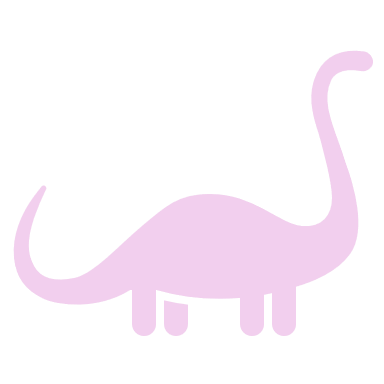
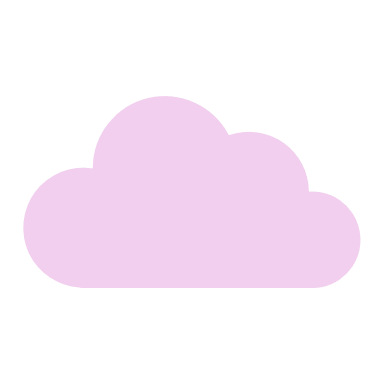
*Kadence’s Queries *

**

**Key**

Yellow = tweak it

Purple = good

cyan = mixed

green = fixed

gray = new query

***note – a lot of the queries’ comments are AI and not my own words unless it is typed in all lowercase. also this isn’t formal documentation but rather just notes***

**------------------------------------------------------------------------------------------------------------------**

Abnormal ssl cert usage – fix it   
**New Query:**  
DeviceNetworkEvents

| where Protocol == "TLS" // Filtering for SSL/TLS protocol

| where ActionType == "ConnectionSuccess" // Filter successful connections

| extend RemoteIP = tostring(RemoteIP) // Extracting Remote IP

| extend RemotePort = tostring(RemotePort) // Extracting Remote Port

| extend RemoteUrl = tostring(RemoteUrl) // Extracting Remote URL

| summarize Count = count() by RemoteIP, RemotePort, RemoteUrl

| where RemotePort == "443" // Focus on common HTTPS traffic

| where Count > 10 // Identify unusually frequent connections, adjust the threshold as necessary

| order by Count desc

*- filters by transport layer security protocol (encryption), only looking for successful connections, I think then we convert things to a string so we can view it nicely, port 443 to see ssl interactions, threshold 10 for now*Abuse of gpo objects – 4 alerts

Abuse remote management – 0 alerts but I think it needs to be tweaked anyway

Anomalous device connection in network – can’t seem to get a working query that doesn’t spit out 10k results but here’s a baseline  
**New Query:**DeviceNetworkEvents

| where Timestamp > ago(7d) // Only include records from the last 7 days

| where ActionType == "ConnectionSuccess" // Focus only on successful network connections

| where isnotempty(RemoteIP) and isnotempty(RemotePort) // Ensure there is a valid RemoteIP and RemotePort for each event

| where RemoteIPType == "Public" // Focus only on external (public) IP addresses

| where RemotePort !in (80, 443, 53, 3389, 22) // Exclude common ports like HTTP (80), HTTPS (443), DNS (53), RDP (3389), and SSH (22)

| summarize connection\_count = count(), // Count how many times each unique connection occurs

first\_seen = min(Timestamp), // Track the earliest time the connection was seen

last\_seen = max(Timestamp) // Track the most recent time the connection was seen

by DeviceId, DeviceName, RemoteIP, RemotePort, RemoteUrl, Protocol, // Group by device, remote IP, remote port, etc.

InitiatingProcessFileName, InitiatingProcessFolderPath, // Include process details (file name and folder path)

InitiatingProcessSHA256, InitiatingProcessCommandLine // Include hash values and command line for further investigation

| order by connection\_count asc, last\_seen desc // Sort results to prioritize rarest connections first (based on count) and then by most recent

Anomalous file deletion pattern- seems wrong  
**New Query:**

DeviceLogonEvents

| where ActionType == "FileDeleted" // Filter for file deletion events

| extend FileName = tostring(InitiatingProcessFileName),

UserName = tostring(InitiatingProcessAccountName),

Time = TimeGenerated

| summarize DeletionCount = count() by FileName, UserName, bin(Time, 1h) // Summarize deletions by file name, user, and time window (1-hour)

| where DeletionCount > 5 // Detect more than 5 deletions in an hour (adjust threshold as needed)

| order by DeletionCount desc

Beaconing traffic detection

Browser exploit detection  
**New Query:**  
DeviceNetworkEvents

| where InitiatingProcessFileName in ("chrome.exe", "msedge.exe", "firefox.exe") // Focus on browser processes

| extend CommandLine = tostring(InitiatingProcessCommandLine),

         ParentFileName = tostring(InitiatingProcessParentFileName),

         RemoteIP = RemoteIP,

         RemotePort = RemotePort,

         RemoteUrl = tostring(RemoteUrl),

         UserName = InitiatingProcessAccountName,

         Time = TimeGenerated

| where RemoteUrl contains "http" or RemoteIP in ("malicious\_ip\_1", "malicious\_ip\_2") // Look for HTTP traffic or suspicious IPs

| where RemotePort != 80 and RemotePort != 443 // Exclude common ports for HTTP/HTTPS

| where ParentFileName != "explorer.exe" and ParentFileName != "winlogon.exe" // Exclude legitimate parent processes

| summarize ConnectionCount = count() by RemoteUrl, RemoteIP, RemotePort, ParentFileName, FileName = InitiatingProcessFileName, UserName, bin(Time, 1h)

| where ConnectionCount > 5 // Trigger alert if more than 5 connections in an hour

| order by ConnectionCount desc

Cloud storage upload spikes

Default or weak creds – seems like it needs refining  
**New Query:**  
DeviceLogonEvents

| where AccountName in ("admin", "administrator", "guest", "root", "test", "user") // Common default account names

    or (FailureReason != "" and FailureReason != "Success") // Failed login attempts indicating weak credentials

| extend LogonAttemptTime = TimeGenerated,

         LogonType = tostring(LogonType),

         AccountName = tostring(AccountName),

         RemoteIP = RemoteIP,

         DeviceName = DeviceName,

         FailureReason = tostring(FailureReason)

| summarize FailedAttempts = count() by AccountName, DeviceName, RemoteIP, bin(LogonAttemptTime, 1h)

| where FailedAttempts > 5 // More than 5 failed login attempts in the last hour indicates suspicious behavior

| order by FailedAttempts desc

Detection of credential dumping

Dns query for known malicious domains - awful

Email attachment with exec. files

Endpoint beaconing to non standard point – why is there 30k entries.

DeviceNetworkEvents

| where RemoteIP !in ('<trusted\_IP\_list>') // Exclude known trusted IP addresses

| where RemotePort !in (80, 443, 21, 25) // Exclude common ports (HTTP, HTTPS, FTP, SMTP)

| where Protocol in ('TCP', 'UDP') // Only focus on TCP/UDP protocols

| where ActionType in ('ConnectionAttempt', 'SuccessfulConnection') // Look for connection attempts or successful connections

| where Timestamp > ago(24h) // Adjust the time window for the last 24 hours (can change to any time frame)

| project Timestamp, DeviceId, DeviceName, ActionType, RemoteIP, RemotePort, RemoteUrl, LocalIP, LocalPort, Protocol, InitiatingProcessSHA1, InitiatingProcessSHA256, InitiatingProcessMD5, InitiatingProcessFileName, InitiatingProcessFileSize, InitiatingProcessVersionInfoCompanyName, InitiatingProcessVersionInfoProductName, InitiatingProcessVersionInfoProductVersion, InitiatingProcessId, InitiatingProcessCommandLine, InitiatingProcessCreationTime, InitiatingProcessFolderPath, InitiatingProcessParentFileName, InitiatingProcessParentId, InitiatingProcessParentCreationTime, InitiatingProcessAccountDomain, InitiatingProcessAccountName, InitiatingProcessAccountSid, InitiatingProcessAccountUpn, InitiatingProcessAccountObjectId, InitiatingProcessIntegrityLevel, InitiatingProcessTokenElevation, ReportId, AppGuardContainerId, AdditionalFields, InitiatingProcessSessionId, IsInitiatingProcessRemoteSession, InitiatingProcessRemoteSessionDeviceName, InitiatingProcessRemoteSessionIP, InitiatingProcessUniqueId, TenantId, Type, SourceSystem, MachineGroup, TimeGenerated

| order by Timestamp desc // Order results by the most recent event

Excessive admin privilege assignments

Excessive dns requests – doesn’t work

DeviceNetworkEvents

| where Protocol == "UDP" // DNS usually runs over UDP

| where RemotePort == 53 // Standard DNS port

| where ActionType == "ConnectionAttempt" // Focus on connection attempts (DNS requests)

| where Timestamp > ago(1h) // Focus on the last 1 hour (adjust as needed)

| summarize DNSRequestCount = count() by DeviceId, RemoteIP, bin(Timestamp, 5m) // Count DNS requests per device/IP every 5 minutes

| where DNSRequestCount > 100 // Set a threshold for excessive requests (adjust as needed)

| order by DNSRequestCount desc // Order by most excessive DNS request counts

| project Timestamp, DeviceId, RemoteIP, DNSRequestCount

Excessive file access

Exploit kit – seems kinda stupid

DeviceNetworkEvents

| where (RemoteIP in ('<malicious\_ip\_list>')) // Known malicious IP addresses associated with exploit kits

or (RemoteUrl has\_any ('exploit', 'payload', 'malicious')) // Filter based on known exploit kit URLs or malicious patterns in the URLs

| where Protocol == "TCP" // Typically exploit kits use TCP for payload delivery

| where RemotePort !in (80, 443) // Exclude standard HTTP/HTTPS ports

| where ActionType in ('ConnectionAttempt', 'SuccessfulConnection') // Look for connection attempts or successful connections

| where Timestamp > ago(1h) // Look at the past hour (adjust based on your needs)

| summarize ExploitKitConnectionCount = count() by DeviceId, RemoteIP, RemoteUrl, bin(Timestamp, 5m) // Count the number of connections per device/IP/URL every 5 minutes

| where ExploitKitConnectionCount > 5 // Set a threshold for frequent connections, indicative of exploit kit activity

| order by ExploitKitConnectionCount desc // Order by the highest connection counts

| project Timestamp, DeviceId, RemoteIP, RemoteUrl, ExploitKitConnectionCount

Mimikatz

Multiple failed logon attempts – seems off  
**New Query:**  
DeviceLogonEvents

| where ActionType == "LogonFailed" // Focus on failed logon attempts

| where Timestamp > ago(1h) // Look at the past 1 hour (adjust the time window as needed)

| summarize FailedAttempts = count() by DeviceId, AccountName, bin(Timestamp, 5m) // Count failed attempts by Device and Account every 5 minutes

| where FailedAttempts > 5 // Threshold for multiple failed attempts (can be adjusted based on your environment)

| order by FailedAttempts desc // Order by the number of failed attempts

| project Timestamp, DeviceId, AccountName, FailedAttempts

Rdp anomalies – doesn’t work  
**New Query:**  
DeviceEvents

| where RemotePort == 3389 // RDP uses port 3389 by default

| where Timestamp > ago(1h) // Analyze the last hour (adjust based on your needs)

| summarize RDP\_ConnectionCount = count() by DeviceId, AccountName, RemoteIP, bin(Timestamp, 5m) // Count RDP connections every 5 minutes

| where RDP\_ConnectionCount > 3 // More than 3 RDP connections in 5 minutes could indicate suspicious activity

| join kind=inner (

DeviceLogonEvents

| where ActionType == "LogonFailed" // Focus on failed logons

| where Timestamp > ago(1h) // Look at the past hour

| summarize FailedAttempts = count() by DeviceId, AccountName, RemoteIP, bin(Timestamp, 5m)

| where FailedAttempts > 3 // More than 3 failed logon attempts

) on DeviceId

| project Timestamp, DeviceId, AccountName, RemoteIP, RDP\_ConnectionCount, FailedAttempts

| order by RDP\_ConnectionCount desc, FailedAttempts desc

Shadow it application usage – I don’t really know what this is & it doesn’t work

Smb lateral movement – I think its ok?

Suspicious rar or zip file –

Suspicious archive file extraction –

Suspicious command exec in cli – could be made more specific   
**New Query:**

DeviceProcessEvents

| where Timestamp > ago(24h) // Limit to the past 24 hours

| where ProcessCommandLine != "" // Only consider entries with a command line

| where ProcessCommandLine has\_any ('powershell', 'cmd', 'net user', 'netstat', 'curl', 'wget', 'ipconfig', 'nc', 'nmap', 'whoami') // Look for common system or network-related commands

| where ProcessIntegrityLevel == "High" // Look for processes with high privileges (admin or system level)

| summarize Count = count() by DeviceId, DeviceName, AccountName, ProcessCommandLine, bin(Timestamp, 1h)

| where Count > 1 // Flag when the same command is executed multiple times within an hour

| project Timestamp, DeviceId, DeviceName, AccountName, ProcessCommandLine, Count

| order by Count desc

Suspicious dhcp activity – I feel like requests and acknowledges are not suspicious.

Suspicious file modifications

Suspicious firmware update

Suspicious script exec in browser – seems broad, still iffy bout this one  
**New Query:**

DeviceProcessEvents

| where FileName in ("chrome.exe", "firefox.exe", "iexplore.exe", "msedge.exe") // Target common browsers

| where ProcessCommandLine contains "script" or ProcessCommandLine contains ".vbs" or ProcessCommandLine contains ".js" or ProcessCommandLine contains ".ps1" // Look for scripts in command lines

| where Timestamp > ago(1d) // Filter to recent events (adjust time window as needed)

| project Timestamp, DeviceId, DeviceName, FileName, ProcessCommandLine, AccountName, AccountDomain, ProcessId, ActionType

| extend IsSuspicious = case(

ProcessCommandLine contains "powershell", "Potential PowerShell Execution",

ProcessCommandLine contains "cmd", "Potential Command Execution",

ProcessCommandLine contains "base64", "Potential Encoded Command",

ProcessCommandLine contains ".vbs" or ProcessCommandLine contains ".js", "Potential Script File Execution",

ProcessCommandLine contains ".ps1", "Potential PowerShell Script Execution",

"Normal")

| where IsSuspicious != "Normal" // Filter out normal activities and keep suspicious activities

| order by Timestamp desc

Suspicious task scheduling

System crash dump access

Tor detection – fix this because it counts the word ‘storage’  
**New Query:**  
let tor\_exit\_node\_ips = dynamic([

"185.220.100.0", "185.220.101.0", "185.220.102.0", "185.220.103.0", // Example Tor exit node IP ranges

"162.255.0.0", "162.255.1.0", "162.255.2.0", "162.255.3.0"

// Add more known Tor exit IP ranges or dynamically update this list

]);

DeviceNetworkEvents

| where RemoteIP in (tor\_exit\_node\_ips) // Check for connections to known Tor exit node IPs

or RemotePort in (9050, 9150) // Check for Tor's default communication ports

or Protocol == "TCP" // Tor traffic typically uses TCP

| where Timestamp > ago(24h) // Look at recent events (e.g., last 24 hours)

| project Timestamp, DeviceId, DeviceName, RemoteIP, RemotePort, Protocol, ActionType, InitiatingProcessFileName, InitiatingProcessCommandLine

| order by Timestamp desc // Order by most recent activity

Unauthorized access to backup files – seems iffy  
**New Query:**  
let backupFileExtensions = dynamic([".bak", ".zip", ".tar", ".sql", ".tar.gz", ".bakx", ".db"]);

let unauthorizedAccountNames = dynamic(["guest", "administrator", "unknown"]); // Add known unauthorized accounts

DeviceProcessEvents

| where FolderPath endswith ".bak" or FolderPath endswith ".zip" or FolderPath endswith ".tar"

or FolderPath endswith ".sql" or FolderPath endswith ".tar.gz" or FolderPath endswith ".bakx" or FolderPath endswith ".db"

| where AccountName in (unauthorizedAccountNames) // Only look for unauthorized access by certain account names

| where ActionType == "Read" or ActionType == "Write" // Detecting file access actions

| where Timestamp > ago(24h) // Look at recent events (last 24 hours)

| project Timestamp, DeviceId, DeviceName, AccountName, FileName, FolderPath, ActionType, ProcessCommandLine, InitiatingProcessFileName, ProcessId, ProcessCreationTime, AccountDomain

| order by Timestamp desc // Sort by most recent

unauthorized access to financial data – this is very broad, I do not know what financial data looks like

**New Query:**  
let financialFileExtensions = dynamic([".xlsx", ".xls", ".csv", ".xml", ".db", ".sql", ".json", ".pdf"]);

let unauthorizedAccountNames = dynamic(["guest", "administrator", "unknown", "service\_account"]); // Add known unauthorized accounts

DeviceProcessEvents

| where FolderPath endswith ".xlsx" or FolderPath endswith ".xls" or FolderPath endswith ".csv"

or FolderPath endswith ".xml" or FolderPath endswith ".db" or FolderPath endswith ".sql"

or FolderPath endswith ".json" or FolderPath endswith ".pdf"

| where AccountName in (unauthorizedAccountNames) // Only look for unauthorized access by certain account names

| where ActionType == "Read" or ActionType == "Write" // Detecting file access actions

| where Timestamp > ago(24h) // Look at recent events (last 24 hours)

| project Timestamp, DeviceId, DeviceName, AccountName, FileName, FolderPath, ActionType, ProcessCommandLine, InitiatingProcessFileName, ProcessId, ProcessCreationTime, AccountDomain

| order by Timestamp desc // Sort by most recent

unauthorized api call – 30k results  
**New Query:**  
DeviceEvents

| where ActionType == "ApiCallFailed"

| where Timestamp > ago(24h)

| extend EventDetails = parse\_json(AdditionalFields)

| project Timestamp, DeviceName, AccountName, RemoteIP, EventDetails.ApiName, EventDetails.ErrorCode, EventDetails.ErrorDescription

| order by Timestamp desc

unauthorized configuration change – I think this one is ok. I think its logging when help desk logs into certain machines.

Unauthorized network share access – stupid and bad 30k results  
**New Query:**  
DeviceEvents

| where ActionType == "NetworkShareAccessDenied"

| where Timestamp > ago(24h)

| extend EventDetails = parse\_json(AdditionalFields)

| project Timestamp, DeviceName, AccountName, RemoteIP, EventDetails.ShareName, EventDetails.ErrorCode, EventDetails.ErrorDescription

| order by Timestamp desc

Unusual binary execution from temp – I think this one is ok

Unusual process parent child relation – this may or may not work. Counseling required  
**New Query:**DeviceProcessEvents

| where Timestamp > ago(24h)

| where isnotempty(InitiatingProcessFileName) and isnotempty(FileName) // Exclude empty values

| extend ParentProcessLower = tolower(InitiatingProcessFileName), ChildProcessLower = tolower(FileName)

| where

(ParentProcessLower in ("cmd.exe", "powershell.exe", "wscript.exe", "cscript.exe", "mshta.exe", "rundll32.exe")

and ChildProcessLower !in ("conhost.exe", "cmd.exe", "powershell.exe", "wscript.exe", "cscript.exe"))

or

(ParentProcessLower in ("winword.exe", "excel.exe", "powerpnt.exe", "outlook.exe")

and ChildProcessLower in ("cmd.exe", "powershell.exe", "wscript.exe", "cscript.exe", "mshta.exe", "rundll32.exe"))

or

(ParentProcessLower in ("chrome.exe", "firefox.exe", "iexplore.exe", "msedge.exe")

and ChildProcessLower in ("cmd.exe", "powershell.exe", "wscript.exe", "cscript.exe", "mshta.exe", "rundll32.exe"))

or

(ParentProcessLower == "explorer.exe"

and ChildProcessLower in ("cmd.exe", "powershell.exe", "wscript.exe", "cscript.exe", "mshta.exe", "rundll32.exe"))

| where ProcessCommandLine !contains "safe" // Exclude known safe executions (customize as needed)

| project Timestamp, DeviceName, AccountName, ParentProcessLower, ChildProcessLower,

ProcessCommandLine, InitiatingProcessCommandLine, InitiatingProcessFolderPath, FolderPath

| order by Timestamp desc

| limit 500 // Restrict to the top 500 results for clarity

Unusual rdp or ssh logon attempt