Threat Hunting Hypothesis-Driven: metodo, pratica e maturità operativa

Negli ultimi tempi il threat hunting “hypothesis-driven” è diventato la forma più evoluta di difesa proattiva.

Non si tratta semplicemente di cercare indicatori nei log o di fare ricerche casuali in un SIEM, l’idea di fondo è proprio diversa: partire da un’ipotesi concreta su come potrebbe agire un avversario e verificare, in modo strutturato, se nei nostri sistemi esistano tracce compatibili con quel comportamento.

In questo approccio il Threat hunter non aspetta un alert dato da una console. Non reagisce: anticipa.

Parte da una domanda investigativa costruita sulla base di intelligence, conoscenza delle TTP di tutti i Threat Actor (noti e non noti) e la reale comprensione dell’infrastruttura aziendale. L’obiettivo non è trovare “qualcosa di strano”, ma cercare evidenze coerenti con uno scenario d’attacco plausibile, anche se mai ancora intercettato dai meccanismi di detection tradizionali.

Threat Hunting Hypothesis-Driven: l’evoluzione dell’hunting proattivo

La maturità di un programma di threat hunting non si costruisce in un giorno, è un percorso evolutivo piuttosto chiaro (e prolungato).

All’inizio, molte organizzazioni svolgono attività sporadiche, spesso legate a IOC statici o a esigenze spot. Il logging a disposizione è spesso limitato, le analisi sono manuali e guidate dal buon senso, di conseguenza, non esiste un processo

formalizzato. In questa fase l’hunting è più un’iniziativa individuale che un servizio strutturato.

Con il tempo, però, l’approccio può maturare grazie ad esperienza e lesson learned (se viene svolto anch’esso in modo strutturato). L’Hunting Maturity Model (https://www.sans.org/tools/hunting-maturity- model) descrive questa evoluzione considerando diversi fattori: qualità e ampiezza delle fonti dati, formalizzazione delle ipotesi e scopes, competenze analitiche del team e integrazione con il detection engineering del SOC o dell’Incident Response Team.

Quando la maturità aumenta, crescono anche le fonti disponibili: endpoint telemetry, network data, identity logs, audit trail cloud. Framework come MITRE ATT&CK o MITRE D3FEND iniziano a essere usati per modellare le ipotesi e strutturare i report. Nei livelli più avanzati, l’hunting diventa ciclico, misurabile e integrato nel miglioramento continuo: ogni attività produce nuove regole, identifica gap di visibilità o rafforza la copertura esistente.

Un passaggio cruciale in questo percorso è l’abbandono dell’approccio “IOC driven” statico in favore di quello “TTP driven”. Gli indicatori sono fragili, cambiano rapidamente e possono essere facilmente aggirati. Le tecniche, invece, riflettono il modus operandi dell’avversario e tendono a essere più stabili nel tempo.

Ma lavorare sulle TTP richiede esperienza reale e conoscenza degli attori, esposizione a casi concreti, capacità di riconoscere varianti e polimorfismi delle tecniche durante attacchi effettivi.

Un hunting efficace non nasce nel vuoto. Senza intelligence, l’ipotesi rischia di essere astratta o scollegata dal contesto reale.

La Cyber Threat Intelligence fornisce il punto di partenza: attori attivi, campagne in corso, tecniche emergenti, pattern

infrastrutturali. Tuttavia, il valore non sta nel report in sé, ma nella capacità di trasformarlo in una domanda operativa.

Se un report descrive l'abuso di token OAuth in ambienti cloud, non è sufficiente conoscere hash o domini. La domanda diventa: "Se un attore con queste capacità stesse operando nel nostro tenant, quali tracce comportamentali dovremmo osservare?"

Da qui nascono query su anomalie nei consensi applicativi, uso atipico di API, persistenza tramite service principal o comportamenti anomali su identity logs.

L'integrazione deve essere bidirezionale. La CTI alimenta l'hunting, ma i risultati dell'hunting evidenze, falsi positivi ricorrenti, nuove tecniche osservate devono a loro volta arricchire il ciclo di intelligence. Solo così si crea un ecosistema realmente dinamico.

Proviamo a vedere assieme un esempio applicato:

Un report CTI segnala che un gruppo sta abusando di service principal in Microsoft 365 per ottenere persistenza tramite permessi API elevati.

Il team di Threat Hunting parte da questa informazione e verifica creazioni recenti di service principal, assegnazioni anomale di privilegi e autenticazioni sospette. Durante l'analisi emergono però due aspetti inattesi: un'applicazione legittima che replica parte di quel comportamento (falso positivo ricorrente) e una tecnica non documentata di assegnazione temporanea dei permessi seguita da revoca immediata. Queste evidenze vengono condivise con il team CTI, che aggiorna il proprio dataset inserendo il nuovo pattern e riclassificando alcuni indicatori come deboli.

In questo modo l'intelligence genera l'hunting, ma è l'hunting stesso a renderla più precisa e aderente al contesto reale.

Il “TTP driven” hunting si basa su una struttura logica chiara:

- scope
- comportamento atteso
- artefatti generati
- fonti dati disponibili
- criteri di validazione

Prendiamo un caso classico: LSASS memory dumping. L’errore sarebbe cercare direttamente “Mimikatz” “mimi*”.

L’ipotesi corretta per questo tipo d’attività è più ampia: “Un attore che intende effettuare credential access potrebbe generare accessi sospetti al processo LSASS, creare handle privilegiati o produrre dump di memoria anomali.”

L’analisi si concentra quindi su eventi EDR relativi a process access, creazione di file dump, utilizzo inconsueto di API di debugging, e se il logging lo permette (difficile) valutare le syscalls utilizzate nella finestra temporale in scope. Non si cerca uno strumento specifico, ma un comportamento.

In scenari più complessi, ad esempio lateral movement tramite WebDAV combinato con NTLM relay in ambienti ibridi, l’ipotesi diventa molto più articolata. Non si cerca “ntlmrelayx”, ma si formula una correlazione comportamentale.

Nella mente del threat hunter potrebbe esserci una frase di questo tipo:

“Se un attore stesse tentando un relay NTLM via WebDAV, potremmo trovare autenticazioni NTLM anomale verso servizi interni, precedute da traffico WebDAV outbound atipico.”

L’attività di hunting si sposta quindi sulla correlazione temporale tra:

- richieste HTTP con metodo PROPFIND o traffico WebDAV inconsueto,

- autenticazioni NTLM verso target interni non usuali,
- accessi SMB o HTTP privi di signing

Anche in assenza di compromissione confermata, un’attività del genere può portare alla scoperta di debolezze strutturali: SMB signing non abilitato, Extended Protection non enforced, logging insufficiente. Il valore, quindi, non è solo individuare un attacco, ma rafforzare l’architettura.

Architettura e strumenti: la base tecnica

Un approccio hypothesis driven richiede una base dati coerente e realmente interrogabile. Se, ad esempio, i log di identity vengono conservati solo per 7 giorni, un’ipotesi su una persistenza di 30 giorni diventa semplicemente non verificabile. Allo stesso modo, un SIEM con parsing incompleto dei log cloud può impedire di correlare un’attività sospetta su Azure AD con un evento su endpoint, frammentando l’analisi.

In ambienti più complessi, un’architettura data lake-oriented consente di centralizzare log eterogenei (endpoint, network, SaaS, IaaS) e di normalizzarli. Questo permette, ad esempio, di testare un’ipotesi che correli autenticazioni anomale, creazione di nuove VM e traffico verso ASN a rischio in un’unica query, anziché tramite analisi manuali separate.

La scelta degli strumenti incide direttamente sulla profondità dell’hunting: senza EDR con visibilità su process tree e command-line, non è possibile validare ipotesi su tecniche di “living off the land”; senza NDR, un beacon a basso rumore su protocollo DNS può rimanere invisibile; senza strumenti di identity analytics, pattern di privilege escalation graduale possono sembrare attività amministrative legittime.

Anche il linguaggio di query è parte dell’architettura: un ambiente che supporta KQL o SPL con join efficienti e funzioni

di aggregazione avanzate consente pivot rapidi e analisi iterative. Se invece il motore non gestisce correlazioni complesse o soffre di latenze elevate, l'hunting perde profondità e diventa superficiale.

Nei contesti più maturi, la disponibilità di modelli comportamentali e baseline storiche abilita analisi di deviazione reali: ad esempio, identificare un service account che accede per la prima volta a una risorsa sensibile fuori dal proprio perimetro abituale.

L'automazione accelera enrichment, contestualizzazione e raccolta di indicatori, ma non sostituisce il ragionamento. Quando l'ipotesi riguarda una tecnica emergente o un comportamento ambiguo, è la qualità dell'architettura sottostante retention, normalizzazione, capacità di correlazione a determinare se l'hunter potrà davvero dimostrarla o dovrà fermarsi a un sospetto.

Elementi infrastrutturali fondamentali per il Threat Hunting

- **Retention adeguata dei log**
  - almeno 6–12 mesi per identity, endpoint e cloud activities
- **Normalizzazione e data modeling coerente tra fonti diverse**
- **Visibilità endpoint avanzata**
  - process tree, command-line, registry, scheduled task, network connections
- **Telemetria di rete est–ovest e north–south**
  - con capacità di analisi comportamentale
- **Logging completo del piano di controllo cloud**
  - audit log, API call, IAM changes
- **Identity telemetry dettagliata**
  - token issuance, conditional access, MFA events, service principal activity
- **Motore di query performante e flessibile**
  - con supporto a join, aggregazioni e funzioni

temporali
- **Integrazione CTI strutturata**
  - con indicator scoring e gestione della qualità degli IOC
- **Capacità di enrichment automatizzato**
  - WHOIS, passive DNS, reputation, asset context
- **Baseline comportamentali e dataset storici**
  - Anche se questo elemento solitamente è molto complesso da ottenere ed indica una grande maturità aziendale, per le analisi di deviazione strutturate resta un asset fondamentale

In assenza completa o parziale di questi elementi, l'hunting tende a ridursi a ricerca reattiva di indicatori; con essi, diventa un'attività esplorativa e realmente orientata alla scoperta.

Uno degli errori più comuni è considerare l'hunting un esercizio teorico. In realtà, ogni ciclo deve produrre risultati tangibili.

Gli output principali si possono ricondurre a tre categorie:

1. Compromissioni identificate
2. Miglioramenti al detection engineering
3. Aumento della visibilità del rischio

Anche un hunting che non rileva incidenti può avere grande valore. Ad esempio, può evidenziare che una tecnica ATT&CK non è coperta da alcuna telemetria disponibile.

La formalizzazione è fondamentale: descrizione dell'ipotesi, dataset analizzati, query utilizzate, risultati ottenuti e decisioni operative. Questo garantisce tracciabilità, auditability e, in contesti regolamentati, dimostrazione di due diligence.

Il threat hunting hypothesis-driven è, per natura, iterativo. Ogni attività alimenta la successiva.

Se un’ipotesi si rivela valida e genera una nuova detection rule (ad esempio implementata anche tramite Sigma o Yara), quella tecnica entra nel monitoraggio continuo, riducendo la necessità di future analisi manuali sullo stesso pattern.

Se emergono gap come l’assenza di logging su specifici eventi, questi devono tradursi in remediation tecnica. In questo modo l’hunting diventa uno strumento diretto di crescita della maturità complessiva.

Il feedback coinvolge anche il team: le tecniche analizzate, le query sviluppate e i casi reali diventano materiale per esercitazioni e tabletop, rafforzando il pensiero analitico e la capacità di modellare ipotesi.

Gli strumenti sono importanti, ma non sono l’elemento decisivo.

Un threat hunter efficace deve saper ragionare per ipotesi, evitare bias cognitivi e comprendere a fondo il funzionamento reale di sistemi operativi, protocolli di rete, Active Directory e workload cloud. Senza una chiara percezione del comportamento “normale” dell’infrastruttura, è impossibile riconoscere l’anomalia significativa.

Serve padronanza dei linguaggi di query, capacità di correlare fonti eterogenee e lettura critica della telemetria. Ma soprattutto serve metodo nel trasformare un insight di CTI o una TTP in una domanda tecnica verificabile sui dati.

Il valore del threat hunting non dipende solo dalla tecnologia adottata (anzi, la tecnologia in questi casi rappresenta solo una parte del valore legata a questo tipo d’attività). Il risultato, infatti, dipende dalla qualità analitica di chi la utilizza e dalla capacità dell’organizzazione di integrare l’hunting in un processo strutturato, misurabile e orientato al miglioramento continuo.

Stay Safe. Be Proactive.

Profilo Autore

Nicolas Fasolo è il Team Leader del Incident Response Team di Yarix (Vargroup). Nel tempo libero lavora come “Security Researcher” e “Malware Developer” indipendente con una passione sfrenata per l’analisi del malware. Durante il suo percorso formativo di certificazione Master CEH ha ottenuto il Top 1 al mondo per il “Quarter 4 Dicembre 2021”. Autore di “Cybersecurity Podcast” e “Cybersecurity Warrior”.

Condividi sui Social Network:
https://www.ictsecuritymagazine.com/articoli/threat-hunting-hypothesis-driven/