WHITE PAPER

Threat Hunting with the Vectra AI Platform:

Uncovering Compliance Gaps, IOCs, and Attacker Tactics In Your Modern Network

VECTRA®

Table of Contents

Introduction

What is Threat Hunting?

Threat hunting is the proactive search for threats beyond alerts. Instead of waiting for an alarm to go off, hunters start from the assumption that something could already be wrong and go searching for evidence. They combine threat intel, behavioral clues, and what they know about their own environment to surface activity that doesn't fit the norm.

Threat Hunting in Modern Networks against Modern Attacks

Modern enterprise networks are sprawling ecosystems. They're no longer just data centers and offices - they now span cloud data centers, SaaS apps, remote users, IoT and OT systems, and identity services like Entra ID. Each layer brings more complexity and more blind spots. Threat hunting gives security teams a way to see across all of it -connecting dots that might otherwise stay hidden and revealing risks before they become incidents.

What This Guide Covers

This guide walks through how to use the Vectra AI Platform to uncover indicators of compromise, compliance violations, and attacker techniques. You'll also learn how to fold those findings back into your detection and response workflow.

While Vectra AI already surfaces high-fidelity detections for known and emerging threats, hunting helps fill in the gaps - spotting the subtle behaviors that don't yet trigger alerts and giving teams a chance to act before damage occurs.

Why Threat Hunting Matters?

Threat hunting helps you understand your modern hybrid network so that you are better prepared during incident response.

Threat hunting offers tangible proactive benefits. At its core, it helps security teams understand their environment deeply, reduce risk, and respond to threats faster. By continuously exploring data, validating assumptions, and uncovering hidden risks, hunting makes your organization more prepared, more resilient, and more agile when incidents arise.

Proactive threat hunting helps detect advanced threats an average of

11 days earlier

and saves

$1.3M per incident

(Gartner - Prioritize Threat Hunting for the Early Detection of Stealthy Attacks Oct 2025)

Here's what consistent threat hunting enables:

1. **Understand your environment to respond faster**

   Regular exploration of network and user activity builds deep familiarity with how systems behave -making it easier to recognize when something's off.

2. **Discover Indicators of Compromise (IoCs)**

   Search for traces like suspicious domains, IP addresses, file hashes, user agents or OAuth tokens, API keys, registry entries that hint at attacker activity.

3. **Identify Compliance Violations**

   Hunting provides visibility into unauthorized access, insecure data transfers, or activity that breaches internal policies or regulatory standards. This supports efforts to maintain compliance with frameworks like GDPR, HIPAA, or PCI-DSS.

4. **Improve Proactive Defense Posture**

   Security teams can uncover misconfigurations, unsafe defaults, and unmonitored services that introduce unnecessary risk. These insights inform detection tuning, policy updates, and preventive hardening to strengthen overall defenses.

5. **Establish Behavioral Baselines**

   By defining what "good" looks like, analysts can triage faster, identify anomalies, and prioritize investigation efforts.

6. **Accelerate investigations and reduce dwell time**

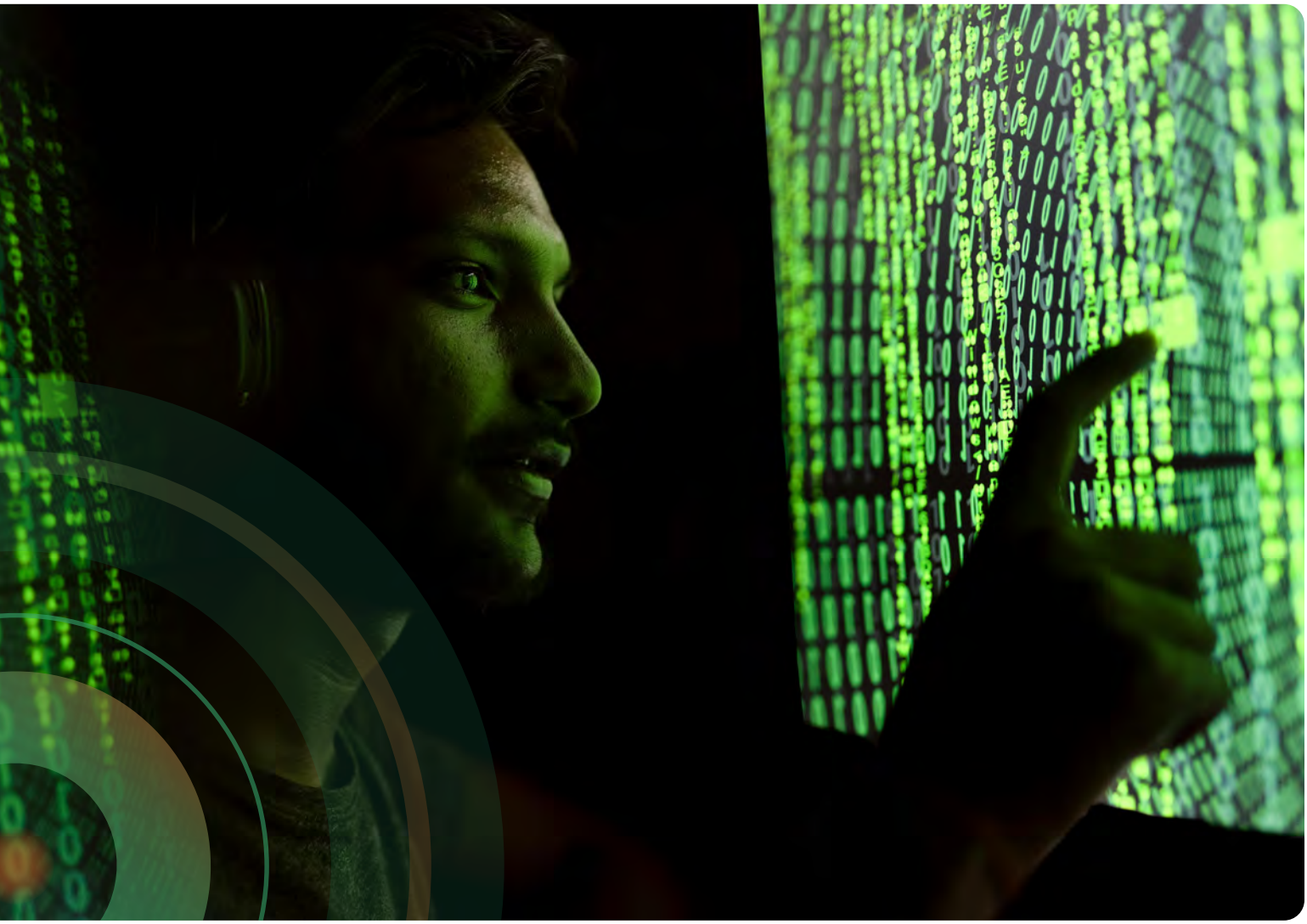
   Familiarity with your tools and telemetry streamlines triage and investigation, reducing attacker dwell time and improving your mean time to respond (MTTR).

What Data Powers Effective Threat Hunting?

You can't hunt what you can't see.

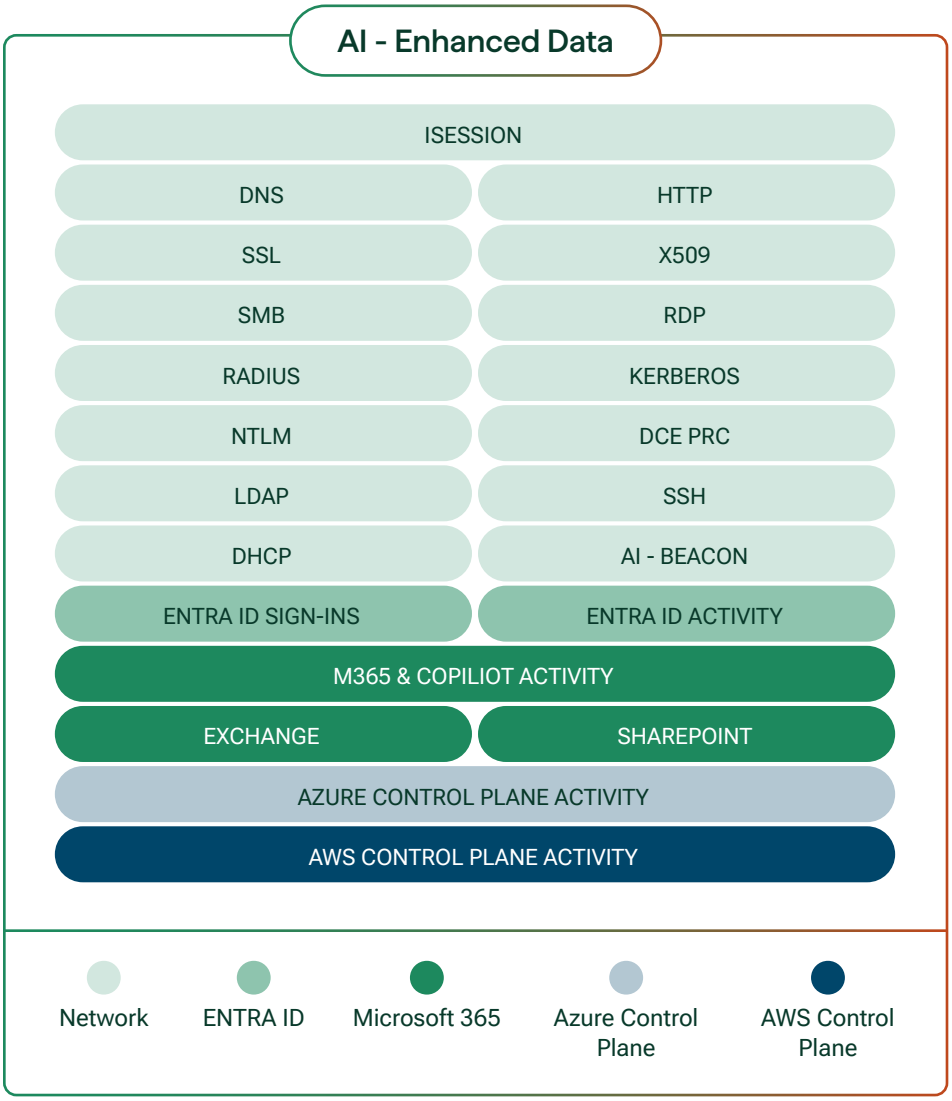
Visibility across the network, endpoints, identities, and cloud is what turns data into insight. The more complete and correlated that data is, the better the outcome.

Network metadata, in particular, provides unmatched context—it shows how systems communicate, not just that they did. Endpoint logs tell you what happened on a single machine; network data shows how the pieces fit together. Seeing those relationships in motion is what turns isolated logs into actionable intelligence.



Overview of Vectra AI AI-enhanced Metadata

The following is a quick reference to the available metadata and the common attributes for each metadata stream.



Below you can find detailed descriptions of the supported fields for metadata attributes:

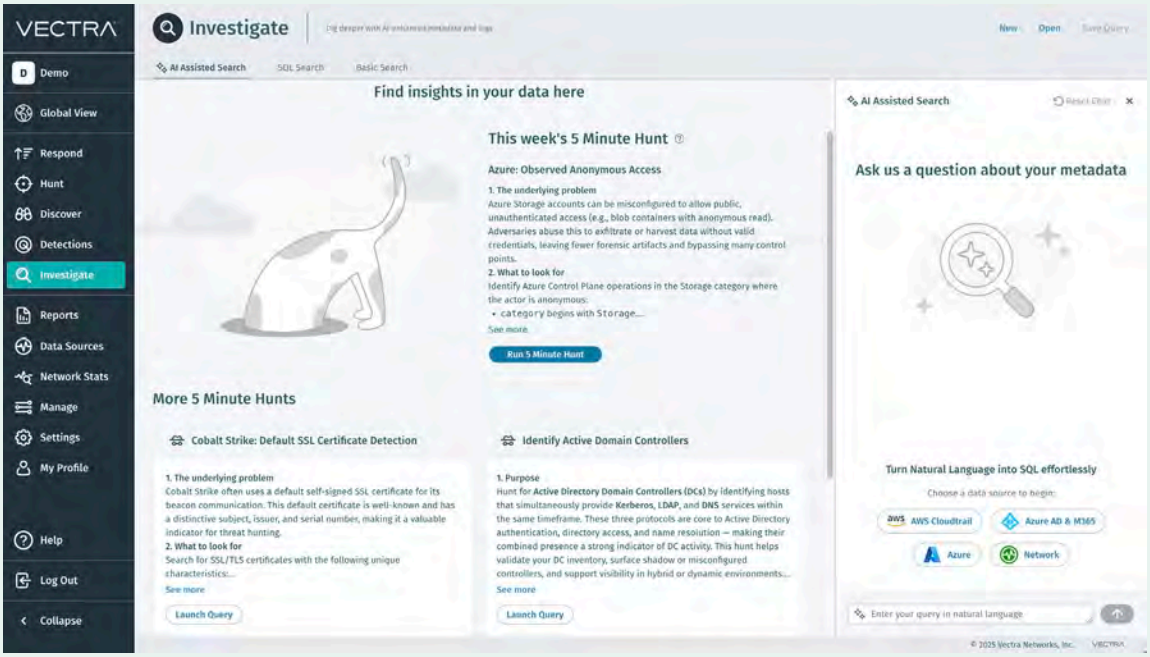
- Network
- AWS
- Microsoft 365
- Azure

Threat Hunting in the Vectra AI Platform

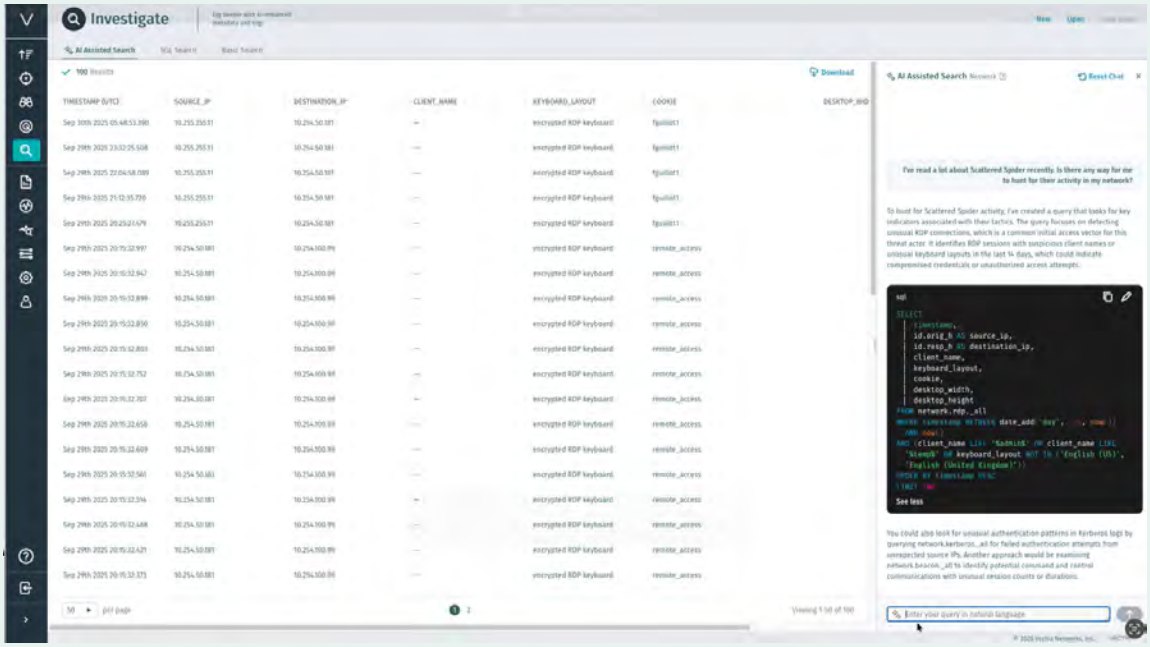
The Vectra AI Platform is purpose-built to power advanced threat investigations and hunting at scale.

Analysts can quickly pivot across rich, AI-enhanced metadata to uncover attacker behaviors, trace activity over time, and correlate signals across identity, cloud, and network layers.

The Investigate landing page within the Vectra AI Platform (Respond UX)



The Vectra AI Platform delivers the context security teams need to investigate, and hunt hybrid attacks in real-time



Basic Search/ SQL Search	• Run structured queries to test hypotheses, track TTPs, or investigate compliance gaps across hybrid environments.
AI Assisted Search	• Ask investigation and hunting questions in plain language and get immediate, context-rich answers plus recommended next steps..
5 Minute Hunts	• Start quickly with expert-built searches, curated by Vectra AI's security researchers, that highlight common attacker behaviors and threat patterns.
Saved Searches	• Click "Open" to access a list of pre-built queries curated by Vectra AI's security researchers • Save and reuse custom queries tailored to your environment for faster, repeatable hunts.
Open Source: GitHub Library	• Get the latest hunting queries and contribute your own on the Vectra AI Threat Hunting GitHub—a living community resource.

TTP-based Hunting

TTP-based hunting focuses on identifying attacker behaviors rather than relying on static indicators.

By targeting specific tactics, techniques, and procedures used during real-world intrusions, security teams can detect threats that often evade signature-based tools. These hunts are designed to surface early-stage compromise, highlight security gaps, and provide insights into how attackers operate within your environment.

1. DPAPI Backup Key Retrieval

What this query finds

This query identifies attempts to retrieve DPAPI backup keys from domain controllers — catching suspicious access patterns to domain controller resources, unusual administrative queries against Active Directory, and the use of known techniques for DPAPI key extraction.

Hunt Logic

The query checks network DCE/RPC logs from the last 14 days for events where a system connects to a domain controller using the LSARPC endpoint and performs the lsarretrieveprivatedata operation. This operation is widely used by hacking tools (like Mimikatz, SharpDPAPI, DSInternals) to extract DPAPI backup keys (Sources:).

What's the Security Implication?

DPAPI (Data Protection API) backup keys are critical security components that Windows uses to encrypt sensitive user data, including saved passwords, certificates, and other credentials stored on domain-joined systems. If your query finds this activity, it could mean an attacker is attempting to steal domain-wide encryption keys—potentially enabling them to decrypt all stored passwords and credentials across your network, escalate privileges, harvest sensitive data, and maintain long-term control.

Query

```
SELECT timestamp, orig_hostname, id.orig_h as "id_orig_h", id.resp_h as "id_resp_h",
resp_hostname, domain, username, endpoint, hostname, operation, sensor_uid

FROM network.dce_rpc

WHERE id.orig_h = '10.254.50.142' AND LOWER(endpoint) = 'lsarpc' AND
LOWER(operation) = 'lsarretrieveprivatedata' AND timestamp > date_add('day', -14,
now())

ORDER BY timestamp DESC

LIMIT 100
```

2. Certutil Binary Usage

What this query finds

This query identifies instances where the certutil.exe binary is being used for potentially malicious purposes, including downloading files from external websites, decoding base64-encoded content, or performing other operations beyond standard certificate management tasks.

Hunt Logic

The query scans HTTP traffic from the last 14 days for requests with this user agent Microsoft-CryptoAPI/10.0, which is created when certutil downloads files from external sources. It groups and counts these by host, revealing any suspicious download patterns or large volumes tied to certutil usage.

What's the Security Implication?

Certutil is a legitimate Windows command-line utility designed for certificate management, but it has become a favored tool among cybercriminals due to its ability to download files from the internet and perform encoding operations while appearing as normal system activity. If your query finds such activity, it could mean attackers are abusing the legit certutil tool to sneak in malware, exfiltrate data, or run obfuscated scripts—making detection tough since it blends in with normal system operations. Understanding certutil usage patterns is crucial for identifying potential compromise before attackers can establish persistence or move laterally through your network.

Query

```
SELECT host, COUNT(*) AS request_count
FROM network.http
WHERE user_agent = 'Microsoft-CryptoAPI/10.0' AND timestamp > date_add('day', -14, now())
GROUP BY host
ORDER BY request_count DESC
LIMIT 50
```

3. Domain Controller Initiating NTLM Authentication

What this query finds

This query identifies instances where a Domain Controller is initiating NTLM authentication requests to other systems, rather than responding to authentication requests as would be expected in normal operations. Domain Controllers typically act as authentication servers that validate credentials from clients, so when they initiate outbound NTLM authentication, it represents an anomalous behavior pattern that warrants investigation.

Hunt Logic

The query scans NTLM traffic from the last 14 days for requests with domain controllers as source IP addresses.

What's the Security Implication?

This behavior is a strong indicator of NTLM relay attacks or coerced authentication scenarios where an attacker has compromised a Domain Controller and is using it to authenticate to other systems in the environment. Since Domain Controllers have extensive privileges and trust relationships throughout the domain, a compromised DC initiating authentication can lead to lateral movement, privilege escalation, and potential domain-wide compromise. Attackers may use techniques like PetitPotam, PrinterBug, or other coercion methods to force the Domain Controller to authenticate to attacker-controlled systems, potentially exposing the computer account's NTLM hash. This represents a critical security incident requiring immediate investigation and containment, as the compromise of a Domain Controller can provide attackers with the keys to the entire domain infrastructure.

Query

```
SELECT id.orig_h, orig_hostname.name AS hostname, COUNT(*) AS ntlm_request_
count
FROM network.ntlm
WHERE (LOWER(orig_hostname.name) LIKE '%dc%' OR TRY_CAST(id.orig_h AS
IPADDRESS) BETWEEN IPADDRESS '10.254.100.0' AND IPADDRESS '10.254.100.255')
AND timestamp > date_add('day', -14, now())
GROUP BY id.orig_h, orig_hostname.name
ORDER BY ntlm_request_count DESC
LIMIT 100
```

4. Coerced Authentications

What this query finds

This query detects coerced authentication attacks targeting specific Remote Procedure Call (RPC) interfaces and operation numbers (opnums) commonly exploited to force Windows systems to authenticate to attacker-controlled servers. The query identifies suspicious RPC calls to MS-RPRN (Print System Remote Protocol), MS-EFSR (Encrypting File System Remote Protocol), MS-FSRVP (File Server Remote VSS Protocol), and MS-DFSNM (Distributed File System Namespace Management Protocol) using their known exploitation opnums associated with PrinterBug, PetitPotam, ShadowCoerce, and DFSCoerce attacks respectively.

Hunt Logic

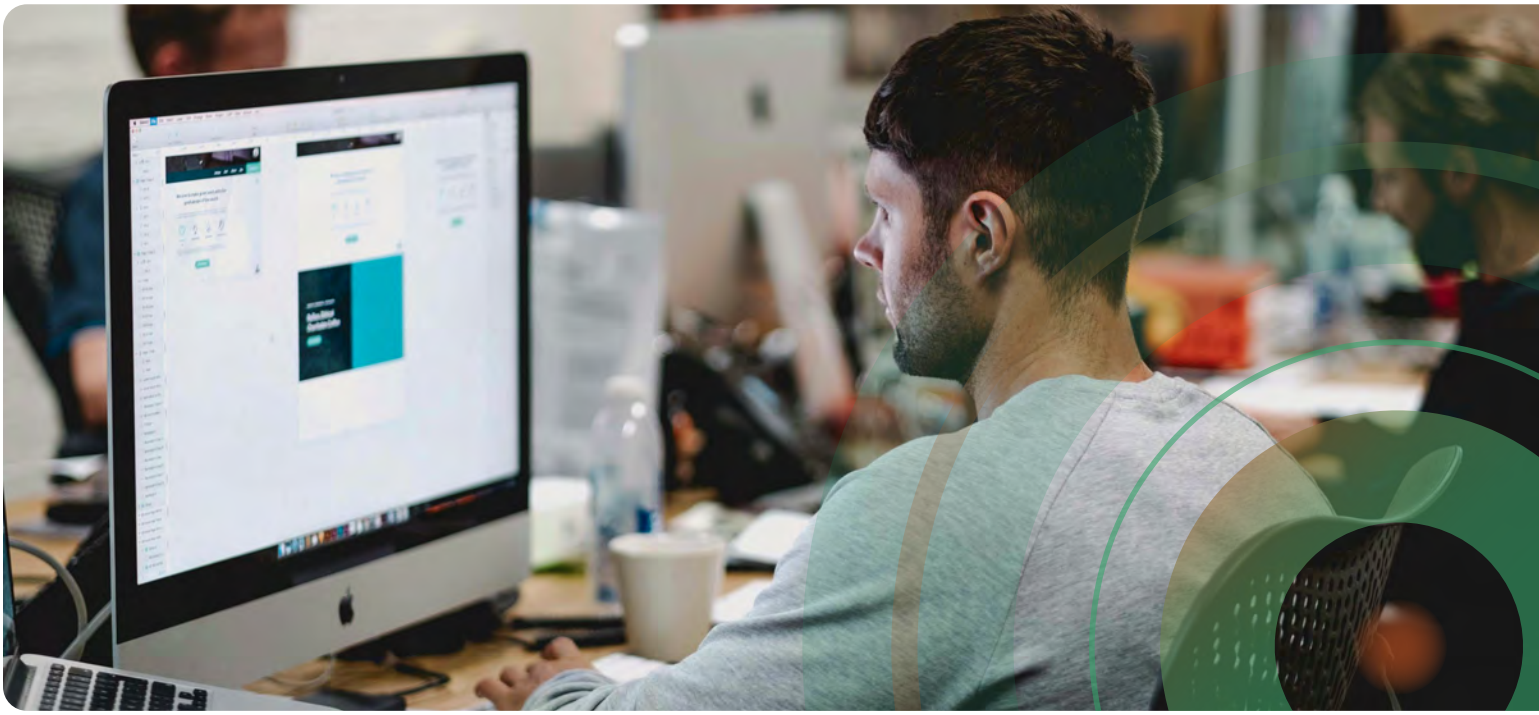
The query examines RPC traffic metadata to identify calls to vulnerable Windows protocols using specific operation numbers that correspond to known coercion techniques. PrinterBug exploits MS-RPRN opnum 65 (0x41) to trigger authentication, while PetitPotam targets multiple MS-EFSR opnums (0, 4, 5, 6, 7, 12, 13, 15) to coerce authentication through the Encrypting File System service. ShadowCoerce leverages MS-FSRVP opnums 8 and 9 (0x08, 0x09) via the Volume Shadow Copy service, and DFSCoerce uses MS-DFSNM opnums 12 and 13 (0x0c, 0x0d) through the Distributed File System service. The detection logic focuses on identifying this specific protocol and opnum combinations, particularly when originating from unexpected sources or targeting high-value systems like Domain Controllers. The query may also correlate timing patterns and source/destination relationships to distinguish between legitimate administrative activities and malicious coercion attempts.

What's the Security Implication?

These coerced authentication techniques represent significant security threats as they can force privileged systems, particularly Domain Controllers, to authenticate to attacker-controlled servers without user interaction. Successful exploitation can lead to NTLM relay attacks, credential theft, and potential compromise of computer accounts with elevated privileges. When targeted against Domain Controllers, these attacks can expose machine account credentials that provide extensive domain privileges, enabling attackers to perform DCSync attacks, create Golden Tickets, or achieve full domain compromise. The techniques are particularly dangerous because they abuse legitimate Windows functionality, making them difficult to block without affecting normal operations. Detection of these specific RPC calls indicates an active adversary attempting to escalate privileges or move laterally through the network, requiring immediate incident response to prevent further compromise of critical infrastructure components.

Query

```
SELECT timestamp, orig_hostname, id.orig_h, id.resp_h, resp_hostname, domain,
username, endpoint, hostname, operation, sensor_uid

FROM network.dce_rpc

WHERE (

    (endpoint = 'unknown-c681d488-d850-11d0-8c52-00c04fd90f7e' AND operation
IN ('unknown-0', 'unknown-4', 'unknown-5', 'unknown-6', 'unknown-7', 'unknown-12',
'unknown-13', 'unknown-15')) OR

    (endpoint = 'spoolss' AND operation IN
('RpcRemoteFindFirstPrinterChangeNotificationEx')) OR

    (endpoint = 'unknown-a8e0653c-2744-4389-a61d-7373df8b2292' AND operation
IN ('unknown-8', 'unknown-9')) OR

    (endpoint = 'netdfs' AND operation IN ('NetrDfsAddStdRoot',
'NetrDfsRemoveStdRoot'))

    ) AND timestamp > date_add('day', -14, now())

ORDER BY timestamp DESC

LIMIT 100
```



5. Outgoing SSH On Non-Standard Port

What this query finds

This query identifies outbound SSH connections that are not using the standard SSH port 22. It detects systems within the network initiating SSH sessions to external or internal destinations on non-standard ports such as 2222, 443, 8080, or other atypical port numbers commonly used to disguise SSH traffic or bypass network security controls that may only monitor standard SSH communications.

Hunt Logic

The query examines iSessions metadata to identify outbound TCP connections with SSH protocol occurring on ports other than the default port 22. Systems that typically don't require SSH access or connections to unusual external destinations warrant particular attention in this analysis.

What's the Security Implication?

Outbound SSH on non-standard ports often indicates attempts to establish covert command and control channels, data exfiltration tunnels, or unauthorized remote access that bypasses standard network monitoring and firewall rules. Attackers frequently use SSH on alternate ports to blend their traffic with legitimate web traffic (such as using port 443) or to avoid detection by security tools that only monitor standard SSH port 22. This technique can facilitate persistent access to compromised systems, enable lateral movement through SSH tunneling, or serve as a conduit for data theft. The use of non-standard ports suggests deliberate evasion tactics and potential policy violations, as legitimate SSH connections typically use standard configurations unless specifically authorized. This activity requires investigation to determine whether it represents unauthorized access, data exfiltration, or the establishment of persistent backdoor access to the network infrastructure.

Query

```
SELECT timestamp, uid, id.orig_h, orig_hostname, id.resp_h, resp_hostname,
id.resp_p, proto_name, orig_ip_bytes, resp_ip_bytes, duration, conn_state, sensor_
uid, service

FROM network.isession

WHERE LOWER(service) = 'ssh' AND id.resp_p != 22 AND local_resp != true AND
timestamp > date_add('day', -14, now())

ORDER BY timestamp DESC

LIMIT 100
```

6. Multi-Country Sign-Ins (Impossible Travel Indicator)

What this query finds

Detects users who have signed in from multiple different countries within the last 24 hours, which may indicate compromised credentials.

Hunt Logic

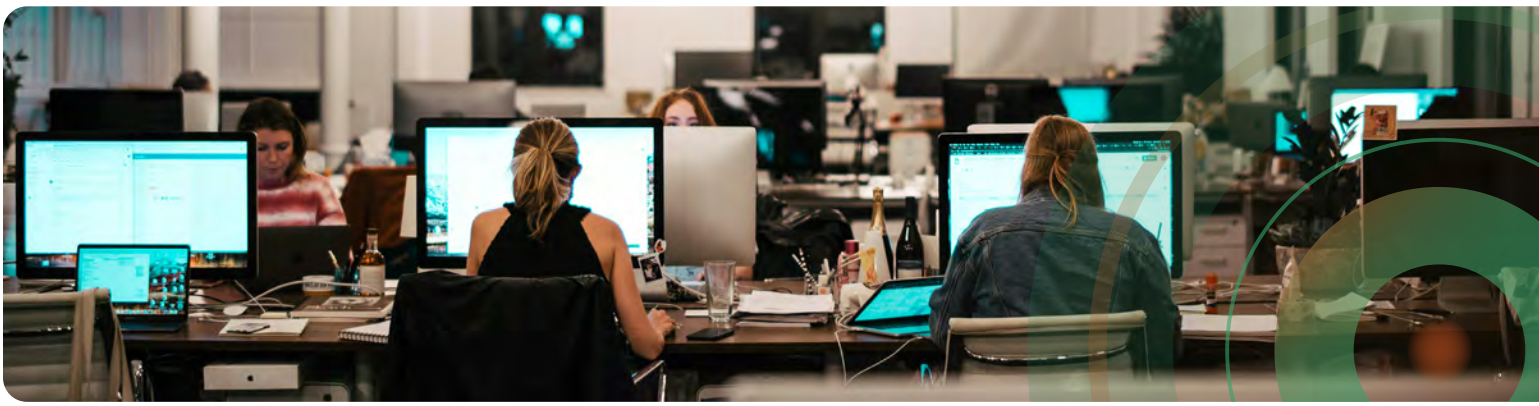
This query groups all successful sign-ins by user over the last 24 hours, counts how many different countries each user signed in from, and identifies users who appeared in more than one country. It focuses on successful logins only and filters out entries without location data.

What's the Security Implication?

If your query finds users signing in from multiple countries in 24 hours, that could mean credential compromise since rapid international travel is unlikely for most users.

Query

```
SELECT DISTINCT(vectra.identity_principal), COUNT(DISTINCT location.country_or_
region) AS CountryCount, MIN(timestamp) AS FirstLogin, MAX(timestamp) AS LastLogin
FROM entra.signins._all
WHERE location.country_or_region IS NOT NULL
  AND timestamp > date_add('day', -1, now())
GROUP BY vectra.identity_principal
HAVING COUNT(DISTINCT location.country_or_region) > 1
ORDER BY CountryCount
LIMIT 100
```



7. Failed Login Patterns by IP Address

What this query finds

Detects IP addresses with high numbers of failed authentication attempts, indicating potential brute force or password spray attacks.

Hunt Logic

This query aggregates all failed sign-in attempts by IP address over the last 6 hours, counts the total failures and unique users targeted from each IP. It identifies IPs with 20 or more failed attempts, which could indicate automated attack tools.

What's the Security Implication?

If your query finds IP addresses with many failed login attempts, that could mean active brute force or password spray attacks targeting your organization.

High unique user counts from single IPs often indicate password spray attacks across multiple accounts

Query

```
SELECT ip_address, COUNT(*) AS FailedAttempts,
  COUNT(DISTINCT vectra.identity_principal) AS UniqueUsers,
  MIN(timestamp) AS FirstFailure,
  MAX(timestamp) AS LastFailure
FROM entra.signins."Demolab-AD"
WHERE ip_address IS NOT NULL
  AND timestamp > date_add(hour, -6, now())
  AND status.error_code != 0
GROUP BY ip_address
HAVING COUNT(*) >= 20
ORDER BY FailedAttempts
LIMIT 100
```

8. Suspicious Storage Account Bulk Data Access

What this query finds

Detects potential data exfiltration by identifying users or applications performing unusually high volumes of blob/file downloads from storage accounts within a short timeframe.

Hunt Logic

This query analyzes Azure activity logs for storage account read operations over the last 6 hours, grouping by caller IP address and user identity. It counts successful blob and file read operations, identifying sources that performed 100 or more read operations which could indicate bulk data downloading.

What's the Security Implication?

If your query finds users or IP addresses with large number of storage blob/file read operations in a short period of time, that could mean data exfiltration attempts where attackers are bulk downloading sensitive files.

This pattern often indicates compromised accounts being used to steal large amounts of organizational data from cloud storage.

Query

```
SELECT vectra.identity,
  resourceid,
  COUNT(*) AS AccessCount,
  COUNT(DISTINCT ResourceId) AS UniqueStorageAccounts,
  MIN(timestamp) AS FirstAccess,
  MAX(timestamp) AS LastAccess
FROM azurecp.operations._all
WHERE timestamp > date_add(hour, -6, now())
  AND UPPER(operationname) IN ('MICROSOFT.STORAGE/STORAGEACCOUNTS/
BLOBSERVICES/CONTAINERS/BLOBS/READ', 'MICROSOFT.STORAGE/
STORAGEACCOUNTS/FILESERVICES/SHARES/FILES/READ')
  AND resulttype = 'Success'
GROUP BY vectra.identity, ResourceId
HAVING COUNT(*) >= 100
ORDER BY AccessCount
LIMIT 100
```

9. Excessive Key Vault Secret Access Patterns

What this query finds

Identifies suspicious Key Vault operations where secrets, keys, or certificates are being accessed in bulk or from unusual sources, potentially indicating credential harvesting attacks.

Hunt Logic

This query examines Azure activity logs for Key Vault access operations over the last 24 hours, focusing on successful secret, key, and certificate retrieval operations. It groups by caller information and identifies sources accessing 20+ secrets or 10+ unique secrets, which exceeds normal application behaviour.

What's the Security Implication?

If your query finds users or applications accessing high number of secrets or multiple unique secrets from Key Vault in one day, that could mean credential harvesting attacks where compromised identities are being used to steal sensitive authentication materials.

Attackers often target Key Vaults to obtain certificates, API keys, and connection strings needed for lateral movement and persistent access.

Query

```
SELECT calleripaddress,
  vectra.identity,
  resourceid,
  operationname,
  COUNT(*) AS SecretAccessCount,
  COUNT(DISTINCT resourceid) AS UniqueSecrets,
  MIN(timestamp) AS FirstAccess,
  MAX(timestamp) AS LastAccess
FROM azurecp.operations._all
WHERE timestamp > date_add('hour', -24, now())
  AND operationname IN ('SecretGet', 'KeyGet', 'CertificateGet')
  AND resulttype = 'Success'
  AND calleripaddress IS NOT NULL
GROUP BY calleripaddress, vectra.identity, resourceid, operationname
HAVING COUNT(*) >= 20 OR COUNT(DISTINCT resourceid) >= 10
ORDER BY SecretAccessCount
LIMIT 100
```

Compliance-Based Hunting

Compliance-based hunting targets violations of security policies, access controls, or architectural boundaries defined by regulatory or internal standards.

This approach surfaces risky behavior that may not be outright malicious but could signal misconfigurations, insider threats, or enforcement gaps.

Below are examples of common compliance-driven hunts.

1. Usage of SMBv1

What this query finds

This query identifies all instances of SMBv1 protocol usage across your network infrastructure, including client connections, server shares, and inter-system communications. It detects both active SMBv1 sessions and systems that have SMBv1 enabled but may not be actively using it. The query will reveal legacy applications, older Windows systems, network-attached storage devices, and third-party software that still rely on this outdated protocol.

Hunt Logic

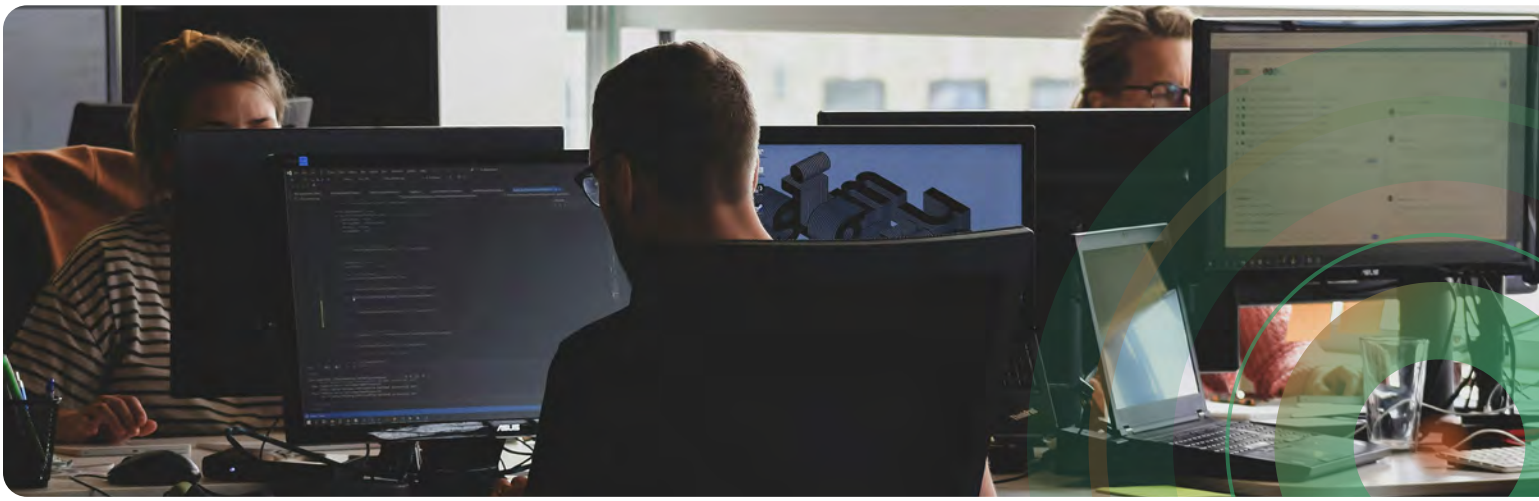
The query scans network SMB activity over the past 14 days, listing distinct source hosts using SMBv1

What's the Security Implication?

If your query finds SMBv1 protocol usage, that could mean your organization faces significant risks of lateral movement attacks, where cybercriminals can spread through networks once they gain initial access. Regulatory compliance is also at risk, since many standards require SMBv1 to be disabled. Beyond security concerns, SMBv1 also impacts network performance and reliability, making it a liability from both operational and regulatory perspectives.

Query

```
SELECT DISTINCT id.orig_h, orig_hostname.name
FROM network.smb_mapping._all
WHERE version = 'SMBv1' AND timestamp > date_add('day', -14, now())
```



2. Usage of HTTP CONNECT on uncommon TCP ports

What this query finds

This query detects HTTP CONNECT method requests targeting TCP ports outside the typical range of standard web services (such as ports other than 80, 443, 8080, or 8443), which is possible thanks to Vectra AI's deep packet service inspection that identifies HTTP traffic regardless of port number. It identifies potential tunneling attempts, proxy abuse, and unauthorized outbound connections that could indicate compromised systems attempting to establish backdoor communications. The query will surface suspicious network behavior including malware attempting to reach external servers, employees potentially circumventing security policies, or attackers using legitimate protocols to mask malicious activity. This helps uncover both targeted attacks and policy violations that traditional web filtering might miss.

Hunt Logic

The query searches the HTTP logs for proxied CONNECT methods that don't use standard HTTPS (port 443), displaying key details like source, destination, user agent, and more.

What's the Security Implication?

If your query finds HTTP CONNECT requests on uncommon TCP ports, this could indicate attempts to bypass network security controls, setting up covert channels, or exfiltrating data through non-standard ports—posing a significant risk to network security.

Query

```
SELECT timestamp, id.orig_h, orig_hostname, user_agent, id.resp_h, resp_hostname,
id.resp_p, method, host, uri, status_code
FROM network.http
WHERE is_proxied = true AND LOWER(method) = 'connect' AND uri NOT LIKE '%:443'
AND timestamp > date_add('day', -14, now())
ORDER BY timestamp DESC
LIMIT 100
```

3. Out-of-Date Browser Detection

What this query finds

Detects HTTP user-agent strings indicating usage of outdated web browsers (e.g. old versions of Chrome, Firefox, Internet Explorer, Safari). Persistent use of such browsers across multiple days may highlight systems that are unpatched and at higher risk of exploitation.

Hunt Logic

Detects HTTP user-agent strings indicating usage of outdated web browsers (e.g. old versions of Chrome, Firefox, Internet Explorer, Safari). Persistent use of such browsers across multiple days may highlight systems that are unpatched and at higher risk of exploitation.

What's the Security Implication?

Outdated browsers often contain unpatched vulnerabilities that are widely documented and easily exploited through drive-by downloads, phishing, or malicious web content. If left unchecked, this creates a significant attack surface and may represent a compliance gap against endpoint hardening policies. Detecting and remediating outdated browser usage is critical for minimizing exploit risk and maintaining a secure browsing environment.

Query

```
SELECT id.orig_h, orig_hostname.name AS hostname, user_agent, COUNT(*) AS request_count
FROM network.http._all
WHERE timestamp BETWEEN date_add('day', -14, now()) AND now()
AND ( user_agent LIKE '%MSIE%'
OR user_agent LIKE '%Firefox/[3-6]%'
OR user_agent LIKE '%Chrome/[1-9]%'
OR user_agent LIKE '%Chrome/[1-4][0-9]%'
OR user_agent LIKE '%Version/(1[0-6](\.\d+)*)\s+Safari%' )
GROUP BY id.orig_h, orig_hostname.name, user_agent
ORDER BY request_count DESC
LIMIT 100
```

4. AI Service Usage - Interactions with Generative AI Platforms

What this query finds

Detects internal systems or users interacting with ChatGPT-related domains such as chat.openai.com or api.openai.com, which may signal unmonitored AI usage across the organization.

Hunt Logic

This query searches HTTP metadata for outbound traffic to known OpenAI domains. It surfaces the hosts, user agents, and volume of requests - helping to identify systems that are using generative AI services either frequently or unexpectedly.

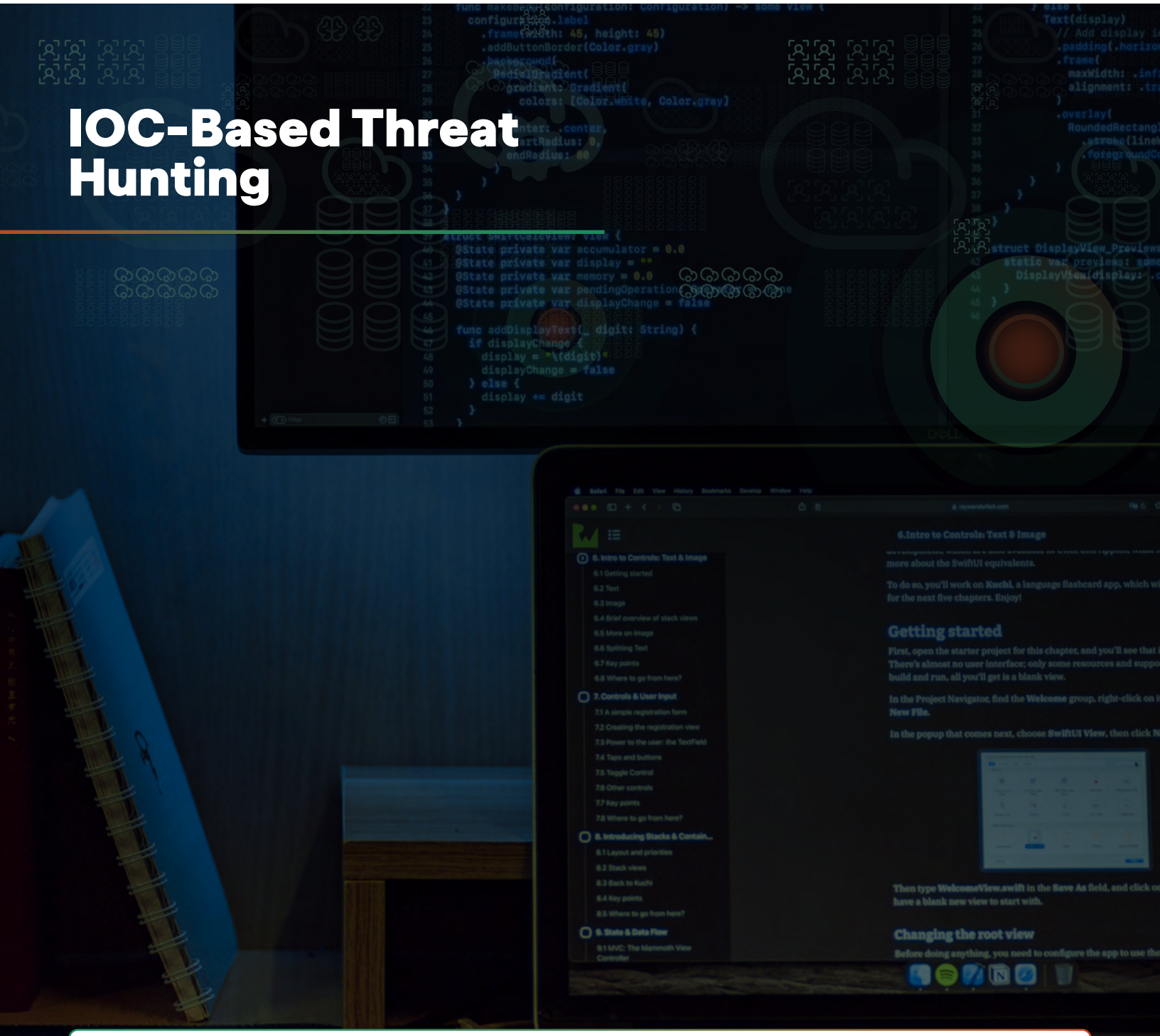
What's the Security Implication?

AI tools like ChatGPT can introduce data exposure risks if users paste sensitive code, credentials, or internal documents into prompts. In regulated environments, unauthorized use may also violate compliance mandates. Additionally, unsanctioned AI adoption, also known as shadow IT, can bypass established security controls, making it harder to govern how sensitive data is handled.

Query

```
SELECT query,  COUNT(DISTINCT orig_hostname.name) as unique_host_count,
COUNT(*) as total_query_count
FROM network.dns
WHERE ( query LIKE '%openai.com'
OR query LIKE '%chat.openai.com'
OR query LIKE '%anthropic.com'
OR query LIKE '%claude.ai'
OR query LIKE '%bard.google.com'
OR query LIKE '%bing.com
OR query LIKE '%cohere.ai'
OR query LIKE '%huggingface.co'
OR query LIKE '%mistral.ai'
OR query LIKE '%deepseek.com%' )
AND timestamp BETWEEN date_add('day', -14, now()) AND now()
GROUP BY query
ORDER BY unique_host_count DESC, total_query_count DESC
LIMIT 100
```

IOC-Based Threat Hunting



IOC-based threat hunting is a fast and effective way to validate potential exposures, uncover attacker infrastructure reuse, and catch threats that may have bypassed initial defenses.

It's especially useful for responding to threat intelligence alerts or public disclosures of active campaigns.

What Are IOCs?

IOCs can include:

- Suspicious domains (e.g. attacker-controlled C2 infrastructure)
- IP addresses associated with malware distribution or exfiltration
- File hashes of known malware samples
- Email addresses or user agents used in phishing or impersonation
- OAuth tokens, API keys, registry keys, or scripts used in persistence mechanisms

These artifacts don't typically generate detections on their own but can indicate compromise when placed in context.

Where Do You Get IOCs?

IOCs are often published by security researchers, vendors, and threat intelligence feeds. Common sources include:

- Threat intelligence reports (e.g. Mandiant, Volexity, CISA, Vectra Threat Research)
- Security news and blogs (e.g. BleepingComputer, KrebsOnSecurity)
- Threat sharing communities (e.g. ISACs, GitHub, AlienVault OTX)
- Social media (e.g. X/Twitter posts by trusted threat hunters)
- Detection engineering platforms (e.g. Sigma rules, YARA repos)

For example, after a report on a new nation-state campaign is released, you may see:

hash: 3f28f99e14b... associated with a credential dumping tool domain: login-auth-service[.]com used in phishing IP: 185.112.84.45 associated with command and control.

How to Hunt for IoCs with Vectra AI

Once you have an IOC or a set of them, use the Vectra AI Platform to check your environment:

1. Search for the domain, IP, or hash across historical metadata and detections.
2. Pivot on impacted entities - which host or user interacted with the IOC?
3. Investigate timelines - when was it first seen? Has it reappeared?
4. Check surrounding activity - look for follow-on behaviors such as privilege escalation, lateral movement, or data access patterns.

Tip

Even if an IOC is stale or generic, it may still help surface long-dwelling threats or confirm containment after an incident.

Example Workflow

Let's say CISA publishes a report with indicators tied to a known APT campaign:

1. You extract the IOCs: 2 domains, 1 IP, and a PowerShell hash.
2. In the Vectra AI Platform, you search for the domain in HTTP metadata and spot activity 3 weeks ago.
3. The destination host is a finance system - you pivot into Investigate and see related detections: "Abnormal Scripting Activity" and "Suspicious Lateral Movement."
4. You now have high-confidence evidence to escalate and investigate further.

Below are additional query examples to help you hunt for IOCs.

1. Malicious Domains – Command and Control / Phishing Infrastructure

What this query finds

Outbound sessions to known-bad domains—potential signs of C2 callbacks or beaconing.

Hunt Logic

The query searches for network sessions in the last 14 days where the destination domain matches a list of known-bad domains and displays detailed connection info.

What's the Security Implication?

If your query finds connections to attacker-controlled domains, this could indicate C2 communication, payload download, or user interaction with phishing lures. This is commonly seen in infostealers (e.g. LummaC2) or initial access brokers like Scattered Spider. If these are not remediated, attackers may maintain long-term access, exfiltrate sensitive data, or launch additional stages of their attack without interruption.

Query

```
SELECT timestamp, uid, id.orig_h as “id_orig_h”, orig_hostname, id.resp_h as “id_
resp_h”,
        resp_hostname, id.resp_p as “id_resp_p”, proto_name, orig_ip_bytes,
        resp_ip_bytes, duration, conn_state, sensor_uid
FROM network.isession
WHERE (resp_domain = ‘baddomain1.com’ OR resp_domain = ‘baddomain2.com’)
AND timestamp > date_add(‘day’, -14, now())
ORDER BY timestamp DESC
LIMIT 100
```

2. Known Malicious IPs – Infrastructure Reuse or Exfiltration

What this query finds

Recent connections to or from blacklisted IP addresses—possible evidence of malware activity or tunneling.

Hunt Logic

This query searches for all network sessions from the past 14 days that involve specific IP addresses (as either source or destination). It helps you track all interactions with the listed IPs, providing full session details like hosts, protocols, and duration.

What's the Security Implication?

IP-based IOCs are often reused across campaigns. Hits here may suggest malware callback, direct data exfiltration, or an ongoing compromise.

Query

```
SELECT timestamp, uid, id.orig_h as "id_orig_h", orig_hostname, id.resp_h as "id_
resp_h", resp_hostname, id.resp_p as "id_resp_p", proto_name, orig_ip_bytes, resp_
ip_bytes, duration, conn_state, sensor_uid

FROM network.isession

WHERE (id.resp_h IN ('192.0.2.1', '192.0.2.2') OR id.orig_h IN ('192.0.2.1', '192.0.2.2'))
AND timestamp > date_add('day', -14, now())

ORDER BY timestamp DESC

LIMIT 100
```

3. Suspicious File Names – Malicious Payload Staging

What this query finds

Transferred files with long, suspicious names that could indicate malicious payload staging or lateral movement tools.

Hunt Logic

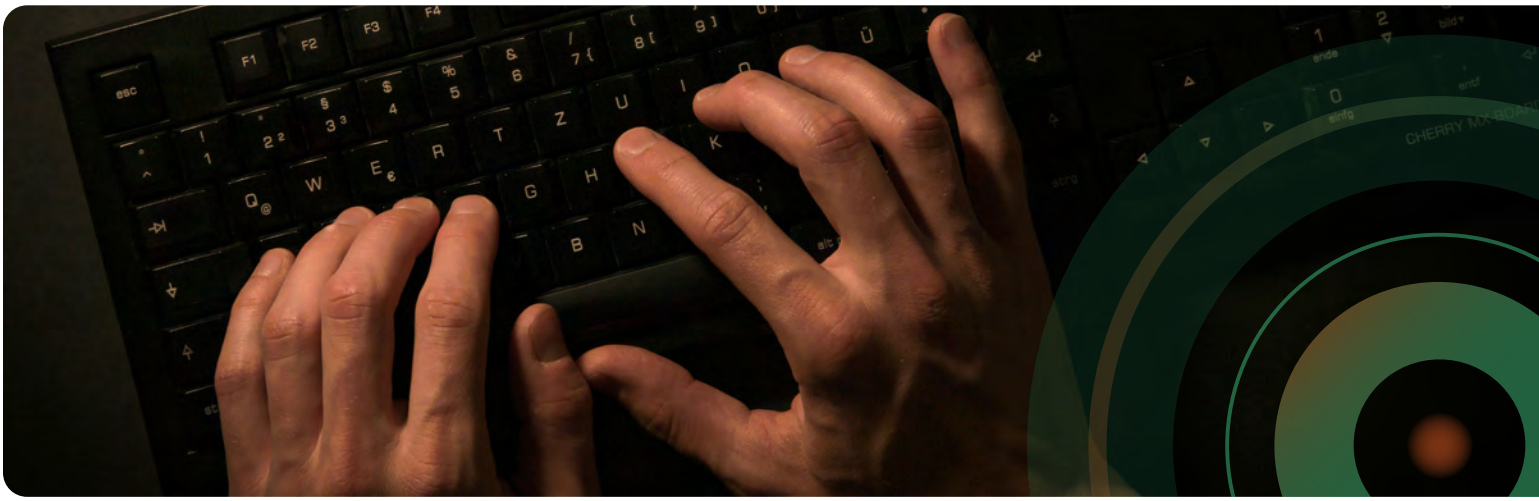
This query scans SMB file activity from the past 14 days, pulling out files with unusually long base filenames (over 50 characters). It extracts and lists these lengthy filenames, which can be a sign of malware or data obfuscation techniques.

What's the Security Implication?

Obfuscated or excessively long file names may signal automated malware drops or script-based delivery—often seen in ransomware and loader campaigns.

Query

```
SELECT name, REGEXP_EXTRACT(name, '([^\\/]+)$') AS base_filename,
LENGTH(REGEXP_EXTRACT(name, '([^\\/]+)$')) AS base_length

FROM network.smb_files

WHERE LENGTH(REGEXP_EXTRACT(name, '([^\\/]+)$')) > 50 AND timestamp > date_
add('day', -14, now())

ORDER BY base_length DESC

LIMIT 50
```



4. Files Without Vowels – Obfuscated or Auto-Generated Malware

What this query finds

File names with no vowels—potential sign of obfuscation or automated tooling.

Hunt Logic

This query looks at SMB file activity from the last 14 days and extracts filenames that contain zero vowels (just consonants or symbols) in their base name. These kinds of filenames can indicate suspicious activity, like malware using obfuscated or generated file names to avoid detection.

What's the Security Implication?

Lack of vowels may indicate programmatically generated files—commonly used by malware builders to evade detection and signature-based tools. Failing to detect these can allow malicious payloads to persist undetected, increasing dwell time and attack impact.

Query

```
SELECT name, REGEXP_EXTRACT(name, r'([^\\/]+)$') AS base_filename
FROM network.smb_files
WHERE REGEXP_LIKE(REGEXP_EXTRACT(name, r'([^\\/]+)$'), '^[^aeiouAEIOU]+$') AND
timestamp > date_add('day', -14, now())
LIMIT 50
```

This search uses a regular expression to extract the filename, with the following logic

- [and] define a character set. Inside, \\/ means both slash / and backslash \. So [^\\/] means "any character except / or \".
- + means "one or more" of those non-slash or non-backslash characters.
- Parentheses () create a capture group that extracts the part of the string matching the pattern.
- $ asserts that this match is at the very end of the string.

5. Suspicious File Paths – Lateral Movement or Staging

What this query finds

Files accessed within suspicious file paths

Hunt Logic

This query searches SMB file activity from the last 14 days for files accessed or modified in /App/Data/Roaming/ paths. It surfaces details about these file operations, which can help spot suspicious access to sensitive or user profile data. You can replace this with other file paths you know are concerning in your organization to monitor for access.

What's the Security Implication?

Malware and tools often drop into less monitored folders like Roaming/ AppData. If not monitored, malware and tooling can operate in these directories indefinitely, giving attackers a quiet foothold.

Query

```
SELECT timestamp, uid, id.orig_h as "id_orig_h", orig_hostname, id.resp_h as "id_resp_h", resp_hostname, id.resp_p as "id_resp_p", version, path, action, name, sensor_uid
FROM network.smb_files
WHERE path LIKE '%/App/Data/Roaming/%' AND timestamp > date_add('day', -14, now())
ORDER BY timestamp DESC
LIMIT 100
```

6. JA3 Fingerprint – TOR or Unusual TLS Clients

What this query finds

Identifies TLS connections using the JA3 fingerprint linked to TOR clients, revealing systems using TOR browsers or TOR-based apps in your environment.

Hunt Logic

The query pulls SSL/TLS session logs from the last 14 days, filtering for a specific JA3 hash known to match TOR clients. It lists details about each connection, including server and client info.

What's the Security Implication?

If your query finds these connections, it could mean data exfiltration attempts, unauthorized remote access, malware communication, penetration testing tools usage, policy violations where employees may be using anonymity software or automated attack frameworks that don't use standard browser TLS implementations. Detecting these fingerprints helps organizations identify potential security incidents that traditional monitoring might miss, as attackers increasingly rely on encrypted communications to avoid detection.

Query

```
SELECT timestamp, uid, id.orig_h as "id_orig_h", orig_hostname, id.resp_h as "id_resp_h", resp_hostname, id.resp_p as "id_resp_p", server_name, client_version, next_protocol, cipher, ja3, established, sensor_uid
FROM network.ssl
WHERE ja3 = 'e7d705a3286e19ea42f587b344ee6865' AND timestamp > date_add('day', -14, now())
ORDER BY timestamp DESC
LIMIT 100
```

You can read more on Ja3, on the github profile https://github.com/salesforce/ja3

And explore community-curated JA3 fingerprints at: https://ja3.zone/

Conclusion

Threat hunting is a powerful complement to the Vectra AI Platform's behavior-based detections, providing an extra layer of proactive defense that helps security teams reduce risk, uncover blind spots, and proactively resolve threats.

Use the examples in this guide to build repeatable workflows, save your queries, and operationalize hunting in your day-to-day practice. Leverage expert-built searches in the Vectra AI Platform, and explore the latest hunting queries on our GitHub page. our environment.

Watch Vectra AI Investigations in Action

About Vectra AI

Vectra AI is the leader in hybrid attack detection, investigation and response. The Vectra AI Platform delivers integrated signal across public cloud, SaaS, identity, and data center networks in a single platform. Vectra AI's patented Attack Signal Intelligence empowers security teams to rapidly detect, prioritize, investigate and stop the most advanced hybrid cyber-attacks. With 35 patents in AI-driven detection and the most vendor references in MITRE D3FEND, organizations worldwide rely on the Vectra AI Platform and MXDR services to move at the speed and scale of hybrid attackers. For more information, visit www.vectra.ai.

For more information please contact us:
Email: info@vectra.ai | vectra.ai