha importado cinco: el puerto de origen `53580`, el usuario objetivo `root`, el agente `debian`, la IP de la víctima `172.18.2.100` y la IP del atacante `172.31.0.2`.

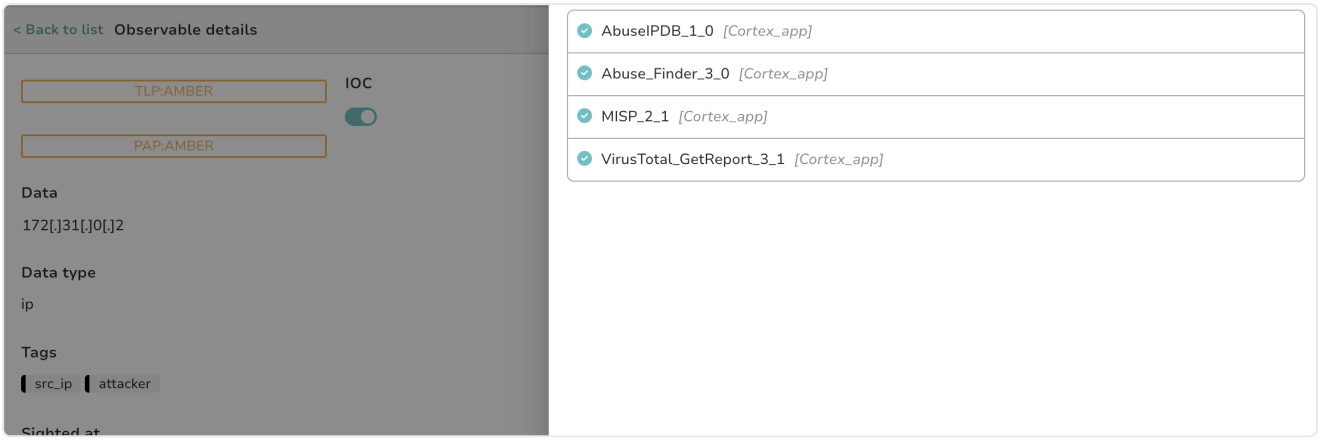
Tenerlos ahí no basta. La fase consiste en decir de cada uno *qué papel juega* y *qué sabemos de él*, porque de eso dependen tanto las búsquedas posteriores como el valor que el caso tendrá para futuras investigaciones. La herramienta que nos ayuda en la segunda parte es **Cortex**: el motor de analizadores que TheHive consulta para enriquecer un observable con fuentes externas de reputación.

Paso 5. Cualificar la IP atacante y lanzar los analizadores de Cortex

Entramos en el observable `172.31.0.2`, la IP del atacante. Hay dos marcas que decidir aquí. La primera es **IOC**: si no viniera ya activada, tendríamos que activarla, porque se trata de un indicador de compromiso. La segunda es **Sighted**, que sirve para dejar constancia de que esa IP se ha visto realmente dentro de nuestra infraestructura y no es un indicador que nos hayamos descargado de un boletín externo. Es una distinción que importa: lo observado en casa y lo importado de fuera no valen lo mismo.

Añadimos también la etiqueta `attacker` si no estuviera —aquí acompaña a `src_ip`— y, muy importante, pulsamos **Save** después de cada cambio; TheHive no guarda solo.

A continuación, desde el menú de tres puntos del observable, elegimos **Run analyzers**. Se abre la lista de analizadores de IP que la organización tenga activados en Cortex: `AbuseIPDB`, `Abuse_Finder`, `MISP` y `VirusTotal_GetReport`. Con *Select all* los seleccionamos todos y lanzamos la ejecución.



Observable 172.31.0.2 con IOC activado y etiquetas src_ip y attacker, junto al diálogo de selección de los cuatro analizadores de Cortex.

Paso 6. Leer los informes que devuelve Cortex

Los informes tardan unos segundos en llegar. Cuando volvemos a la lista de observables, la IP del atacante aparece ya con las etiquetas de resultado de cada analizador: `AbuseIPDB:UsageType="Reserved"`, `AbuseIPDB:Records="0"`, `VT:GetReport="0/91"`, `VT:GetReport="124 resolution(s)"` y `MISP:Search="1 event(s)"`.

La lectura es la que se ve: los servicios de reputación no tienen nada relevante sobre esta dirección. Y tiene su explicación, porque `172.31.0.2` es una dirección de rango reservado —AbuseIPDB lo dice literalmente— y los repositorios públicos apenas registran actividad de direcciones que no son enrutables en Internet. Frente a una IP pública real, estos mismos analizadores nos habrían dado muchísima más información.

Que no haya reputación negativa no significa que la IP sea inofensiva: significa que esta vía no nos aporta nada y que la respuesta tendrá que salir de nuestros propios registros y de nuestro propio tráfico.

> **Resultado:** Ningún analizador aporta reputación negativa sobre 172.31.0.2. La calificación del atacante tendrá que apoyarse en la telemetría interna.

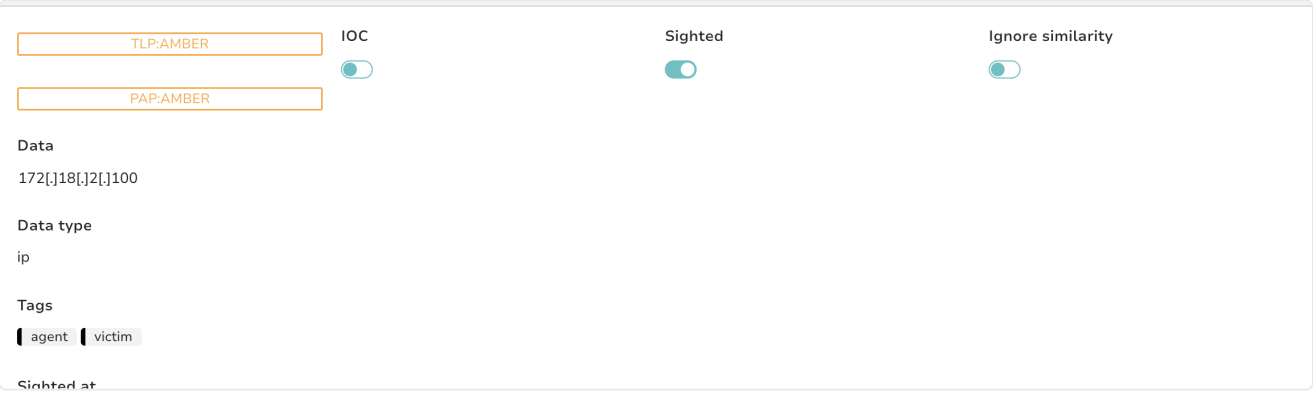
Etiquetas de resultado de Cortex sobre el observable 172.31.0.2: AbuseIPDB "Reserved" y "0 records", VirusTotal 0/91 y MISP 1 evento.

Paso 7. Etiquetar la máquina víctima

Pasamos ahora al otro extremo de la conexión, el observable `172.18.2.100`. Le añadimos la etiqueta `victim` junto a la de `agent` que ya traía, para que quede claro de un vistazo quién ataca y quién recibe.

Aquí la decisión sobre IOC es la contraria que en el atacante: no marcamos como indicador de compromiso una IP propia y conocida. Se puede defender lo contrario —hay organizaciones que sí la registran, para poder rastrear después todos los casos que han tocado ese activo—, así que este es uno de esos puntos en los que conviene seguir el criterio de la casa. Lo que sí dejamos activo es **Sighted**, porque el activo efectivamente ha aparecido en el incidente.

Con los dos extremos cualificados, el trabajo de identificación está hecho: sabemos quién ataca, a quién y con qué resultado aparente.



Observable 172.18.2.100 con las etiquetas agent y victim, IOC desactivado y Sighted activo.

FASE 3

ANALISTA N1 → N2

Traspaso del caso al nivel 2

Muchas organizaciones sitúan aquí el límite del nivel 1: el N1 cualifica y documenta, pero no analiza tráfico ni toca el cortafuegos. Al llegar a ese punto, el caso cambia de manos.

El traspaso se hace por escrito, en la pestaña de **comentarios** del caso. No es burocracia: un traspaso bien redactado es lo que evita que el N2 repita las comprobaciones que ya están hechas y lo que permite reconstruir después por qué se tomó cada decisión. Como todo lo demás, dónde está exactamente la frontera entre N1 y N2 depende de cada organización.

Paso 8. Escribir el comentario de traspaso N1→N2

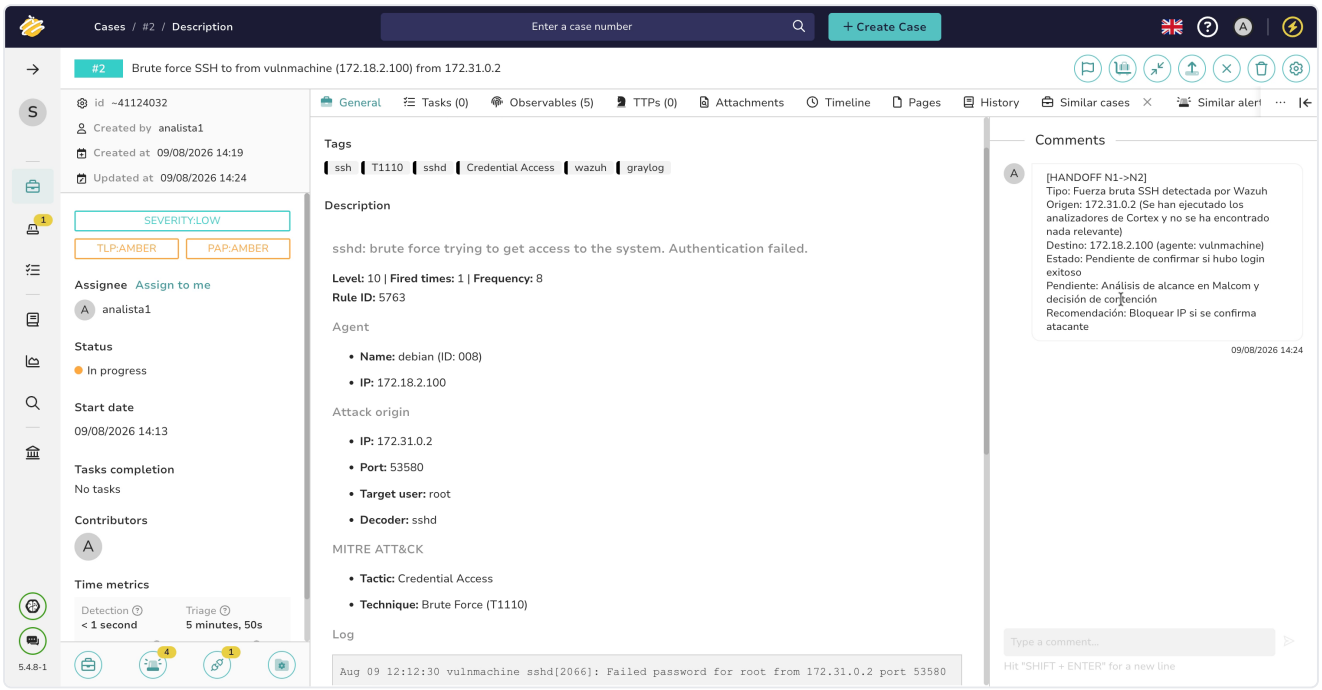
En **Comments** dejamos una nota estructurada. La estructura importa más que el estilo: quien la lea tiene que saber en veinte segundos qué se sabe, qué falta y qué se propone.

```
[HANDOFF N1->N2]
Tipo: Fuerza bruta SSH detectada por Wazuh
Origen: 172.31.0.2 (Se han ejecutado los analizadores de Cortex
        y no se ha encontrado nada relevante)
Destino: 172.18.2.100 (agente: vulnmachine)
Estado: Pendiente de confirmar si hubo login exitoso
Pendiente: Análisis de alcance en Malcolm y decisión de contención
Recomendación: Bloquear IP si se confirma atacante
```

Fíjate en lo que hace cada línea. *Tipo*, *Origen* y *Destino* resumen el hecho y ya incorporan el resultado de Cortex, para que nadie vuelva a lanzarlos. *Estado* nombra la pregunta que sigue abierta —si hubo un inicio de sesión correcto—, que es exactamente la pista que el N2 necesita para saber por dónde empezar. *Pendiente* reparte el trabajo que queda, y *Recomendación* propone una acción condicionada: bloquear la IP *si* se confirma que es un atacante, no antes.

Con la nota publicada, el N1 cierra su turno. El relevo se hace cerrando sesión y entrando con las credenciales del analista de nivel 2, que abre el caso desde la lista y encuentra el traspaso esperándole en la columna de comentarios.

Nota: El cambio de usuario no es un formalismo: además de dejar trazabilidad de quién hizo cada cosa, en TheHive los permisos y las acciones disponibles pueden variar según el perfil.



El caso #2 visto por el analista de nivel 2, con el comentario de traspaso ya publicado en el panel de comentarios.

FASE 4

ANALISTA N2

¿Llegó a entrar el atacante?

La pregunta que el N1 ha dejado abierta solo se puede responder en los registros. **Graylog** es el gestor de registros donde aterrizan los eventos que recogen los agentes de Wazuh, y su buscador permite filtrar por los campos de esos eventos en una ventana de tiempo concreta.

El razonamiento que vamos a seguir tiene dos movimientos. Primero confirmamos que los intentos fallidos están ahí y de dónde vienen; después buscamos explícitamente lo contrario —una autenticación aceptada — porque en seguridad la ausencia de una prueba hay que ir a buscarla, no darla por supuesta.

Paso 9. Acotar la ventana de tiempo y buscar los eventos de fuerza bruta

Entramos en Graylog con las credenciales del analista y vamos al apartado **Search**. Lo primero es fijar la ventana temporal: la desplegamos y elegimos **las últimas 2 horas**, suficiente para cubrir el incidente sin arrastrar ruido de días anteriores.

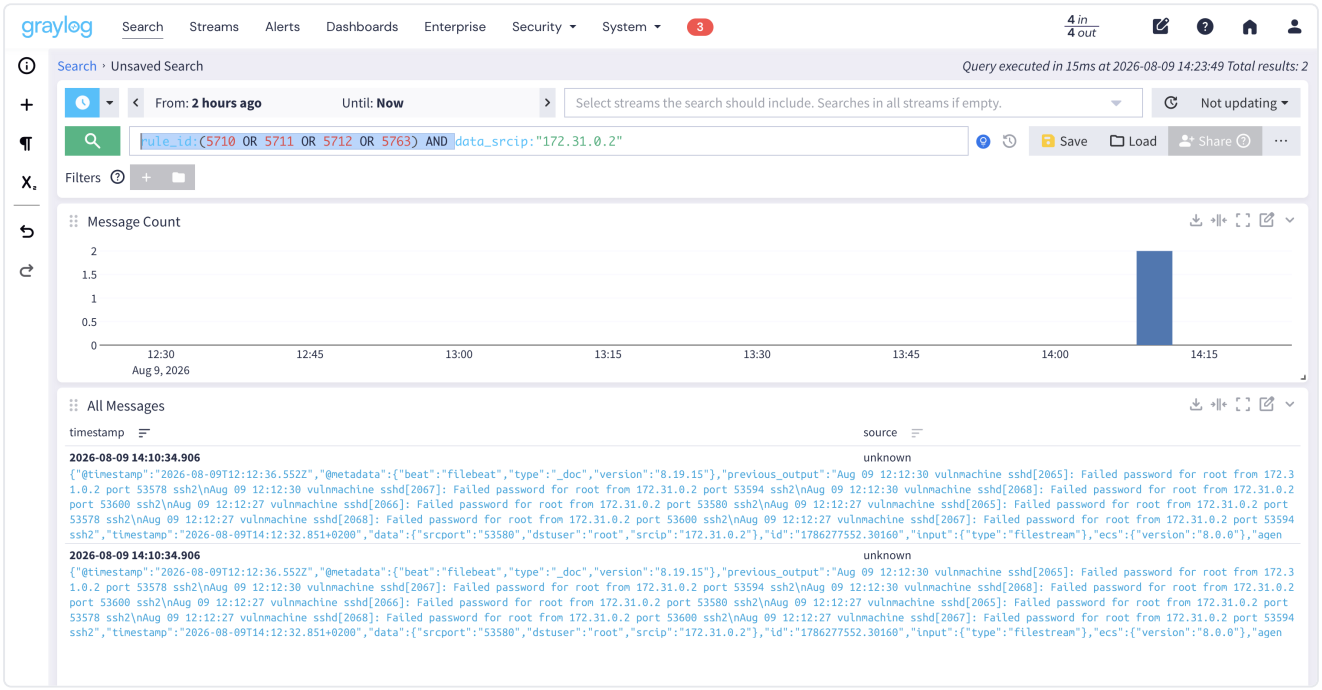
La consulta que escribimos combina dos condiciones:

```
rule_id:(5710 OR 5711 OR 5712 OR 5763) AND data_srcip:"172.31.0.2"
```

La primera parte, `rule_id:(...)`, selecciona la familia de reglas de `sshd` que Wazuh asocia a autenticaciones fallidas y a fuerza bruta —entre ellas la 5763, que es la que disparó nuestra alerta—. La segunda, `data_srcip`, restringe el resultado a los eventos cuya IP de origen es la del atacante. Unidas por `AND`, la consulta pregunta: *de todo lo que ha registrado Wazuh en las dos últimas horas, ¿qué intentos de fuerza bruta SSH vienen de esta dirección?*

El resultado son 2 mensajes, y al abrirlos vemos repetido el texto que esperábamos: `Failed password for root from 172.31.0.2 port 53578 ssh2`, `port 53594`, `port 53600`… Cada intento sale de un puerto de origen distinto, que es el patrón típico de una serie de conexiones automatizadas.

Con esto queda confirmado que el ataque existió y que todo lo registrado son contraseñas fallidas. Pero *fallidas las que vemos* no es lo mismo que *ninguna correcta*, y ahí es donde entra la segunda consulta.



Graylog con la consulta por rule_id y data_srcip en las últimas 2 horas: 2 resultados con mensajes «Failed password for root from 172.31.0.2».

Paso 10. Buscar explícitamente una autenticación aceptada

Para responder a la pregunta del traspaso le damos la vuelta a la búsqueda: en lugar de contar fallos, buscamos éxitos. Partimos de la misma IP de origen y le añadimos dos formas de detectar un acceso correcto:

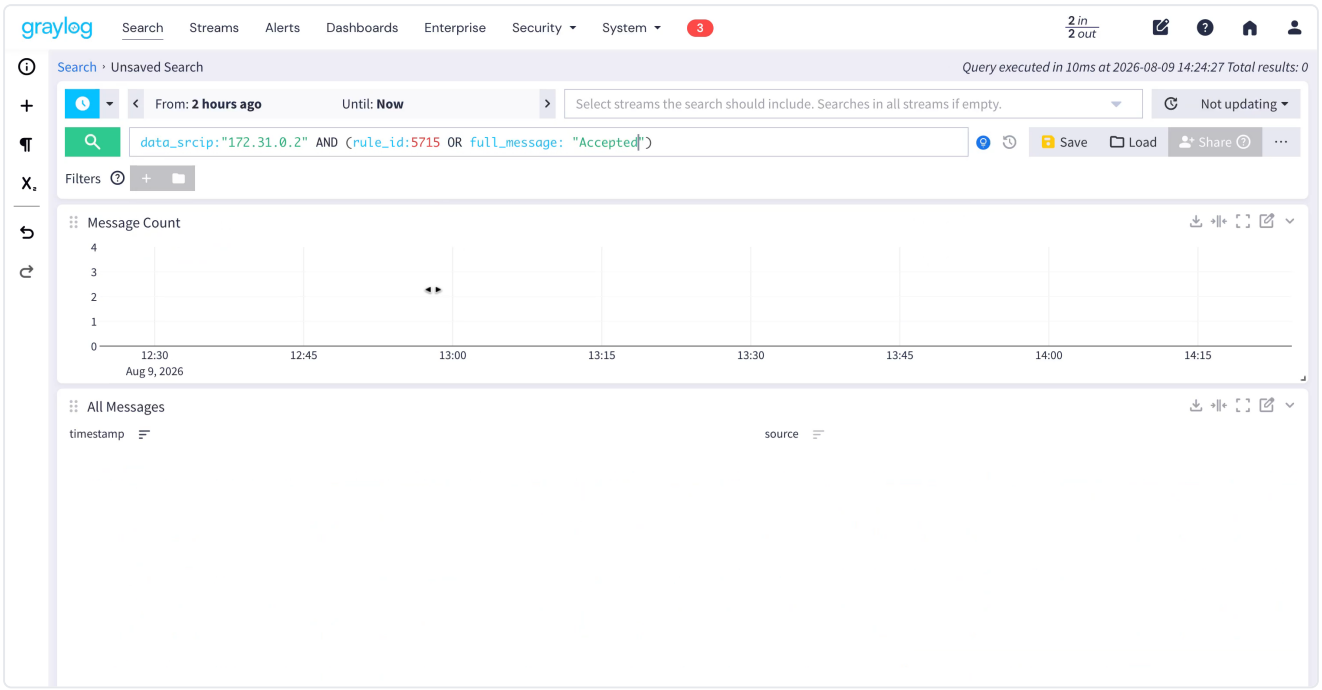
```
data_srcip:"172.31.0.2" AND (rule_id:5715 OR full_message: "Accepted")
```

La condición `rule_id:5715` apunta a la regla con la que Wazuh marca un inicio de sesión SSH aceptado. La segunda, `full_message: "Accepted"`, busca directamente la palabra *Accepted* en el texto del mensaje, que es la que escribe `sshd` cuando una autenticación prospera. Se buscan las dos con un `OR` a propósito: si por lo que fuera la regla no hubiera clasificado bien el evento, el texto del propio registro seguiría delatándolo.

La búsqueda devuelve **Total results: 0**, con el histograma vacío y sin ningún mensaje en la tabla. En la ventana revisada no hay ni un solo acceso correcto desde esa dirección.

> **Resultado:** Cero eventos. El atacante no consiguió autenticarse por SSH: el intento fue de fuerza bruta, pero no hubo compromiso de credenciales.

> **Nota:** Un resultado vacío solo vale dentro de la ventana y del filtro consultados. Si el ataque hubiera empezado antes del rango elegido, o si el agente hubiera dejado de enviar registros, el cero no probaría nada. Por eso la ventana se fija de forma consciente antes de lanzar la consulta.



La búsqueda de autenticaciones aceptadas desde 172.31.0.2 devuelve «Total results: 0» y un histograma sin ningún mensaje.

FASE 5

ANALISTA N2

Alcance del ataque en la red

Los registros nos dicen lo que la máquina atacada apuntó en su diario. El tráfico capturado nos dice lo que realmente pasó por la red, y eso permite responder a una pregunta distinta: *¿hasta dónde llegó este atacante?*

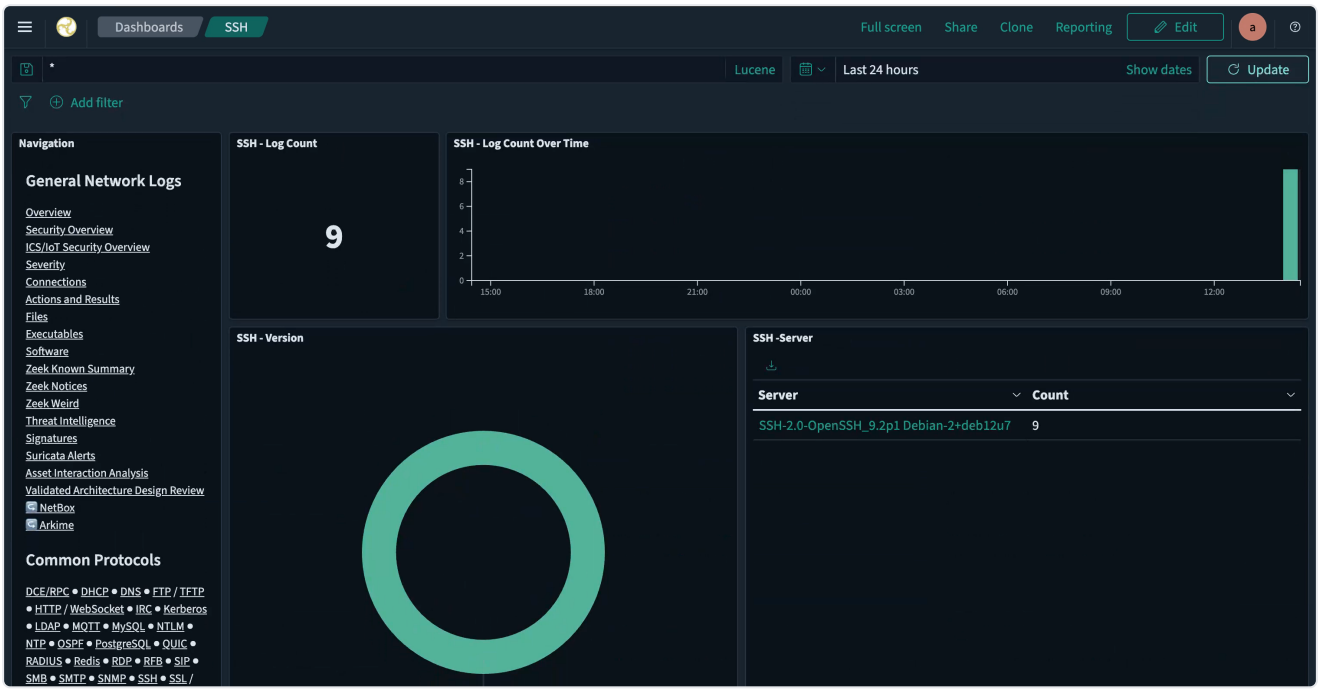
Malcolm es la plataforma de análisis de tráfico del laboratorio y ofrece cuadros de mando ya montados por protocolo; da la fotografía agregada. **Arkime**, accesible desde el propio Malcolm, guarda las sesiones una a una y permite consultarlas con expresiones de filtrado; da el detalle. Vamos a usar los dos en ese orden: primero el panorama, después la comprobación fina.

Paso 11. Ver la fotografía del tráfico SSH en el cuadro de mando de Malcolm

Desde la portada de Malcolm entramos en **Dashboards** y, en el cuadro de mando general, seleccionamos el panel **SSH**. Nos da directamente el conteo de tráfico SSH del periodo y su desglose.

Lo que muestra encaja con lo que ya sabíamos y añade matices: **9 registros SSH** en las últimas 24 horas, todos entre `172.31.0.2` y `172.18.2.100` por el puerto **22**, contra un servidor que se identifica como `SSH-2.0-OpenSSH_9.2p1 Debian`. El panel desglosa además los algoritmos negociados por cliente y servidor y las huellas HASSH de ambos extremos.

Ese desglose es útil porque caracteriza al atacante más allá de su dirección: la versión de cliente anunciada y la huella HASSH sirven para reconocer la misma herramienta si vuelve a aparecer desde otra IP.



Cuadro de mando SSH de Malcolm: 9 registros SSH en 24 horas contra un servidor SSH-2.0-OpenSSH_9.2p1 Debian.

Paso 12. Listar las sesiones SSH en Arkime

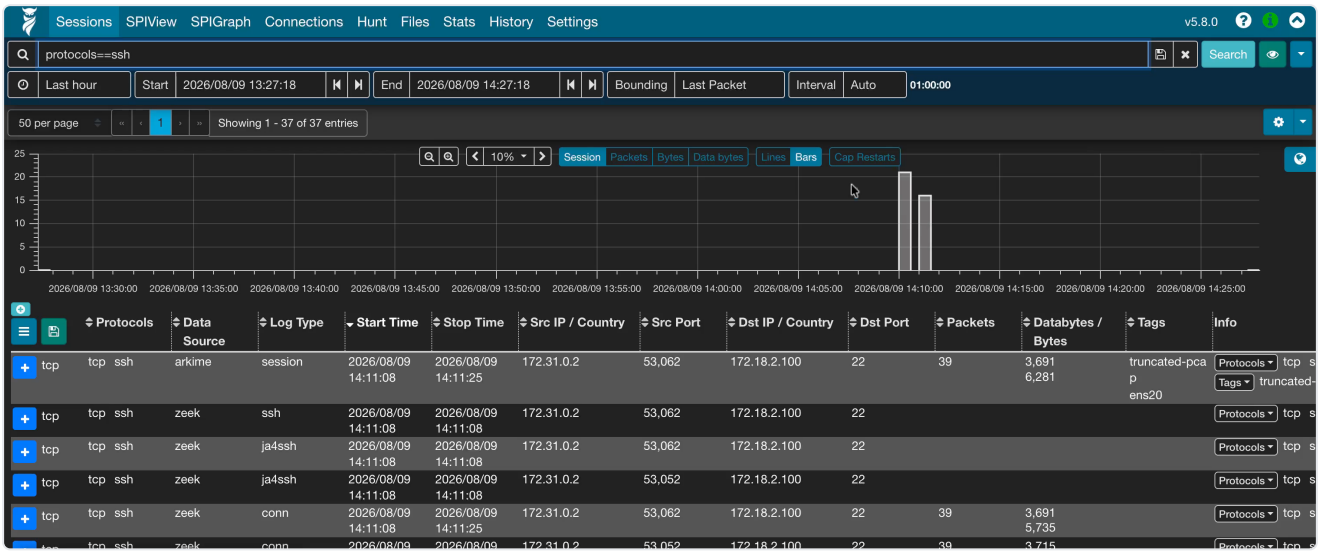
Pasamos a Arkime, donde podemos ejecutar consultas sobre las sesiones capturadas. La barra de búsqueda usa expresiones sencillas de tipo `campo == valor`, encadenables con `&&`.

Empezamos por lo más amplio, todo el tráfico SSH de la última hora:

```
protocols==ssh
```

Devuelve 37 sesiones. Al mirar las columnas se ve que todas comparten el mismo par: origen `172.31.0.2`, destino `172.18.2.100`, puerto destino `22`, con puertos de origen distintos en cada una (`53062`, `53052`, `53038`…) y todas concentradas en un intervalo de segundos, entre las 14:11:08 y las 14:11:25.

Ese es el retrato de una fuerza bruta: muchas conexiones cortas, seguidas, desde el mismo origen. Podemos estrechar el filtro añadiendo el origen y el destino — `protocols==ssh && ip.src==172.31.0.2 && ip.dst==172.18.2.100` — y el número apenas baja, lo que confirma que prácticamente todo el SSH del periodo es este ataque.



Arkime con el filtro protocols==ssh: 37 sesiones, todas de 172.31.0.2 hacia el puerto 22 de 172.18.2.100.

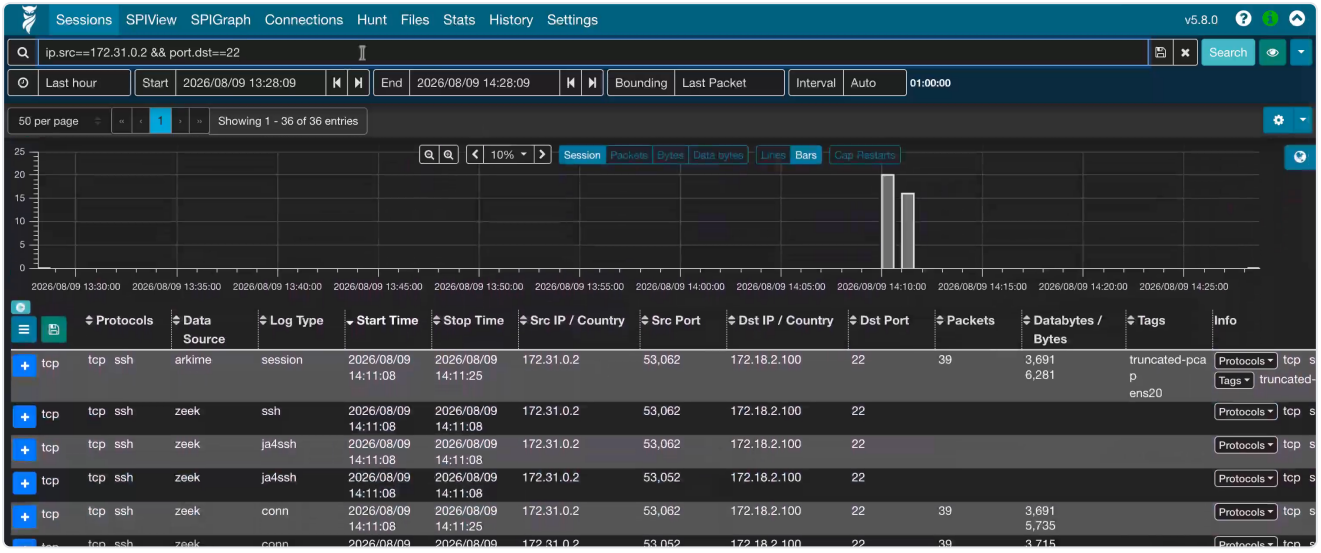
Paso 13. Comprobar si el atacante buscó otras máquinas

Ahora viene la pregunta que de verdad mide el alcance: ¿esta máquina ha intentado atacar el puerto 22 de *otros* equipos, o se ha centrado en uno solo? Para responderla quitamos el destino del filtro y dejamos únicamente el origen y el puerto de destino:

```
ip.src==172.31.0.2 && port.dst==22
```

Al dejar libre `ip.dst`, cualquier otra víctima aparecería en la columna de destino. El resultado son 36 sesiones y en todas ellas el destino sigue siendo `172.18.2.100`. No hay barrido: el atacante fue a por una única máquina.

> **Resultado:** Todas las sesiones hacia el puerto 22 tienen el mismo destino. El ataque está acotado a 172.18.2.100.



Filtro ip.src==172.31.0.2 && port.dst==22 en Arkime: las 36 sesiones tienen como único destino 172.18.2.100.

Paso 14. Ampliar la ventana para ver toda la actividad de la IP

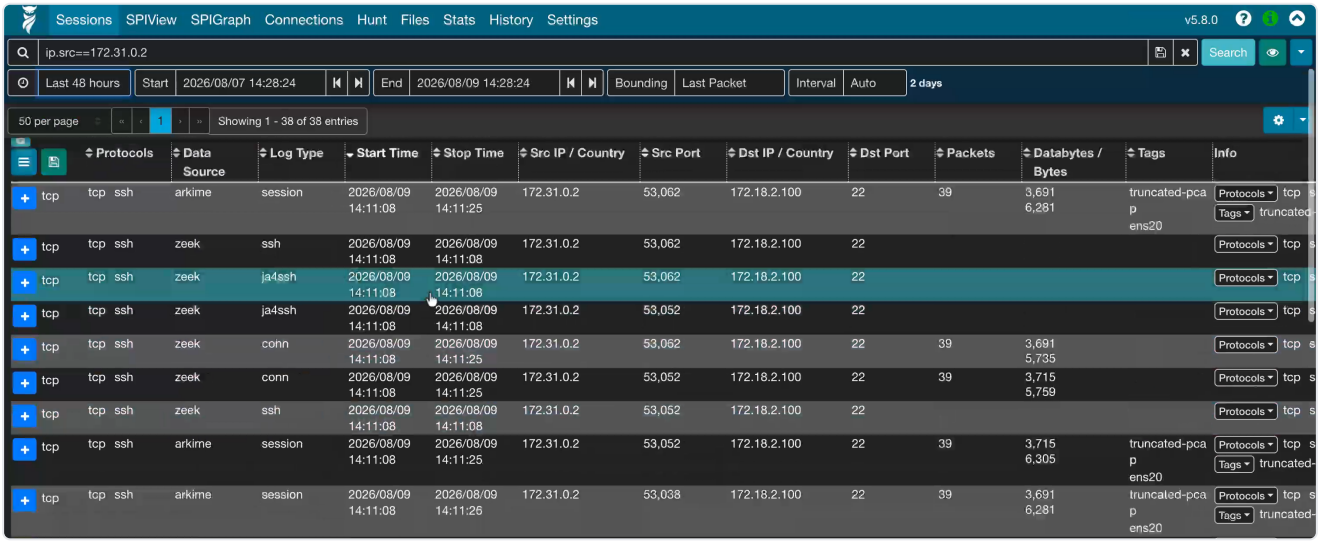
Queda un último cabo suelto. Todo lo anterior mira la última hora y solo el puerto 22; podría ser que esta dirección llevara días haciendo otras cosas en la red. Así que quitamos también la condición del puerto y ampliamos el rango temporal a las **últimas 48 horas**:

```
ip.src==172.31.0.2
```

Aparecen 38 sesiones y todas son SSH contra `172.18.2.100`. Fuera de este incidente, la dirección no ha hecho nada más en el tráfico capturado.

Con esto el N2 ya tiene el cuadro completo: hubo un intento de fuerza bruta por SSH, no se logró la autenticación, y la actividad de esa IP se limita a atacar una sola máquina. La IP no ha comprometido nada, pero se comporta como atacante, y eso es suficiente para aplicar la contención que el N1 había recomendado.

Evidencia que conviene guardar: Conviene guardar capturas de los conteos SSH en Arkime, Malcolm y Graylog en distintos momentos: son la prueba del alcance medido y sostienen la decisión de contención cuando el caso se revise más adelante.



Toda la actividad de 172.31.0.2 en las últimas 48 horas: 38 sesiones, todas SSH contra 172.18.2.100.

FASE 6

ANALISTA N2

Contención en el cortafuegos

La conclusión del análisis es que la IP parece maliciosa aunque no haya conseguido entrar, así que toca contenerla. En este laboratorio el cortafuegos es **OPNsense**, y el bloqueo no se hace creando una regla nueva para cada incidente, sino añadiendo la dirección a un **alias** que una regla ya existente utiliza.

La diferencia es importante para el trabajo diario de un SOC: un alias es una lista con nombre (aquí, `SOC_BLOCKED_IPS`) que agrupa direcciones; la regla que la bloquea se escribe una sola vez, y a partir de

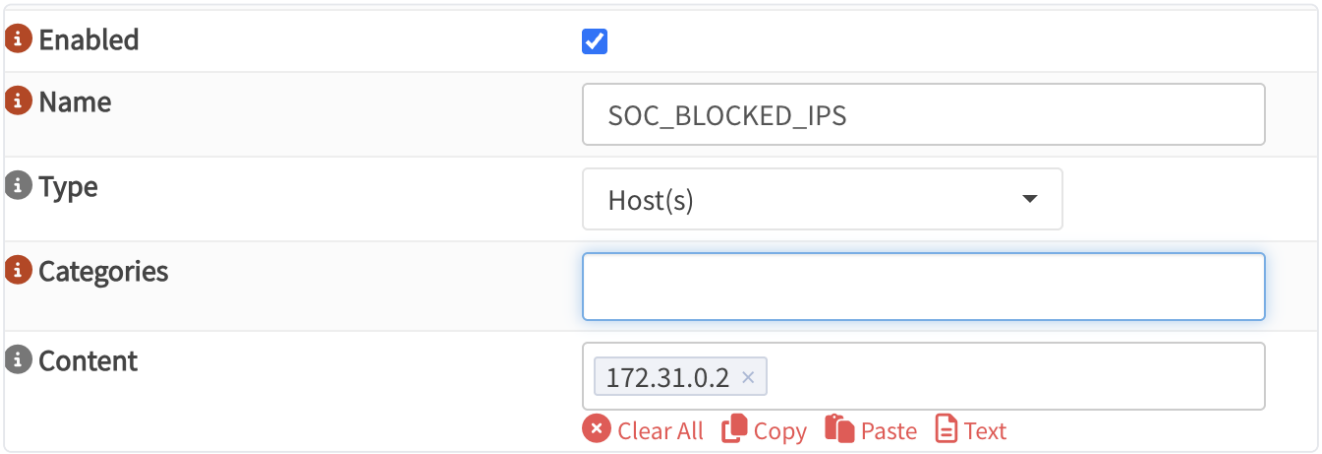
ahí contener una IP nueva es añadir una línea a la lista. Menos margen de error, revisión más sencilla y una única lista de la que quitar direcciones cuando dejen de ser relevantes.

Paso 15. Añadir la IP atacante al alias de bloqueo

Accedemos a la interfaz de OPNsense y vamos a **Firewall → Aliases**. Entre los alias definidos está `SOC_BLOCKED_IPS`, de tipo *Host(s)*, que es el que la regla de bloqueo del cortafuegos consume.

Lo editamos con el icono del lápiz y, en el campo **Content**, añadimos `172.31.0.2` como un elemento más de la lista. Guardamos con **Save**. Desde ese momento la dirección forma parte del conjunto que el cortafuegos deja fuera.

> **Nota:** Después de guardar, OPNsense muestra el aviso «After changing settings, please remember to apply them with the button below». En la grabación no llega a pulsarse ese botón **Apply**, de modo que el cambio queda registrado en la configuración pero no se ve confirmada su activación. Al reproducir la práctica hay que pulsarlo para que el bloqueo entre realmente en vigor.



Edición del alias SOC_BLOCKED_IPS en OPNsense, de tipo Host(s), con 172.31.0.2 añadida al contenido.

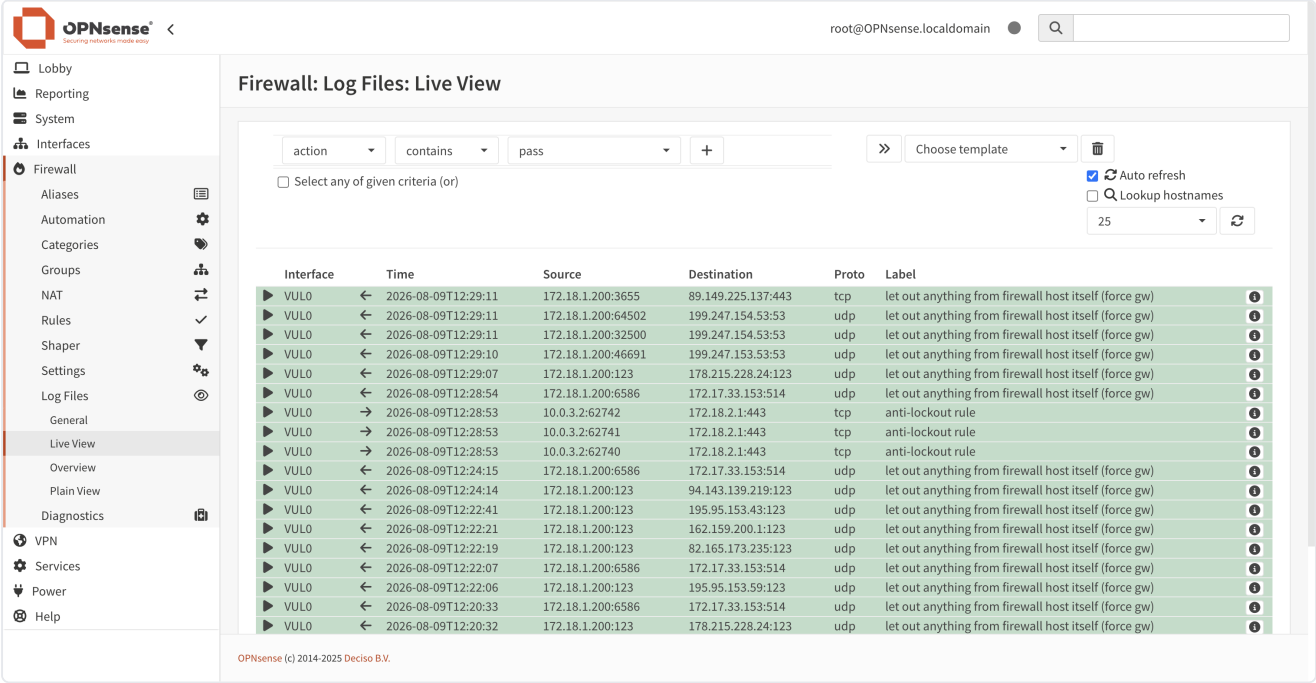
Paso 16. Intentar verificar el bloqueo en Live View

La contención no termina al guardar: hay que comprobar que funciona. Para eso OPNsense ofrece **Firewall → Log Files → Live View**, donde el registro del cortafuegos se ve en tiempo real y se puede filtrar por la acción aplicada a cada paquete.

La comprobación natural sería observar los descartes de la IP bloqueada, pero solo se producen si el atacante vuelve a intentar una conexión. En el momento de la revisión el registro solo muestra tráfico permitido del propio cortafuegos y no aparece nada procedente de `172.31.0.2`: la dirección ya había dejado de generar actividad.

Esa limitación no invalida la contención, pero sí hay que dejarla escrita. Una verificación que no se ha podido hacer es un dato del caso, no un detalle que se omite.

> **Decisión condicional**
>
> **Si el atacante vuelve a intentar conectarse:** en Live View aparecerán las entradas de descarte correspondientes a esa IP, y esa captura es la prueba de que el bloqueo funciona; conviene adjuntarla al caso.
>
> **Si no:** se documenta que la verificación no ha sido posible por falta de actividad de la IP, como se hizo aquí, y se deja la monitorización como recomendación de cierre.



Live View del cortafuegos durante la comprobación: solo tráfico permitido del propio OPNsense, sin actividad de 172.31.0.2.

FASE 7

ANALISTA N2

Documentación y cierre del caso

Todo el trabajo de investigación y contención vuelve ahora al portal central. Cerrar un caso no es cambiarle el estado: es dejar el expediente en condiciones de que alguien que no ha participado entienda qué pasó, qué se hizo y qué quedó pendiente.

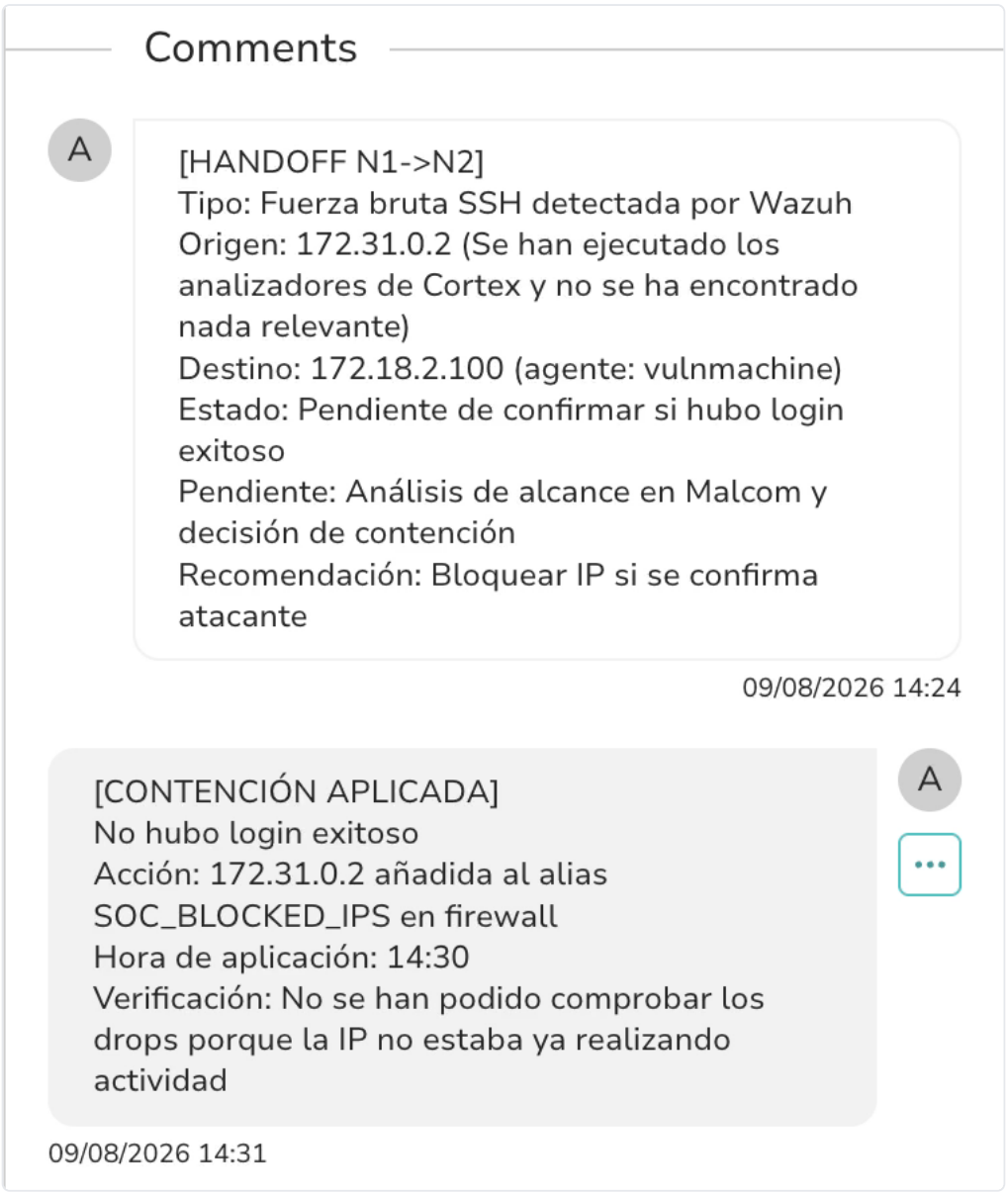
Se hace en tres movimientos: registrar la contención como comentario, adjuntar las evidencias recogidas por el camino y cerrar el caso con una clasificación y un resumen. Y queda un cabo por atar: la segunda alerta que vimos al principio.

Paso 17. Documentar la contención aplicada

Volvemos a los comentarios del caso y añadimos una segunda nota, con la misma lógica que el traspaso: hechos, acción, momento y verificación.

```
[CONTENCIÓN APLICADA]
No hubo login exitoso
Acción: 172.31.0.2 añadida al alias SOC_BLOCKED_IPS en firewall
Hora de aplicación: 14:30
Verificación: No se han podido comprobar los drops porque la IP
              no estaba ya realizando actividad
```

La **hora de aplicación** es la que permitirá después correlacionar el bloqueo con cualquier cambio de comportamiento en la red. Y la línea de **verificación** es la que da valor al comentario: dice exactamente qué se comprobó y qué no se pudo comprobar, en lugar de dar por hecho un resultado. Si en la organización se conoce el número de la regla del cortafuegos que aplica el bloqueo, ese es también un buen dato para añadir aquí.



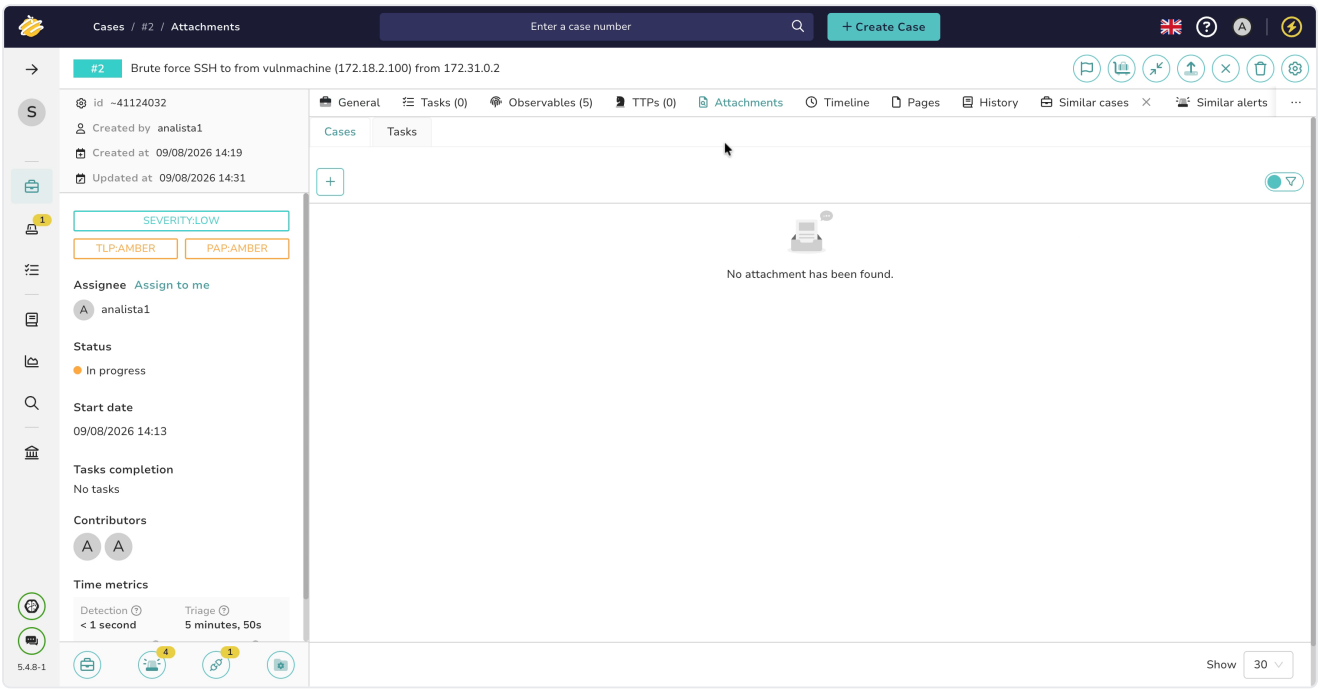
Los dos comentarios del caso: el traspaso N1→N2 y la nota de contención, con la hora de aplicación y el resultado de la verificación.

Paso 18. Adjuntar las evidencias del caso

La pestaña **Attachments** del caso es donde se guardan las pruebas gráficas: capturas de las consultas de Graylog, de los cuadros de mando de Malcolm, de las sesiones de Arkime y de la configuración del alias en OPNsense.

En la grabación esta pestaña se muestra pero no se llega a subir ningún archivo, así que el caso se cierra sin adjuntos. Queda planteado como el trabajo que corresponde hacer al reproducir la práctica: cada consulta que sostiene una conclusión del caso debería tener su captura correspondiente en esta pestaña.

Evidencia que conviene guardar: Capturas de la consulta de fuerza bruta y de la consulta de acceso aceptado en Graylog, del cuadro de mando SSH de Malcolm, de los filtros de Arkime y del alias SOC_BLOCKED_IPS ya modificado.



Pestaña Attachments del caso: en la grabación no llega a subirse ningún archivo.

Paso 19. Cerrar el caso con clasificación y resumen

Con la investigación documentada, pulsamos **Close**. TheHive pide tres cosas.

El **estado** lo marcamos como *True positive*: la detección era correcta, hubo realmente un intento de ataque por fuerza bruta. Que el atacante fracasara no convierte la alerta en un falso positivo; eso es una confusión frecuente y merece la pena tenerla clara.

El **impacto** lo marcamos como *No*, porque ninguna autenticación prosperó y no hubo efecto sobre el equipo.

El **resumen** recoge en seis líneas todo el recorrido:

```
Incidente: fuerza bruta SSH contra 172.18.2.100
Origen: 172.31.0.2
Impacto: Ninguno. Todos los intentos de autenticación fallaron.
Contención: IP bloqueada en firewall (alias SOC_BLOCKED_IPS)
Verificación: No ha sido posible porque la IP atacante no tenía actividad
Recomendaciones: Monitorizar la IP por si reaparece
```

Ese bloque es lo que se leerá dentro de seis meses si la misma dirección vuelve a aparecer, y por eso incluye tanto lo que se hizo como lo que quedó sin confirmar. Pulsamos **Confirm** y el caso queda cerrado.

> **Resultado:** El caso #2 se cierra como verdadero positivo sin impacto, con la contención aplicada y la monitorización de la IP como recomendación.

Diálogo de cierre del caso #2: estado True positive, impacto No y el resumen completo del incidente.

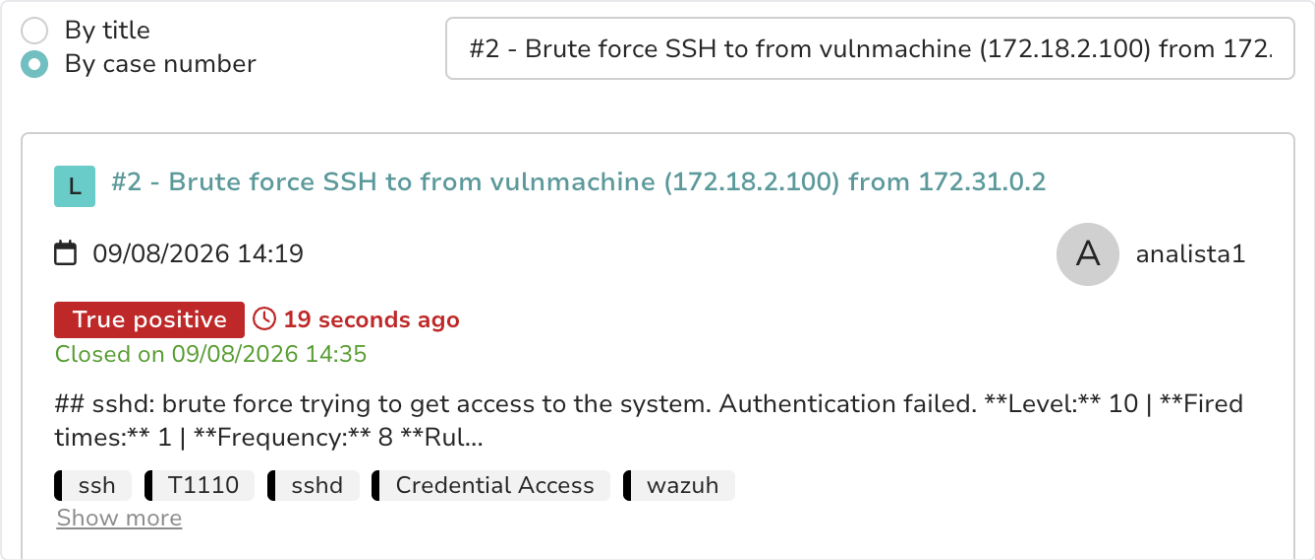
Paso 20. Resolver la alerta duplicada

Queda la segunda alerta del principio, la que describía el mismo ataque por otra vía de detección. Dejarla en la cola como *New* haría que otro analista la trabajase de cero, así que hay que resolverla.

Hay dos caminos. El primero es seleccionarla y cerrarla desde el menú de acciones marcándola como **Duplicate**: se saca de la cola dejando constancia de por qué. El segundo, que es el que seguimos aquí, es **Merge alert into case**: se busca el caso ya cerrado —por número o por título, en nuestro caso el `#2` — y se fusiona la alerta dentro de él.

La fusión es preferible cuando la segunda alerta puede aportar algo: sus observables y su información pasan directamente al expediente, de modo que todo lo relativo al incidente queda en un único sitio. TheHive confirma la operación con el mensaje *Alert(s) have been merged into case!* y la cola de alertas se queda vacía.

Con la alerta asociada al caso y el caso cerrado, el análisis de este incidente está terminado.



Diálogo de fusión de la alerta restante en el caso #2, ya cerrado como True positive.

Resumen del flujo de trabajo

Fase	Rol	Herramienta	Decisión principal
1	Analista N1	TheHive	Revisión de la cola de alertas, lectura de la alerta de sshd y creación del caso de trabajo.
2	Analista N1	TheHive + Cortex	Marcado de la IP atacante como IOC, ejecución de los analizadores de Cortex y etiquetado de la máquina víctima.
3	Analista N1 → N2	TheHive	Comentario de handoff con lo comprobado, lo que queda pendiente y la recomendación de contención.
4	Analista N2	Graylog (registros de Wazuh)	Búsqueda de los eventos de fuerza bruta y comprobación de que no hubo ninguna autenticación aceptada.
5	Analista N2	Malcolm y Arkime	Revisión del tráfico SSH capturado para medir el volumen del ataque y comprobar si hubo otros objetivos.
6	Analista N2	OPNsense	Bloqueo de la IP atacante mediante un alias de cortafuegos y comprobación en Live View.
7	Analista N2	TheHive	Registro de la contención, cierre del caso como verdadero positivo y tratamiento de la alerta duplicada.

Conclusión:

El caso se resuelve como un verdadero positivo sin impacto: hubo un intento de fuerza bruta por SSH desde `172.31.0.2` contra `172.18.2.100`, ninguna autenticación prosperó y la dirección quedó bloqueada en el alias `SOC_BLOCKED_IPS` del cortafuegos.

Más allá del desenlace, lo que conviene retener es el encadenamiento. Cada herramienta responde a una pregunta distinta y en un orden que no es casual: TheHive fija *qué* ha pasado y quién trabaja el caso; Cortex intenta decirnos si el indicador ya era conocido; Graylog responde a la pregunta decisiva sobre el éxito del ataque; Malcolm y Arkime miden hasta dónde llegó; OPNsense aplica la medida, y TheHive vuelve a recoger el resultado. Cada paso se justifica por lo que devolvió el anterior.

Merece la pena fijarse también en dos hábitos que atraviesan todo el recorrido. Uno es buscar la prueba en negativo: no basta con ver fallos, hay que consultar explícitamente si hubo un acceso aceptado. El otro es documentar los límites: cuando la verificación del bloqueo no fue posible, se escribió que no fue posible y por qué. Un caso bien cerrado no es el que no tiene huecos, sino el que los deja identificados.

Guía generada automáticamente usando MENTORA. MENTORA graba las acciones del profesor sobre las herramientas reales del SOC y produce automáticamente este material didáctico, garantizando que la documentación coincide con la práctica actual del aula.

Grabación MENTORA del 9 de agosto de 2026 · Tiempo activo: 18:42 · 20 pasos