Production-Grade KQL Query Library

Microsoft Sentinel · Microsoft Defender XDR (Advanced Hunting) · Azure Monitor / Log Analytics

50 verified detection & hunting queries — schema validated against Microsoft documentation (June 2026)

How to use this library

- **Schema accuracy:** Every table and column used is documented by Microsoft. No tables, columns, EventIDs, or ActionType values are invented.
- **ActionType discipline:** Microsoft keeps the full ActionType enumeration in the in-portal schema reference, not the public web docs. Where a value is firmly documented it is used directly; otherwise detection is driven from documented command-line / file fields.
- **Timestamp columns:** Defender XDR tables use Timestamp; Sentinel / Log Analytics tables use TimeGenerated. This is correct per table, not an inconsistency.
- **Thresholds & time ranges:** Counts and windows are starting recommendations — tune them to your ingestion volume, retention, and baseline.

Section A — Identity: Sign-ins, Audit, Conditional Access, MFA, Privileged Accounts, OAuth

Query 1 — Entra ID Risk-Flagged Successful Sign-ins

Security Use Case: Surface successful interactive sign-ins that Entra ID Protection scored as medium/high risk — an early account-compromise indicator.

Supported Platform: Microsoft Sentinel · Azure Monitor / Log Analytics

Tables Used

Table	Purpose
SigninLogs	Entra ID interactive sign-in telemetry

Event / Action Types Used

Field	Value
ResultType	0 (successful sign-in)
RiskLevelDuringSignIn	high, medium

Fields Used

Field	Table	Description
TimeGenerated	SigninLogs	Event time
UserPrincipalName	SigninLogs	Account that signed in
IPAddress	SigninLogs	Source IP
Location	SigninLogs	Geo of source IP
AppDisplayName	SigninLogs	Target application

RiskLevelDuringSignIn	SigninLogs	Real-time risk score
RiskState	SigninLogs	Risk remediation state
ResultType	SigninLogs	Sign-in result code

Query

```
SigninLogs
| where TimeGenerated > ago(7d)
| where ResultType == 0
| where RiskLevelDuringSignIn in ("high", "medium")
| project TimeGenerated, UserPrincipalName, AppDisplayName, IPAddress, Location,
RiskLevelDuringSignIn, RiskState
| sort by TimeGenerated desc
```

Expected Output

- **Returned records:** Successful sign-ins carrying a medium/high real-time risk score, with user, app, IP and geo.
- **Detection value:** Risk-scored success is a stronger compromise signal than failures alone.
- **Investigation value:** Pivot on UserPrincipalName / IPAddress to find session scope and follow-on activity.

Performance Notes

- **High volume table:** Yes
- **Recommended time range:** 7 days
- **Optimization:** Filter ResultType == 0 and risk level early; project only needed columns.

Schema Verification

Table	Verified
SigninLogs	Yes
Field	**Verified**
TimeGenerated	Yes
UserPrincipalName	Yes
IPAddress	Yes
Location	Yes
AppDisplayName	Yes
RiskLevelDuringSignIn	Yes
RiskState	Yes
ResultType	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
SigninLogs	High	Identity sign-in hunting

Query 2 — Password Spray Detection (Failed Sign-ins Across Many Accounts)

Security Use Case: Detect a single source IP failing authentication against many distinct accounts in a short window.

Supported Platform: Microsoft Sentinel · Azure Monitor / Log Analytics

Tables Used

Table	Purpose
SigninLogs	Entra ID sign-in telemetry

Event / Action Types Used

Field	Value
ResultType	50126 (invalid username or password)

Fields Used

Field	Table	Description
TimeGenerated	SigninLogs	Event time
IPAddress	SigninLogs	Source IP
UserPrincipalName	SigninLogs	Targeted account
ResultType	SigninLogs	Result code
AppDisplayName	SigninLogs	Target app

Query

```
SigninLogs
| where TimeGenerated > ago(1d)
| where ResultType == 50126
| summarize FailedAttempts = count(), DistinctUsers = dcount(UserPrincipalName),
            Users = make_set(UserPrincipalName, 50) by IPAddress, bin(TimeGenerated, 1h)
| where DistinctUsers >= 10
| sort by DistinctUsers desc
```

Expected Output

- **Returned records:** Source IPs failing against 10+ distinct accounts per hour.
- **Detection value:** Classic spray signature (low attempts per account, broad account spread).
- **Investigation value:** Determine whether any account later succeeded from the same IP.

Performance Notes

- **High volume table:** Yes
- **Recommended time range:** 24 hours
- **Optimization:** Filter the single failure ResultType before summarize; tune DistinctUsers threshold.

Schema Verification

Table	Verified
SigninLogs	Yes
Field	**Verified**
TimeGenerated	Yes
IPAddress	Yes
UserPrincipalName	Yes
ResultType	Yes
AppDisplayName	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use

SigninLogs	High	Brute-force / spray hunting

Query 3 — Improbable Travel — Same User, Multiple Countries

Security Use Case: Detect a single account signing in successfully from two or more countries within a short window.

Supported Platform: Microsoft Sentinel · Azure Monitor / Log Analytics

Tables Used

Table	Purpose
SigninLogs	Entra ID sign-in telemetry

Event / Action Types Used

Field	Value
ResultType	0 (successful sign-in)

Fields Used

Field	Table	Description
TimeGenerated	SigninLogs	Event time
UserPrincipalName	SigninLogs	Account
IPAddress	SigninLogs	Source IP
LocationDetails	SigninLogs	Structured geo data

Query

```
SigninLogs
| where TimeGenerated > ago(1d)
| where ResultType == 0
| extend Country = tostring(LocationDetails.countryOrRegion)
| where isnotempty(Country)
| summarize Countries = make_set(Country), CountryCount = dcount(Country),
            IPs = make_set(IPAddress, 20) by UserPrincipalName, bin(TimeGenerated, 6h)
| where CountryCount >= 2
| sort by CountryCount desc
```

Expected Output

- **Returned records:** Users with successful sign-ins from 2+ countries inside a 6-hour bucket.
- **Detection value:** Geo-velocity anomaly suggesting token theft or shared credentials.
- **Investigation value:** Compare IPs and apps; correlate with known VPN / travel patterns.

Performance Notes

- **High volume table:** Yes
- **Recommended time range:** 24 hours
- **Optimization:** Extract country after filtering on success; heuristic — validate against legitimate VPN egress.

Schema Verification

Table	Verified
SigninLogs	Yes
Field	**Verified**

TimeGenerated	Yes
UserPrincipalName	Yes
IPAddress	Yes
LocationDetails	Yes
ResultType	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
SigninLogs	High	Anomalous geo / token theft

Query 4 — MFA Challenge Failures (Possible MFA Fatigue)

Security Use Case: Identify accounts accumulating repeated multi-factor challenge failures, indicating push-bombing or interrupted compromise.

Supported Platform: Microsoft Sentinel · Azure Monitor / Log Analytics

Tables Used

Table	Purpose
SigninLogs	Entra ID sign-in telemetry

Event / Action Types Used

Field	Value
ResultType	50074 (strong auth required)
ResultType	500121 (MFA authentication failed)

Fields Used

Field	Table	Description
TimeGenerated	SigninLogs	Event time
UserPrincipalName	SigninLogs	Account
IPAddress	SigninLogs	Source IP
ResultType	SigninLogs	Result code
ResultDescription	SigninLogs	Result text

Query

```
SigninLogs
| where TimeGenerated > ago(1d)
| where ResultType in (50074, 500121)
| summarize MfaFailures = count(), SourceIPs = make_set(IPAddress, 20),
            FirstSeen = min(TimeGenerated), LastSeen = max(TimeGenerated)
            by UserPrincipalName
| where MfaFailures >= 5
| sort by MfaFailures desc
```

Expected Output

- **Returned records:** Accounts with 5+ MFA-related failures in 24h plus their source IPs.
- **Detection value:** Repeated MFA denials are a hallmark of fatigue / push-bombing attacks.
- **Investigation value:** Check whether the same account had a subsequent ResultType == 0 (eventually approved).

Performance Notes

- **High volume table:** Yes
- **Recommended time range:** 24 hours
- **Optimization:** Filter the two MFA result codes before aggregation.

Schema Verification

Table	Verified
SigninLogs	Yes
Field	**Verified**
TimeGenerated	Yes
UserPrincipalName	Yes
IPAddress	Yes
ResultType	Yes
ResultDescription	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
SigninLogs	High	MFA abuse / account takeover

Query 5 — Conditional Access — Blocked Sign-ins by Policy

Security Use Case: Review sign-ins that Conditional Access blocked, to spot attacker probing and policy effectiveness.

Supported Platform: Microsoft Sentinel · Azure Monitor / Log Analytics

Tables Used

Table	Purpose
SigninLogs	Entra ID sign-in telemetry

Event / Action Types Used

Field	Value
ConditionalAccessStatus	failure

Fields Used

Field	Table	Description
TimeGenerated	SigninLogs	Event time
UserPrincipalName	SigninLogs	Account
IPAddress	SigninLogs	Source IP
AppDisplayName	SigninLogs	Target app
ConditionalAccessStatus	SigninLogs	CA evaluation result
ConditionalAccessPolicies	SigninLogs	Policies evaluated

Query

```
SigninLogs
| where TimeGenerated > ago(7d)
| where ConditionalAccessStatus == "failure"
```

```
| mv-expand Policy = ConditionalAccessPolicies
| where tostring(Policy.result) == "failure"
| project TimeGenerated, UserPrincipalName, AppDisplayName, IPAddress,
          PolicyName = tostring(Policy.displayName)
| summarize Blocks = count(), Users = dcount(UserPrincipalName) by PolicyName
| sort by Blocks desc
```

Expected Output

- **Returned records:** Conditional Access policies and how many sign-ins each blocked.
- **Detection value:** Surfaces policy-triggering activity (legacy auth, blocked geos, etc.).
- **Investigation value:** Drill into a high-block policy to find the accounts / IPs being denied.

Performance Notes

- **High volume table:** Yes
- **Recommended time range:** 7 days
- **Optimization:** mv-expand is expensive — filter ConditionalAccessStatus == "failure" first.

Schema Verification

Table	Verified
SigninLogs	Yes
Field	**Verified**
TimeGenerated	Yes
UserPrincipalName	Yes
IPAddress	Yes
AppDisplayName	Yes
ConditionalAccessStatus	Yes
ConditionalAccessPolicies	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
SigninLogs	High	CA policy / access-control review

Query 6 — Privileged Role Assignment via Entra Audit Logs

Security Use Case: Detect addition of accounts to privileged directory roles — a key privilege-escalation / persistence step.

Supported Platform: Microsoft Sentinel · Azure Monitor / Log Analytics

Tables Used

Table	Purpose
AuditLogs	Entra ID directory audit activity

Event / Action Types Used

Field	Value
OperationName	Add member to role

Fields Used

Field	Table	Description
TimeGenerated	AuditLogs	Event time
OperationName	AuditLogs	Directory operation
Result	AuditLogs	Operation outcome
InitiatedBy	AuditLogs	Actor performing the change
TargetResources	AuditLogs	Affected object(s)

Query

```
AuditLogs
| where TimeGenerated > ago(7d)
| where OperationName == "Add member to role"
| where Result == "success"
| extend Actor = tostring(InitiatedBy.user.userPrincipalName)
| mv-expand Target = TargetResources
| project TimeGenerated, OperationName, Actor,
          TargetUser = tostring(Target.userPrincipalName),
          RoleDetails = tostring(Target.modifiedProperties)
| sort by TimeGenerated desc
```

Expected Output

- **Returned records:** Successful additions of users to directory roles, with actor and target.
- **Detection value:** New privileged-role grants are high-value persistence indicators.
- **Investigation value:** Confirm the change matches a ticket / approval; review actor's recent activity.

Performance Notes

- **High volume table:** No
- **Recommended time range:** 7 days
- **Optimization:** Filter OperationName and Result before mv-expand.

Schema Verification

Table	Verified
AuditLogs	Yes
Field	**Verified**
TimeGenerated	Yes
OperationName	Yes
Result	Yes
InitiatedBy	Yes
TargetResources	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
AuditLogs	Moderate	Privilege escalation / config change

Query 7 — OAuth Application Consent Grants

Security Use Case: Detect users consenting to OAuth applications — a common illicit-consent / token-abuse vector.

Supported Platform: Microsoft Sentinel · Azure Monitor / Log Analytics

Tables Used

Table	Purpose
AuditLogs	Entra ID directory audit activity

Event / Action Types Used

Field	Value
OperationName	Consent to application

Fields Used

Field	Table	Description
TimeGenerated	AuditLogs	Event time
OperationName	AuditLogs	Directory operation
Result	AuditLogs	Outcome
InitiatedBy	AuditLogs	Consenting actor
TargetResources	AuditLogs	Application object

Query

```
AuditLogs
| where TimeGenerated > ago(14d)
| where OperationName == "Consent to application"
| extend Actor = tostring(InitiatedBy.user.userPrincipalName)
| mv-expand Target = TargetResources
| project TimeGenerated, Actor, Result,
          AppName = tostring(Target.displayName),
          Details = tostring(Target.modifiedProperties)
| sort by TimeGenerated desc
```

Expected Output

- **Returned records:** Consent events with the consenting user and target application.
- **Detection value:** Illicit consent grants enable persistent mailbox / data access without a password.
- **Investigation value:** Review requested permission scopes in modifiedProperties; verify app legitimacy.

Performance Notes

- **High volume table:** No
- **Recommended time range:** 14 days
- **Optimization:** Filter OperationName before expanding target resources.

Schema Verification

Table	Verified
AuditLogs	Yes
Field	**Verified**
TimeGenerated	Yes
OperationName	Yes
Result	Yes
InitiatedBy	Yes
TargetResources	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
AuditLogs	Moderate	OAuth / illicit consent hunting

Query 8 — New User Account Creation (Entra)

Security Use Case: Track creation of new Entra user accounts to catch attacker-provisioned identities.

Supported Platform: Microsoft Sentinel · Azure Monitor / Log Analytics

Tables Used

Table	Purpose
AuditLogs	Entra ID directory audit activity

Event / Action Types Used

Field	Value
OperationName	Add user

Fields Used

Field	Table	Description
TimeGenerated	AuditLogs	Event time
OperationName	AuditLogs	Directory operation
Result	AuditLogs	Outcome
InitiatedBy	AuditLogs	Actor
TargetResources	AuditLogs	New user object

Query

```
AuditLogs
| where TimeGenerated > ago(7d)
| where OperationName == "Add user"
| where Result == "success"
| extend Actor = tostring(InitiatedBy.user.userPrincipalName)
| mv-expand Target = TargetResources
| project TimeGenerated, Actor, NewUser = tostring(Target.userPrincipalName)
| sort by TimeGenerated desc
```

Expected Output

- **Returned records:** Newly created accounts with the creating actor.
- **Detection value:** Unexpected account creation may indicate attacker persistence.
- **Investigation value:** Cross-check against HR onboarding; review what the new account did next.

Performance Notes

- **High volume table:** No
- **Recommended time range:** 7 days
- **Optimization:** Filter operation and result first.

Schema Verification

Table	Verified
AuditLogs	Yes

Field	Verified
TimeGenerated	Yes
OperationName	Yes
Result	Yes
InitiatedBy	Yes
TargetResources	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
AuditLogs	Moderate	Identity provisioning anomaly

Section B — Endpoint: Process, PowerShell, CMD, WMI, Tasks, Services, Registry, Credential Access, Logon

Query 9 — Encoded PowerShell Command Execution

Security Use Case: Detect PowerShell invoked with an encoded command argument — a near-ubiquitous obfuscation technique.

Supported Platform: Microsoft Defender XDR

Tables Used

Table	Purpose
DeviceProcessEvents	Process execution telemetry

Event / Action Types Used

Field	Value
ActionType	ProcessCreated

Fields Used

Field	Table	Description
Timestamp	DeviceProcessEvents	Event time
DeviceName	DeviceProcessEvents	Endpoint hostname
AccountName	DeviceProcessEvents	Executing account
FileName	DeviceProcessEvents	Process image name
ProcessCommandLine	DeviceProcessEvents	Full command line
InitiatingProcessFileName	DeviceProcessEvents	Parent process

Query

```
DeviceProcessEvents
| where Timestamp > ago(7d)
| where ActionType == "ProcessCreated"
| where FileName in~ ("powershell.exe", "pwsh.exe")
| where ProcessCommandLine has_any ("-enc", "-encodedcommand", "-e ")
| project Timestamp, DeviceName, AccountName, ProcessCommandLine, InitiatingProcessFileName
| sort by Timestamp desc
```

Expected Output

- **Returned records:** PowerShell launches using encoded-command flags, with parent process.
- **Detection value:** Encoded commands are heavily used by malware and operators to hide intent.
- **Investigation value:** Base64-decode the payload; pivot on parent process and device.

Performance Notes

- **High volume table:** Yes
- **Recommended time range:** 7 days
- **Optimization:** Use has_any (term-indexed) rather than contains; filter FileName first.

Schema Verification

Table	Verified
DeviceProcessEvents	Yes
Field	**Verified**
Timestamp	Yes
DeviceName	Yes
AccountName	Yes
FileName	Yes
ProcessCommandLine	Yes
InitiatingProcessFileName	Yes
ActionType	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
DeviceProcessEvents	Very High	Process execution hunting

Query 10 — PowerShell Download Cradle

Security Use Case: Detect PowerShell pulling remote content via common download primitives (download cradles).

Supported Platform: Microsoft Defender XDR

Tables Used

Table	Purpose
DeviceProcessEvents	Process execution telemetry

Event / Action Types Used

Field	Value
ActionType	ProcessCreated

Fields Used

Field	Table	Description
Timestamp	DeviceProcessEvents	Event time
DeviceName	DeviceProcessEvents	Endpoint hostname
AccountName	DeviceProcessEvents	Executing account

FileName	DeviceProcessEvents	Process image name
ProcessCommandLine	DeviceProcessEvents	Full command line
InitiatingProcessFileName	DeviceProcessEvents	Parent process

Query

```
DeviceProcessEvents
| where Timestamp > ago(7d)
| where FileName in~ ("powershell.exe", "pwsh.exe")
| where ProcessCommandLine has_any
        ("downloadstring", "downloadfile", "invoke-webrequest", "iwr",
         "invoke-restmethod", "net.webclient", "start-bitstransfer")
| project Timestamp, DeviceName, AccountName, ProcessCommandLine, InitiatingProcessFileName
| sort by Timestamp desc
```

Expected Output

- **Returned records:** PowerShell processes using web / download primitives.
- **Detection value:** Download cradles are a primary delivery mechanism for second-stage payloads.
- **Investigation value:** Extract the URL/host; correlate with DeviceNetworkEvents for the actual connection.

Performance Notes

- **High volume table:** Yes
- **Recommended time range:** 7 days
- **Optimization:** has_any over a single term list; pre-filter on FileName.

Schema Verification

Table	Verified
DeviceProcessEvents	Yes
Field	**Verified**
Timestamp	Yes
DeviceName	Yes
AccountName	Yes
FileName	Yes
ProcessCommandLine	Yes
InitiatingProcessFileName	Yes
ActionType	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
DeviceProcessEvents	Very High	Malware delivery hunting

Query 11 — Suspicious CMD Chaining / Discovery

Security Use Case: Detect cmd.exe running chained discovery commands often used in hands-on-keyboard intrusions.

Supported Platform: Microsoft Defender XDR

Tables Used

Table	Purpose

DeviceProcessEvents	Process execution telemetry

Event / Action Types Used

Field	Value
ActionType	ProcessCreated

Fields Used

Field	Table	Description
Timestamp	DeviceProcessEvents	Event time
DeviceName	DeviceProcessEvents	Endpoint hostname
AccountName	DeviceProcessEvents	Executing account
FileName	DeviceProcessEvents	Process image name
ProcessCommandLine	DeviceProcessEvents	Full command line
InitiatingProcessFileName	DeviceProcessEvents	Parent process

Query

```
DeviceProcessEvents
| where Timestamp > ago(3d)
| where FileName =~ "cmd.exe"
| where ProcessCommandLine has_any
        ("whoami", "net group", "net user", "nltest", "systeminfo",
         "ipconfig /all", "tasklist", "query user")
| project Timestamp, DeviceName, AccountName, ProcessCommandLine, InitiatingProcessFileName
| sort by Timestamp desc
```

Expected Output

- **Returned records:** cmd.exe invocations running host / domain discovery commands.
- **Detection value:** Clustered discovery is a strong early-intrusion behavior.
- **Investigation value:** Group by device/account; bursts from one host within minutes warrant triage.

Performance Notes

- **High volume table:** Yes
- **Recommended time range:** 3 days
- **Optimization:** Term-indexed has_any; consider summarizing by device to find bursts.

Schema Verification

Table	Verified
DeviceProcessEvents	Yes
Field	**Verified**
Timestamp	Yes
DeviceName	Yes
AccountName	Yes
FileName	Yes
ProcessCommandLine	Yes
InitiatingProcessFileName	Yes
ActionType	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
DeviceProcessEvents	Very High	Discovery / recon hunting

Query 12 — WMI Command-Line Execution (wmic process spawn)

Security Use Case: Detect wmic.exe used for remote execution or lateral movement (process call create, /node:).

Supported Platform: Microsoft Defender XDR

Tables Used

Table	Purpose
DeviceProcessEvents	Process execution telemetry

Event / Action Types Used

Field	Value
ActionType	ProcessCreated

Fields Used

Field	Table	Description
Timestamp	DeviceProcessEvents	Event time
DeviceName	DeviceProcessEvents	Endpoint hostname
AccountName	DeviceProcessEvents	Executing account
FileName	DeviceProcessEvents	Process image name
ProcessCommandLine	DeviceProcessEvents	Full command line
InitiatingProcessFileName	DeviceProcessEvents	Parent process

Query

```
DeviceProcessEvents
| where Timestamp > ago(7d)
| where FileName =~ "wmic.exe"
| where ProcessCommandLine has_any ("process call create", "/node:", "/user:")
| project Timestamp, DeviceName, AccountName, ProcessCommandLine, InitiatingProcessFileName
| sort by Timestamp desc
```

Expected Output

- **Returned records:** wmic.exe used to spawn processes or target remote nodes.
- **Detection value:** WMI is a favored fileless lateral-movement / execution channel.
- **Investigation value:** Note /node: targets and remote credentials; pivot to the target host.

Performance Notes

- **High volume table:** Yes
- **Recommended time range:** 7 days
- **Optimization:** Filter on FileName then has_any.

Schema Verification

Table	Verified
DeviceProcessEvents	Yes

Field	Verified
Timestamp	Yes
DeviceName	Yes
AccountName	Yes
FileName	Yes
ProcessCommandLine	Yes
InitiatingProcessFileName	Yes
ActionType	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
DeviceProcessEvents	Very High	WMI / lateral movement hunting

Query 13 — Scheduled Task Creation (DeviceEvents)

Security Use Case: Detect scheduled task creation, a common persistence and lateral-execution mechanism.

Supported Platform: Microsoft Defender XDR

Tables Used

Table	Purpose
DeviceEvents	Miscellaneous device / security-control events

Event / Action Types Used

Field	Value
ActionType	ScheduledTaskCreated

Fields Used

Field	Table	Description
Timestamp	DeviceEvents	Event time
DeviceName	DeviceEvents	Endpoint hostname
ActionType	DeviceEvents	Event type
InitiatingProcessFileName	DeviceEvents	Process that created the task
InitiatingProcessCommandLine	DeviceEvents	Creating command line
InitiatingProcessAccountName	DeviceEvents	Account
AdditionalFields	DeviceEvents	Task metadata

Query

```
DeviceEvents
| where Timestamp > ago(7d)
| where ActionType == "ScheduledTaskCreated"
| extend TaskName = tostring(AdditionalFields.TaskName)
| project Timestamp, DeviceName, TaskName, InitiatingProcessAccountName,
          InitiatingProcessFileName, InitiatingProcessCommandLine
| sort by Timestamp desc
```

Expected Output

- **Returned records:** Scheduled tasks created, with task name and creating process / account.

- **Detection value:** Persistence via Task Scheduler is heavily used by commodity and targeted malware.
- **Investigation value:** Inspect task action/command in AdditionalFields; flag tasks pointing at scripts or temp paths.

Performance Notes

- **High volume table:** No
- **Recommended time range:** 7 days
- **Optimization:** Filter ActionType first; extract AdditionalFields only after filtering.

Schema Verification

Table	Verified
DeviceEvents	Yes
Field	**Verified**
Timestamp	Yes
DeviceName	Yes
ActionType	Yes
InitiatingProcessFileName	Yes
InitiatingProcessCommandLine	Yes
InitiatingProcessAccountName	Yes
AdditionalFields	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
DeviceEvents	Moderate	Persistence hunting

Query 14 — Windows Service Installation (SecurityEvent 7045)

Security Use Case: Detect new Windows service installation — used for persistence and remote execution (e.g. PsExec).

Supported Platform: Microsoft Sentinel · Azure Monitor / Log Analytics

Tables Used

Table	Purpose
SecurityEvent	Windows Security event log via AMA / Log Analytics

Event / Action Types Used

Field	Value
EventID	7045 (a service was installed in the system)

Fields Used

Field	Table	Description
TimeGenerated	SecurityEvent	Event time
Computer	SecurityEvent	Host
EventID	SecurityEvent	Windows event identifier
Account	SecurityEvent	Installing account

ServiceName	SecurityEvent	New service name
ServiceFileName	SecurityEvent	Service binary path

Query

```
SecurityEvent
| where TimeGenerated > ago(7d)
| where EventID == 7045
| project TimeGenerated, Computer, Account, ServiceName, ServiceFileName
| sort by TimeGenerated desc
```

Expected Output

- **Returned records:** Service-install events with service name and binary path.
- **Detection value:** Suspicious binary paths (temp dirs, encoded names) indicate malicious services.
- **Investigation value:** Validate the service against known software; inspect ServiceFileName.

Performance Notes

- **High volume table:** Yes
- **Recommended time range:** 7 days
- **Optimization:** Filter EventID == 7045 first; ensure 7045 collection is enabled in the DCR.

Schema Verification

Table	Verified
SecurityEvent	Yes
Field	**Verified**
TimeGenerated	Yes
Computer	Yes
EventID	Yes
Account	Yes
ServiceName	Yes
ServiceFileName	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
SecurityEvent	Very High	Service persistence hunting

Query 15 — Registry Run-Key Persistence

Security Use Case: Detect writes to autorun registry locations used for persistence at logon / boot.

Supported Platform: Microsoft Defender XDR

Tables Used

Table	Purpose
DeviceRegistryEvents	Registry creation / modification telemetry

Event / Action Types Used

Field	Value
ActionType	RegistryValueSet

Fields Used

Field	Table	Description
Timestamp	DeviceRegistryEvents	Event time
DeviceName	DeviceRegistryEvents	Host
ActionType	DeviceRegistryEvents	Event type
RegistryKey	DeviceRegistryEvents	Affected key
RegistryValueName	DeviceRegistryEvents	Value name
RegistryValueData	DeviceRegistryEvents	Value data
InitiatingProcessFileName	DeviceRegistryEvents	Writing process

Query

```
DeviceRegistryEvents
| where Timestamp > ago(7d)
| where ActionType == "RegistryValueSet"
| where RegistryKey has_any
        (@"CurrentVersion\Run", @"CurrentVersion\RunOnce",
         @"CurrentVersion\Explorer\Shell Folders")
| project Timestamp, DeviceName, RegistryKey, RegistryValueName,
          RegistryValueData, InitiatingProcessFileName
| sort by Timestamp desc
```

Expected Output

- **Returned records:** Autorun registry value writes with the data and writing process.
- **Detection value:** Run / RunOnce keys are classic persistence anchors.
- **Investigation value:** Examine RegistryValueData for the executed path; correlate to process events.

Performance Notes

- **High volume table:** No
- **Recommended time range:** 7 days
- **Optimization:** Filter ActionType first; has_any on the key path.

Schema Verification

Table	Verified
DeviceRegistryEvents	Yes
Field	**Verified**
Timestamp	Yes
DeviceName	Yes
ActionType	Yes
RegistryKey	Yes
RegistryValueName	Yes
RegistryValueData	Yes
InitiatingProcessFileName	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
DeviceRegistryEvents	Moderate	Persistence hunting

Query 16 — LSASS Credential Dumping via Command-Line Tooling

Security Use Case: Detect command lines targeting LSASS for credential theft (comsvcs.dll MiniDump, procdump on lsass).

Supported Platform: Microsoft Defender XDR

Tables Used

Table	Purpose
DeviceProcessEvents	Process execution telemetry

Event / Action Types Used

Field	Value
ActionType	ProcessCreated

Fields Used

Field	Table	Description
Timestamp	DeviceProcessEvents	Event time
DeviceName	DeviceProcessEvents	Host
AccountName	DeviceProcessEvents	Account
FileName	DeviceProcessEvents	Process image
ProcessCommandLine	DeviceProcessEvents	Command line
InitiatingProcessFileName	DeviceProcessEvents	Parent process

Query

```
DeviceProcessEvents
| where Timestamp > ago(7d)
| where ProcessCommandLine has "lsass"
| where ProcessCommandLine has_any ("minidump", "procdump", "comsvcs", "-ma ", "dump")
| project Timestamp, DeviceName, AccountName, FileName, ProcessCommandLine,
InitiatingProcessFileName
| sort by Timestamp desc
```

Expected Output

- **Returned records:** Command lines that reference LSASS together with dumping verbs.
- **Detection value:** Credential dumping from LSASS is a high-severity ATT&CK technique (T1003.001).
- **Investigation value:** Identify the tool / process; isolate the host; rotate exposed credentials.

Performance Notes

- **High volume table:** Yes
- **Recommended time range:** 7 days
- **Optimization:** Lead with has "lsass" (term-indexed) to shrink the set before the second filter.

Schema Verification

Table	Verified
DeviceProcessEvents	Yes
Field	**Verified**
Timestamp	Yes

DeviceName	Yes
AccountName	Yes
FileName	Yes
ProcessCommandLine	Yes
InitiatingProcessFileName	Yes
ActionType	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
DeviceProcessEvents	Very High	Credential access hunting

Query 17 — LOLBin Abuse — rundll32 / regsvr32 / mshta

Security Use Case: Detect living-off-the-land binaries executing remote or scriptlet payloads.

Supported Platform: Microsoft Defender XDR

Tables Used

Table	Purpose
DeviceProcessEvents	Process execution telemetry

Event / Action Types Used

Field	Value
ActionType	ProcessCreated

Fields Used

Field	Table	Description
Timestamp	DeviceProcessEvents	Event time
DeviceName	DeviceProcessEvents	Host
AccountName	DeviceProcessEvents	Account
FileName	DeviceProcessEvents	Process image
ProcessCommandLine	DeviceProcessEvents	Command line
InitiatingProcessFileName	DeviceProcessEvents	Parent process

Query

```
DeviceProcessEvents
| where Timestamp > ago(7d)
| where FileName in~ ("rundll32.exe", "regsvr32.exe", "mshta.exe")
| where ProcessCommandLine has_any
        ("http://", "https://", "javascript:", "vbscript:", "scrobj", "/i:")
| project Timestamp, DeviceName, AccountName, FileName, ProcessCommandLine,
InitiatingProcessFileName
| sort by Timestamp desc
```

Expected Output

- **Returned records:** LOLBins invoked with URL or scriptlet arguments.
- **Detection value:** These signed binaries are abused to proxy execution and bypass controls.
- **Investigation value:** Inspect remote URL / COM scriptlet; check parent process legitimacy.

Performance Notes

- **High volume table:** Yes
- **Recommended time range:** 7 days
- **Optimization:** Filter FileName set first; has_any for argument terms.

Schema Verification

Table	Verified
DeviceProcessEvents	Yes
Field	**Verified**
Timestamp	Yes
DeviceName	Yes
AccountName	Yes
FileName	Yes
ProcessCommandLine	Yes
InitiatingProcessFileName	Yes
ActionType	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
DeviceProcessEvents	Very High	LOLBin / proxy-execution hunting

Query 18 — Failed Logons by Host (Endpoint Brute Force)

Security Use Case: Identify endpoints accumulating many failed local / network logons.

Supported Platform: Microsoft Defender XDR

Tables Used

Table	Purpose
DeviceLogonEvents	Authentication activity on devices

Event / Action Types Used

Field	Value
ActionType	LogonFailed

Fields Used

Field	Table	Description
Timestamp	DeviceLogonEvents	Event time
DeviceName	DeviceLogonEvents	Host
ActionType	DeviceLogonEvents	Logon result
AccountName	DeviceLogonEvents	Target account
LogonType	DeviceLogonEvents	Logon session type
RemoteIP	DeviceLogonEvents	Source IP
FailureReason	DeviceLogonEvents	Why it failed

Query

```
DeviceLogonEvents
| where Timestamp > ago(1d)
| where ActionType == "LogonFailed"
| summarize Failures = count(), Accounts = dcount(AccountName),
            SourceIPs = make_set(RemoteIP, 20) by DeviceName, bin(Timestamp, 1h)
| where Failures >= 20
| sort by Failures desc
```

Expected Output

- **Returned records:** Hosts with 20+ failed logons per hour and the targeted accounts / IPs.
- **Detection value:** Endpoint-side brute force / spray indicator.
- **Investigation value:** Determine logon type and whether any attempt later succeeded.

Performance Notes

- **High volume table:** Yes
- **Recommended time range:** 24 hours
- **Optimization:** Filter ActionType before summarize; tune threshold per environment.

Schema Verification

Table	Verified
DeviceLogonEvents	Yes
Field	**Verified**
Timestamp	Yes
DeviceName	Yes
ActionType	Yes
AccountName	Yes
LogonType	Yes
RemoteIP	Yes
FailureReason	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
DeviceLogonEvents	High	Authentication abuse hunting

Query 19 — Inbound RDP Logons (RemoteInteractive)

Security Use Case: Track successful RDP sessions to spot lateral movement and unexpected remote access.

Supported Platform: Microsoft Defender XDR

Tables Used

Table	Purpose
DeviceLogonEvents	Authentication activity on devices

Event / Action Types Used

Field	Value
ActionType	LogonSuccess
LogonType	RemoteInteractive

Fields Used

Field	Table	Description
Timestamp	DeviceLogonEvents	Event time
DeviceName	DeviceLogonEvents	Host
ActionType	DeviceLogonEvents	Logon result
LogonType	DeviceLogonEvents	Session type
AccountName	DeviceLogonEvents	Account
RemoteIP	DeviceLogonEvents	Source IP
RemoteDeviceName	DeviceLogonEvents	Source host

Query

```
DeviceLogonEvents
| where Timestamp > ago(7d)
| where ActionType == "LogonSuccess"
| where LogonType == "RemoteInteractive"
| project Timestamp, DeviceName, AccountName, RemoteIP, RemoteDeviceName
| sort by Timestamp desc
```

Expected Output

- **Returned records:** Successful RDP logons with account, source IP and source host.
- **Detection value:** Reveals interactive remote access paths used in lateral movement.
- **Investigation value:** Map RDP chains (host A -> B -> C) and flag external / unknown sources.

Performance Notes

- **High volume table:** Yes
- **Recommended time range:** 7 days
- **Optimization:** Filter on both ActionType and LogonType early.

Schema Verification

Table	Verified
DeviceLogonEvents	Yes
Field	**Verified**
Timestamp	Yes
DeviceName	Yes
ActionType	Yes
LogonType	Yes
AccountName	Yes
RemoteIP	Yes
RemoteDeviceName	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
DeviceLogonEvents	High	RDP / lateral movement hunting

Section C — Network, DNS, SMB, Exfiltration

Query 20 — SMB Connections to Internal Hosts (Lateral Movement)

Security Use Case: Detect outbound SMB (TCP 445) initiated by non-standard processes — a lateral-movement indicator.

Supported Platform: Microsoft Defender XDR

Tables Used

Table	Purpose
DeviceNetworkEvents	Network connection telemetry

Event / Action Types Used

Field	Value
ActionType	ConnectionSuccess

Fields Used

Field	Table	Description
Timestamp	DeviceNetworkEvents	Event time
DeviceName	DeviceNetworkEvents	Source host
ActionType	DeviceNetworkEvents	Connection result
RemoteIP	DeviceNetworkEvents	Destination IP
RemotePort	DeviceNetworkEvents	Destination port
InitiatingProcessFileName	DeviceNetworkEvents	Initiating process
InitiatingProcessCommandLine	DeviceNetworkEvents	Process command line

Query

```
DeviceNetworkEvents
| where Timestamp > ago(3d)
| where ActionType == "ConnectionSuccess"
| where RemotePort == 445
| where InitiatingProcessFileName !in~ ("System", "svchost.exe")
| project Timestamp, DeviceName, RemoteIP, RemotePort,
          InitiatingProcessFileName, InitiatingProcessCommandLine
| sort by Timestamp desc
```

Expected Output

- **Returned records:** Successful SMB connections from unusual processes.
- **Detection value:** Tools copying payloads / executing over SMB stand out from normal OS traffic.
- **Investigation value:** Map source -> destination pairs; investigate the initiating process.

Performance Notes

- **High volume table:** Yes
- **Recommended time range:** 3 days
- **Optimization:** Filter RemotePort == 445 and ActionType first; exclude noisy OS processes.

Schema Verification

Table	Verified

DeviceNetworkEvents	Yes
Field	**Verified**
Timestamp	Yes
DeviceName	Yes
ActionType	Yes
RemoteIP	Yes
RemotePort	Yes
InitiatingProcessFileName	Yes
InitiatingProcessCommandLine	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
DeviceNetworkEvents	High	Lateral movement hunting

Query 21 — Beacon-Like Outbound Connections (Rare Destinations)

Security Use Case: Surface processes making outbound connections to rarely-contacted remote IPs — possible C2.

Supported Platform: Microsoft Defender XDR

Tables Used

Table	Purpose
DeviceNetworkEvents	Network connection telemetry

Event / Action Types Used

Field	Value
ActionType	ConnectionSuccess

Fields Used

Field	Table	Description
Timestamp	DeviceNetworkEvents	Event time
DeviceName	DeviceNetworkEvents	Source host
RemoteIP	DeviceNetworkEvents	Destination IP
RemoteUrl	DeviceNetworkEvents	Destination URL/FQDN
InitiatingProcessFileName	DeviceNetworkEvents	Initiating process

Query

```
DeviceNetworkEvents
| where Timestamp > ago(7d)
| where ActionType == "ConnectionSuccess"
| where isnotempty(RemoteIP)
| summarize Hosts = dcount(DeviceName), Connections = count(),
            Processes = make_set(InitiatingProcessFileName, 10)
            by RemoteIP, RemoteUrl
| where Hosts <= 2 and Connections >= 20
| sort by Connections desc
```

Expected Output

- **Returned records:** Remote destinations contacted by very few hosts but with high connection counts.

- **Detection value:** Rare-destination + repetitive connections is a beaconing heuristic.
- **Investigation value:** Examine the initiating process and timing regularity for C2 confirmation.

Performance Notes

- **High volume table:** Yes
- **Recommended time range:** 7 days
- **Optimization:** Aggregates a high-volume table — keep the window tight; validate against known SaaS endpoints.

Schema Verification

Table	Verified
DeviceNetworkEvents	Yes
Field	**Verified**
Timestamp	Yes
DeviceName	Yes
RemoteIP	Yes
RemoteUrl	Yes
InitiatingProcessFileName	Yes
ActionType	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
DeviceNetworkEvents	High	C2 / beaconing hunting

Query 22 — DNS Queries to Long / High-Entropy Domains

Security Use Case: Detect connections to unusually long domains, a heuristic for DGA / DNS tunneling.

Supported Platform: Microsoft Defender XDR

Tables Used

Table	Purpose
DeviceNetworkEvents	Network connection telemetry (includes resolved URL/FQDN)

Event / Action Types Used

No specific event type filter used.

Fields Used

Field	Table	Description
Timestamp	DeviceNetworkEvents	Event time
DeviceName	DeviceNetworkEvents	Host
RemoteUrl	DeviceNetworkEvents	Queried / contacted FQDN
InitiatingProcessFileName	DeviceNetworkEvents	Initiating process

Query

```
DeviceNetworkEvents
| where Timestamp > ago(3d)
```

```
| where isnotempty(RemoteUrl)
| extend DomainLength = strlen(RemoteUrl)
| where DomainLength > 50
| project Timestamp, DeviceName, RemoteUrl, DomainLength, InitiatingProcessFileName
| sort by DomainLength desc
```

Expected Output

- **Returned records:** Connections / lookups to abnormally long FQDNs with the initiating process.
- **Detection value:** DGA domains and DNS tunneling often produce very long labels.
- **Investigation value:** Inspect subdomain structure and the process generating the traffic.

Performance Notes

- **High volume table:** Yes
- **Recommended time range:** 3 days
- **Optimization:** strlen runs per row — keep the window small; raise the length threshold to cut noise.

Schema Verification

Table	Verified
DeviceNetworkEvents	Yes
Field	**Verified**
Timestamp	Yes
DeviceName	Yes
RemoteUrl	Yes
InitiatingProcessFileName	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
DeviceNetworkEvents	High	DNS tunneling / DGA hunting

Section D — Email, Phishing, Malware

Query 23 — Phishing Emails Delivered to Inbox

Security Use Case: Find messages classified as phish that still reached the inbox (not blocked / junked).

Supported Platform: Microsoft Defender XDR

Tables Used

Table	Purpose
EmailEvents	Email delivery and verdict telemetry

Event / Action Types Used

Field	Value
ThreatTypes	Phish
DeliveryLocation	Inbox/folder

Fields Used

Field	Table	Description
Timestamp	EmailEvents	Event time
SenderFromAddress	EmailEvents	From address
RecipientEmailAddress	EmailEvents	Recipient
Subject	EmailEvents	Subject line
ThreatTypes	EmailEvents	Detected threat category
DeliveryAction	EmailEvents	Delivered / Blocked
DeliveryLocation	EmailEvents	Final placement

Query

```
EmailEvents
| where Timestamp > ago(7d)
| where ThreatTypes has "Phish"
| where DeliveryLocation == "Inbox/folder"
| project Timestamp, SenderFromAddress, RecipientEmailAddress, Subject,
          ThreatTypes, DeliveryAction, DeliveryLocation
| sort by Timestamp desc
```

Expected Output

- **Returned records:** Phish-verdict emails landing in the inbox with sender / recipient / subject.
- **Detection value:** Highlights detection gaps where phish bypassed remediation.
- **Investigation value:** Trigger ZAP / remediation; pivot to EmailUrlInfo for clicked links.

Performance Notes

- **High volume table:** Yes
- **Recommended time range:** 7 days
- **Optimization:** Filter ThreatTypes and DeliveryLocation before projecting.

Schema Verification

Table	Verified
EmailEvents	Yes
Field	**Verified**
Timestamp	Yes
SenderFromAddress	Yes
RecipientEmailAddress	Yes
Subject	Yes
ThreatTypes	Yes
DeliveryAction	Yes
DeliveryLocation	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
EmailEvents	High	Phishing / email threat hunting

Query 24 — Malicious Email Attachments by Threat Name

Security Use Case: Correlate emails with their attachments flagged as malware.

Supported Platform: Microsoft Defender XDR

Tables Used

Table	Purpose
EmailAttachmentInfo	Attachment metadata and verdicts
EmailEvents	Delivery context (join)

Event / Action Types Used

Field	Value
ThreatTypes	Malware

Fields Used

Field	Table	Description
Timestamp	EmailAttachmentInfo	Event time
NetworkMessageId	EmailAttachmentInfo	Message correlation key
FileName	EmailAttachmentInfo	Attachment name
FileType	EmailAttachmentInfo	Attachment type
SHA256	EmailAttachmentInfo	Attachment hash
ThreatTypes	EmailAttachmentInfo	Verdict
RecipientEmailAddress	EmailEvents	Recipient
SenderFromAddress	EmailEvents	Sender

Query

```
EmailAttachmentInfo
| where Timestamp > ago(7d)
| where ThreatTypes has "Malware"
| join kind=inner (
    EmailEvents
    | where Timestamp > ago(7d)
    | project NetworkMessageId, SenderFromAddress, RecipientEmailAddress, Subject
) on NetworkMessageId
| project Timestamp, SenderFromAddress, RecipientEmailAddress, Subject,
          FileName, FileType, SHA256, ThreatTypes
| sort by Timestamp desc
```

Expected Output

- **Returned records:** Malicious attachments joined to their delivery context.
- **Detection value:** Identifies malware-laden mail and the recipients exposed.
- **Investigation value:** Use SHA256 to search DeviceFileEvents for execution on endpoints.

Performance Notes

- **High volume table:** Yes
- **Recommended time range:** 7 days
- **Optimization:** Filter ThreatTypes on the left table before the join; project minimal columns inside the join.

Schema Verification

Table	Verified

EmailAttachmentInfo	Yes
EmailEvents	Yes
Field	**Verified**
Timestamp	Yes
NetworkMessageId	Yes
FileName	Yes
FileType	Yes
SHA256	Yes
ThreatTypes	Yes
RecipientEmailAddress	Yes
SenderFromAddress	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
EmailAttachmentInfo	High	Malware delivery hunting
EmailEvents	High	Email context correlation

Section E — Cloud Control Plane, Alerts, Incident & Insider Threat

Query 25 — Azure Key Vault Control-Plane Operations

Security Use Case: Monitor management-plane operations against Key Vault resources (create/update/delete, access-policy changes). Note: data-plane secret reads go to AzureDiagnostics, which is outside this library's approved-table set; this query covers control-plane activity via the approved AzureActivity table.

Supported Platform: Microsoft Sentinel · Azure Monitor / Log Analytics

Tables Used

Table	Purpose
AzureActivity	Azure Resource Manager control-plane activity

Event / Action Types Used

No specific event type filter used.

Fields Used

Field	Table	Description
TimeGenerated	AzureActivity	Event time
OperationNameValue	AzureActivity	ARM operation
ActivityStatusValue	AzureActivity	Operation status
Caller	AzureActivity	Identity performing the action
CallerIpAddress	AzureActivity	Source IP
ResourceGroup	AzureActivity	Resource group
ResourceProviderValue	AzureActivity	Resource provider

_ResourceId	AzureActivity	Target resource

Query

```
AzureActivity
| where TimeGenerated > ago(7d)
| where ResourceProviderValue =~ "Microsoft.KeyVault"
| project TimeGenerated, OperationNameValue, ActivityStatusValue,
          Caller, CallerIpAddress, ResourceGroup, _ResourceId
| sort by TimeGenerated desc
```

Expected Output

- **Returned records:** Control-plane Key Vault operations with caller identity and source IP.
- **Detection value:** Vault deletion, access-policy edits, or unexpected callers can indicate tampering.
- **Investigation value:** Correlate Caller / CallerIpAddress with sign-in activity; verify change authorization.

Performance Notes

- **High volume table:** No
- **Recommended time range:** 7 days
- **Optimization:** Filter ResourceProviderValue first; use ActivityStatusValue to focus on succeeded / failed.

Schema Verification

Table	Verified
AzureActivity	Yes
Field	**Verified**
TimeGenerated	Yes
OperationNameValue	Yes
ActivityStatusValue	Yes
Caller	Yes
CallerIpAddress	Yes
ResourceGroup	Yes
ResourceProviderValue	Yes
_ResourceId	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
AzureActivity	Moderate	Cloud control-plane monitoring

Query 26 — Kerberoasting — RC4 Service Ticket Requests (4769)

Security Use Case: Detect TGS requests using weak RC4 encryption against user service accounts — the signature of Kerberoasting.

Supported Platform: Microsoft Sentinel · Azure Monitor / Log Analytics

Tables Used

Table	Purpose
SecurityEvent	Windows Security event log via AMA / Log Analytics

Event / Action Types Used

Field	Value
EventID	4769 (a Kerberos service ticket was requested)
TicketEncryptionType	0x17 (RC4-HMAC)

Fields Used

Field	Table	Description
TimeGenerated	SecurityEvent	Event time
Computer	SecurityEvent	Domain controller
EventID	SecurityEvent	Windows event identifier
TargetUserName	SecurityEvent	Requesting account
ServiceName	SecurityEvent	Target SPN / service account
TicketEncryptionType	SecurityEvent	Kerberos encryption type
IpAddress	SecurityEvent	Source IP

Query

```
SecurityEvent
| where TimeGenerated > ago(7d)
| where EventID == 4769
| where TicketEncryptionType == "0x17"
| where ServiceName !endswith "$"
| summarize Requests = count(), Services = make_set(ServiceName, 20)
            by TargetUserName, IpAddress, bin(TimeGenerated, 1h)
| where Requests >= 10
| sort by Requests desc
```

Expected Output

- **Returned records:** Accounts requesting many RC4 service tickets within an hour.
- **Detection value:** RC4 TGS bursts indicate offline-crackable ticket harvesting (T1558.003).
- **Investigation value:** Identify targeted SPNs / service accounts; rotate passwords; check the requesting host.

Performance Notes

- **High volume table:** Yes
- **Recommended time range:** 7 days
- **Optimization:** Filter EventID and encryption type first; ensure 4769 is collected in the DCR.

Schema Verification

Table	Verified
SecurityEvent	Yes
Field	**Verified**
TimeGenerated	Yes
Computer	Yes
EventID	Yes
TargetUserName	Yes
ServiceName	Yes
TicketEncryptionType	Yes
IpAddress	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
SecurityEvent	Very High	Kerberos abuse hunting

Query 27 — Active Directory Reconnaissance (Identity Queries)

Security Use Case: Detect LDAP / SAMR enumeration of users and groups preceding lateral movement.

Supported Platform: Microsoft Defender XDR

Tables Used

Table	Purpose
IdentityQueryEvents	Active Directory query telemetry (Defender for Identity)

Event / Action Types Used

Field	Value
QueryType	EnumerateUsers
QueryType	QueryGroup
QueryType	QueryUser

Fields Used

Field	Table	Description
Timestamp	IdentityQueryEvents	Event time
AccountName	IdentityQueryEvents	Account performing the query
DeviceName	IdentityQueryEvents	Source device
QueryType	IdentityQueryEvents	Type of AD query
QueryTarget	IdentityQueryEvents	Target object
Protocol	IdentityQueryEvents	Protocol used

Query

```
IdentityQueryEvents
| where Timestamp > ago(7d)
| where QueryType in ("EnumerateUsers", "QueryGroup", "QueryUser")
| summarize Queries = count(), Targets = dcount(QueryTarget),
            TargetSample = make_set(QueryTarget, 25)
            by AccountName, DeviceName, QueryType, bin(Timestamp, 1h)
| where Targets >= 20
| sort by Targets desc
```

Expected Output

- **Returned records:** Accounts enumerating many distinct AD objects in a short window.
- **Detection value:** Bulk directory enumeration is classic pre-lateral-movement reconnaissance.
- **Investigation value:** Review the source device and account; correlate with subsequent logon / process activity.

Performance Notes

- **High volume table:** No
- **Recommended time range:** 7 days
- **Optimization:** Filter on the documented QueryType values first; tune the Targets threshold.

Schema Verification

Table	Verified
IdentityQueryEvents	Yes
Field	**Verified**
Timestamp	Yes
AccountName	Yes
DeviceName	Yes
QueryType	Yes
QueryTarget	Yes
Protocol	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
IdentityQueryEvents	Moderate	AD reconnaissance hunting

Query 28 — Sensitive Privileges Assigned at Logon (4672)

Security Use Case: Track accounts receiving sensitive (administrator-equivalent) privileges at logon, to baseline and spot unexpected privileged use.

Supported Platform: Microsoft Sentinel · Azure Monitor / Log Analytics

Tables Used

Table	Purpose
SecurityEvent	Windows Security event log via AMA / Log Analytics

Event / Action Types Used

Field	Value
EventID	4672 (special privileges assigned to new logon)

Fields Used

Field	Table	Description
TimeGenerated	SecurityEvent	Event time
Computer	SecurityEvent	Host
EventID	SecurityEvent	Windows event identifier
Account	SecurityEvent	Privileged account
Activity	SecurityEvent	Event activity description

Query

```
SecurityEvent
| where TimeGenerated > ago(7d)
| where EventID == 4672
| summarize PrivLogons = count(), Hosts = make_set(Computer, 20)
            by Account, bin(TimeGenerated, 1d)
| sort by PrivLogons desc
```

Expected Output

- **Returned records:** Accounts granted sensitive privileges at logon, by day and host set.

- **Detection value:** Unexpected accounts appearing with admin privileges may indicate escalation.
- **Investigation value:** Compare against the known privileged-account inventory; investigate outliers.

Performance Notes

- **High volume table:** Yes
- **Recommended time range:** 7 days
- **Optimization:** Very common event — summarize to reduce volume; filter EventID first.

Schema Verification

Table	Verified
SecurityEvent	Yes
Field	**Verified**
TimeGenerated	Yes
Computer	Yes
EventID	Yes
Account	Yes
Activity	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
SecurityEvent	Very High	Privileged account monitoring

Query 29 — Account Added to a Privileged Group (4728 / 4732 / 4756)

Security Use Case: Detect membership additions to security groups (global, local, universal) used for privilege escalation.

Supported Platform: Microsoft Sentinel · Azure Monitor / Log Analytics

Tables Used

Table	Purpose
SecurityEvent	Windows Security event log via AMA / Log Analytics

Event / Action Types Used

Field	Value
EventID	4728 (member added to global group)
EventID	4732 (member added to local group)
EventID	4756 (member added to universal group)

Fields Used

Field	Table	Description
TimeGenerated	SecurityEvent	Event time
Computer	SecurityEvent	Host
EventID	SecurityEvent	Windows event identifier
Activity	SecurityEvent	Event description
SubjectAccount	SecurityEvent	Actor performing the change

MemberName	SecurityEvent	Account added
TargetAccount	SecurityEvent	Group modified

Query

```
SecurityEvent
| where TimeGenerated > ago(7d)
| where EventID in (4728, 4732, 4756)
| project TimeGenerated, Computer, EventID, Activity,
          Actor = SubjectAccount, AddedMember = MemberName, Group = TargetAccount
| sort by TimeGenerated desc
```

Expected Output

- **Returned records:** Group-membership additions with actor, added member and target group.
- **Detection value:** Adds to Administrators / Domain Admins are direct escalation indicators.
- **Investigation value:** Verify the change is authorized; review actor activity around the event.

Performance Notes

- **High volume table:** Yes
- **Recommended time range:** 7 days
- **Optimization:** Filter the three EventIDs first; project documented columns only.

Schema Verification

Table	Verified
SecurityEvent	Yes
Field	**Verified**
TimeGenerated	Yes
Computer	Yes
EventID	Yes
Activity	Yes
SubjectAccount	Yes
MemberName	Yes
TargetAccount	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
SecurityEvent	Very High	Privilege escalation hunting

Query 30 — PsExec-Style Remote Execution

Security Use Case: Detect PsExec / PAExec service execution patterns indicative of remote command execution and lateral movement.

Supported Platform: Microsoft Defender XDR

Tables Used

Table	Purpose
DeviceProcessEvents	Process execution telemetry

Event / Action Types Used

Field	Value
ActionType	ProcessCreated

Fields Used

Field	Table	Description
Timestamp	DeviceProcessEvents	Event time
DeviceName	DeviceProcessEvents	Host
AccountName	DeviceProcessEvents	Account
FileName	DeviceProcessEvents	Process image
ProcessCommandLine	DeviceProcessEvents	Command line
InitiatingProcessFileName	DeviceProcessEvents	Parent process

Query

```
DeviceProcessEvents
| where Timestamp > ago(7d)
| where FileName =~ "psexesvc.exe"
    or ProcessCommandLine has_any ("psexec", "paexec", "-accepteula")
| project Timestamp, DeviceName, AccountName, FileName,
          ProcessCommandLine, InitiatingProcessFileName
| sort by Timestamp desc
```

Expected Output

- **Returned records:** Hosts running PsExec service binaries or invoking PsExec-style command lines.
- **Detection value:** PsExec is a dual-use admin tool heavily abused for lateral movement.
- **Investigation value:** Identify source and target hosts; confirm whether usage is sanctioned admin activity.

Performance Notes

- **High volume table:** Yes
- **Recommended time range:** 7 days
- **Optimization:** Combine the service filename and command-line indicators with has_any.

Schema Verification

Table	Verified
DeviceProcessEvents	Yes
Field	**Verified**
Timestamp	Yes
DeviceName	Yes
AccountName	Yes
FileName	Yes
ProcessCommandLine	Yes
InitiatingProcessFileName	Yes
ActionType	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
DeviceProcessEvents	Very High	Lateral movement hunting

Query 31 — Remote Service Creation via sc.exe

Security Use Case: Detect sc.exe creating or configuring services, optionally against remote hosts — a persistence / lateral-execution technique.

Supported Platform: Microsoft Defender XDR

Tables Used

Table	Purpose
DeviceProcessEvents	Process execution telemetry

Event / Action Types Used

Field	Value
ActionType	ProcessCreated

Fields Used

Field	Table	Description
Timestamp	DeviceProcessEvents	Event time
DeviceName	DeviceProcessEvents	Host
AccountName	DeviceProcessEvents	Account
FileName	DeviceProcessEvents	Process image
ProcessCommandLine	DeviceProcessEvents	Command line
InitiatingProcessFileName	DeviceProcessEvents	Parent process

Query

```
DeviceProcessEvents
| where Timestamp > ago(7d)
| where FileName =~ "sc.exe"
| where ProcessCommandLine has_any ("create", "config", @"\\")
| project Timestamp, DeviceName, AccountName, ProcessCommandLine, InitiatingProcessFileName
| sort by Timestamp desc
```

Expected Output

- **Returned records:** sc.exe invocations creating / reconfiguring services, including remote (\\host) targets.
- **Detection value:** Service creation is a frequent persistence and remote-execution mechanism.
- **Investigation value:** Inspect the binPath / target; confirm against change records.

Performance Notes

- **High volume table:** Yes
- **Recommended time range:** 7 days
- **Optimization:** Filter on FileName first; the literal backslash term catches remote targets.

Schema Verification

Table	Verified
DeviceProcessEvents	Yes
Field	**Verified**
Timestamp	Yes
DeviceName	Yes

AccountName	Yes
FileName	Yes
ProcessCommandLine	Yes
InitiatingProcessFileName	Yes
ActionType	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
DeviceProcessEvents	Very High	Persistence / lateral movement hunting

Query 32 — Shadow Copy Deletion (Ransomware Precursor)

Security Use Case: Detect anti-recovery commands (delete shadow copies, disable boot recovery) that typically precede ransomware encryption.

Supported Platform: Microsoft Defender XDR

Tables Used

Table	Purpose
DeviceProcessEvents	Process execution telemetry

Event / Action Types Used

Field	Value
ActionType	ProcessCreated

Fields Used

Field	Table	Description
Timestamp	DeviceProcessEvents	Event time
DeviceName	DeviceProcessEvents	Host
AccountName	DeviceProcessEvents	Account
FileName	DeviceProcessEvents	Process image
ProcessCommandLine	DeviceProcessEvents	Command line
InitiatingProcessFileName	DeviceProcessEvents	Parent process

Query

```
DeviceProcessEvents
| where Timestamp > ago(7d)
| where (FileName in~ ("vssadmin.exe", "wmic.exe")
         and ProcessCommandLine has_any ("delete shadows", "shadowcopy delete"))
    or (FileName =~ "wbadmin.exe" and ProcessCommandLine has "delete")
    or (FileName =~ "bcdedit.exe"
        and ProcessCommandLine has_any ("recoveryenabled no", "bootstatuspolicy
ignoreallfailures"))
| project Timestamp, DeviceName, AccountName, FileName, ProcessCommandLine,
InitiatingProcessFileName
| sort by Timestamp desc
```

Expected Output

- **Returned records:** Commands that delete shadow copies / backup catalogs or disable Windows recovery.

- **Detection value:** These are near-universal ransomware anti-recovery steps (T1490).
- **Investigation value:** Treat as high priority; isolate the host immediately and look for encryption activity.

Performance Notes

- **High volume table:** Yes
- **Recommended time range:** 7 days
- **Optimization:** Group the filename + command-line conditions; very low false-positive rate.

Schema Verification

Table	Verified
DeviceProcessEvents	Yes
Field	**Verified**
Timestamp	Yes
DeviceName	Yes
AccountName	Yes
FileName	Yes
ProcessCommandLine	Yes
InitiatingProcessFileName	Yes
ActionType	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
DeviceProcessEvents	Very High	Ransomware hunting

Query 33 — Mass File Modification by a Single Process (Ransomware Encryption)

Security Use Case: Detect a single process modifying a very large number of files quickly — a behavioral ransomware-encryption signature.

Supported Platform: Microsoft Defender XDR

Tables Used

Table	Purpose
DeviceFileEvents	File system event telemetry

Event / Action Types Used

Field	Value
ActionType	FileModified

Fields Used

Field	Table	Description
Timestamp	DeviceFileEvents	Event time
DeviceName	DeviceFileEvents	Host
ActionType	DeviceFileEvents	File operation
FileName	DeviceFileEvents	Affected file
FolderPath	DeviceFileEvents	Affected folder

InitiatingProcessFileName	DeviceFileEvents	Modifying process
InitiatingProcessAccountName	DeviceFileEvents	Account

Query

```
DeviceFileEvents
| where Timestamp > ago(1d)
| where ActionType == "FileModified"
| summarize ModifiedFiles = count(), DistinctFolders = dcount(FolderPath),
            Sample = make_set(FileName, 10)
            by DeviceName, InitiatingProcessFileName, bin(Timestamp, 10m)
| where ModifiedFiles >= 200
| sort by ModifiedFiles desc
```

Expected Output

- **Returned records:** Processes modifying 200+ files within a 10-minute window, by host.
- **Detection value:** Rapid bulk modification across many folders is characteristic of encryption.
- **Investigation value:** Identify the process binary; isolate the host; preserve a sample for analysis.

Performance Notes

- **High volume table:** Yes
- **Recommended time range:** 24 hours
- **Optimization:** High-volume table — keep window short; tune the file-count threshold to the environment.

Schema Verification

Table	Verified
DeviceFileEvents	Yes
Field	**Verified**
Timestamp	Yes
DeviceName	Yes
ActionType	Yes
FileName	Yes
FolderPath	Yes
InitiatingProcessFileName	Yes
InitiatingProcessAccountName	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
DeviceFileEvents	Very High	Ransomware behavioral hunting

Query 34 — Archive Creation for Staging / Exfiltration

Security Use Case: Detect compression utilities packaging data, often used to stage files before exfiltration.

Supported Platform: Microsoft Defender XDR

Tables Used

Table	Purpose
DeviceProcessEvents	Process execution telemetry

Event / Action Types Used

Field	Value
ActionType	ProcessCreated

Fields Used

Field	Table	Description
Timestamp	DeviceProcessEvents	Event time
DeviceName	DeviceProcessEvents	Host
AccountName	DeviceProcessEvents	Account
FileName	DeviceProcessEvents	Process image
ProcessCommandLine	DeviceProcessEvents	Command line
InitiatingProcessFileName	DeviceProcessEvents	Parent process

Query

```
DeviceProcessEvents
| where Timestamp > ago(7d)
| where FileName in~ ("7z.exe", "7za.exe", "rar.exe", "winrar.exe", "tar.exe")
| where ProcessCommandLine has_any (" a ", "-hp", "-p", "-r ", "cf ")
| project Timestamp, DeviceName, AccountName, FileName, ProcessCommandLine,
InitiatingProcessFileName
| sort by Timestamp desc
```

Expected Output

- **Returned records:** Archiving tools invoked with add / password / recurse flags.
- **Detection value:** Bulk archiving (especially password-protected) frequently precedes data theft (T1560).
- **Investigation value:** Inspect the archive target path and size; correlate with subsequent outbound transfers.

Performance Notes

- **High volume table:** Yes
- **Recommended time range:** 7 days
- **Optimization:** Filter the tool set first; expect benign hits — prioritize password-protected archives.

Schema Verification

Table	Verified
DeviceProcessEvents	Yes
Field	**Verified**
Timestamp	Yes
DeviceName	Yes
AccountName	Yes
FileName	Yes
ProcessCommandLine	Yes
InitiatingProcessFileName	Yes
ActionType	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use

DeviceProcessEvents	Very High	Exfiltration staging hunting

Query 35 — Connections to Cloud Storage / File-Sharing Sites

Security Use Case: Surface endpoint connections to consumer file-sharing services commonly abused for data exfiltration.

Supported Platform: Microsoft Defender XDR

Tables Used

Table	Purpose
DeviceNetworkEvents	Network connection telemetry

Event / Action Types Used

Field	Value
ActionType	ConnectionSuccess

Fields Used

Field	Table	Description
Timestamp	DeviceNetworkEvents	Event time
DeviceName	DeviceNetworkEvents	Host
RemoteUrl	DeviceNetworkEvents	Destination URL/FQDN
RemoteIP	DeviceNetworkEvents	Destination IP
InitiatingProcessFileName	DeviceNetworkEvents	Initiating process
InitiatingProcessCommandLine	DeviceNetworkEvents	Process command line

Query

```
DeviceNetworkEvents
| where Timestamp > ago(7d)
| where ActionType == "ConnectionSuccess"
| where RemoteUrl has_any
        ("dropbox", "mega.nz", "mega.io", "anonfiles", "transfer.sh",
         "wetransfer", "pastebin", "gofile")
| project Timestamp, DeviceName, RemoteUrl, RemoteIP,
          InitiatingProcessFileName, InitiatingProcessCommandLine
| sort by Timestamp desc
```

Expected Output

- **Returned records:** Connections to consumer file-sharing / paste services with the initiating process.
- **Detection value:** These destinations are frequent exfiltration channels (T1567).
- **Investigation value:** Heuristic — validate against sanctioned business use before alerting.

Performance Notes

- **High volume table:** Yes
- **Recommended time range:** 7 days
- **Optimization:** Term-indexed has_any on RemoteUrl; maintain an allowlist of approved services.

Schema Verification

Table	Verified

DeviceNetworkEvents	Yes
Field	**Verified**
Timestamp	Yes
DeviceName	Yes
RemoteUrl	Yes
RemoteIP	Yes
InitiatingProcessFileName	Yes
InitiatingProcessCommandLine	Yes
ActionType	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
DeviceNetworkEvents	High	Exfiltration channel hunting

Query 36 — Malicious URLs in Delivered Email

Security Use Case: Join phishing / malware email verdicts to the URLs they contained for click-risk triage.

Supported Platform: Microsoft Defender XDR

Tables Used

Table	Purpose
EmailEvents	Email delivery and verdict telemetry
EmailUrlInfo	URLs extracted from email

Event / Action Types Used

Field	Value
ThreatTypes	Phish
ThreatTypes	Malware

Fields Used

Field	Table	Description
Timestamp	EmailEvents	Event time
NetworkMessageId	EmailEvents	Message correlation key
SenderFromAddress	EmailEvents	Sender
RecipientEmailAddress	EmailEvents	Recipient
Subject	EmailEvents	Subject
ThreatTypes	EmailEvents	Verdict
Url	EmailUrlInfo	Embedded URL
UrlDomain	EmailUrlInfo	URL domain

Query

```
EmailEvents
| where Timestamp > ago(7d)
| where ThreatTypes has "Phish" or ThreatTypes has "Malware"
| join kind=inner (EmailUrlInfo | where Timestamp > ago(7d)) on NetworkMessageId
| project Timestamp, SenderFromAddress, RecipientEmailAddress, Subject,
```

```
          Url, UrlDomain, ThreatTypes
| sort by Timestamp desc
```

Expected Output

- **Returned records:** Malicious emails joined to their embedded URLs and domains.
- **Detection value:** Exposes the destinations recipients may have clicked.
- **Investigation value:** Block the domains; check UrlClickEvents or proxy logs for actual clicks.

Performance Notes

- **High volume table:** Yes
- **Recommended time range:** 7 days
- **Optimization:** Filter verdicts on EmailEvents before the join; keep the join window aligned.

Schema Verification

Table	Verified
EmailEvents	Yes
EmailUrlInfo	Yes
Field	**Verified**
Timestamp	Yes
NetworkMessageId	Yes
SenderFromAddress	Yes
RecipientEmailAddress	Yes
Subject	Yes
ThreatTypes	Yes
Url	Yes
UrlDomain	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
EmailEvents	High	Phishing hunting
EmailUrlInfo	High	URL threat correlation

Query 37 — Inbound Email Failing Sender Authentication (Spoofing)

Security Use Case: Find inbound mail that failed SPF or DMARC, a common indicator of sender spoofing / impersonation.

Supported Platform: Microsoft Defender XDR

Tables Used

Table	Purpose
EmailEvents	Email delivery and verdict telemetry

Event / Action Types Used

No specific event type filter used.

Fields Used

Field	Table	Description
Timestamp	EmailEvents	Event time
EmailDirection	EmailEvents	Inbound / Outbound
SenderFromAddress	EmailEvents	Header From address
SenderMailFromAddress	EmailEvents	Envelope From
RecipientEmailAddress	EmailEvents	Recipient
Subject	EmailEvents	Subject
AuthenticationDetails	EmailEvents	SPF / DKIM / DMARC results

Query

```
EmailEvents
| where Timestamp > ago(7d)
| where EmailDirection == "Inbound"
| extend Auth = parse_json(AuthenticationDetails)
| where tostring(Auth.SPF) == "fail" or tostring(Auth.DMARC) == "fail"
| project Timestamp, SenderFromAddress, SenderMailFromAddress, RecipientEmailAddress,
          Subject, SPF = tostring(Auth.SPF), DKIM = tostring(Auth.DKIM), DMARC =
tostring(Auth.DMARC)
| sort by Timestamp desc
```

Expected Output

- **Returned records:** Inbound messages with failing SPF or DMARC verdicts.
- **Detection value:** Authentication failures are a strong spoofing / BEC indicator.
- **Investigation value:** Compare header vs envelope sender; prioritize messages targeting executives / finance.

Performance Notes

- **High volume table:** Yes
- **Recommended time range:** 7 days
- **Optimization:** Filter EmailDirection first; parse AuthenticationDetails only on the reduced set.

Schema Verification

Table	Verified
EmailEvents	Yes
Field	**Verified**
Timestamp	Yes
EmailDirection	Yes
SenderFromAddress	Yes
SenderMailFromAddress	Yes
RecipientEmailAddress	Yes
Subject	Yes
AuthenticationDetails	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
EmailEvents	High	Spoofing / BEC hunting

Query 38 — Mailbox Inbox Rule Creation (BEC)

Security Use Case: Detect creation / modification of mailbox inbox rules, frequently used in business email compromise to hide replies.

Supported Platform: Microsoft Sentinel · Azure Monitor / Log Analytics

Tables Used

Table	Purpose
OfficeActivity	Microsoft 365 unified audit activity

Event / Action Types Used

Field	Value
Operation	New-InboxRule
Operation	Set-InboxRule
Operation	UpdateInboxRules

Fields Used

Field	Table	Description
TimeGenerated	OfficeActivity	Event time
UserId	OfficeActivity	Acting user
ClientIP	OfficeActivity	Source IP
Operation	OfficeActivity	Audited operation
OfficeWorkload	OfficeActivity	Workload (Exchange)
Parameters	OfficeActivity	Operation parameters

Query

```
OfficeActivity
| where TimeGenerated > ago(7d)
| where Operation in ("New-InboxRule", "Set-InboxRule", "UpdateInboxRules")
| project TimeGenerated, UserId, ClientIP, Operation, OfficeWorkload, Parameters
| sort by TimeGenerated desc
```

Expected Output

- **Returned records:** Inbox-rule create / update operations with the acting user and source IP.
- **Detection value:** Rules that delete or move incoming mail are a hallmark of BEC.
- **Investigation value:** Inspect rule parameters for forwarding / delete / move-to-folder actions.

Performance Notes

- **High volume table:** No
- **Recommended time range:** 7 days
- **Optimization:** Filter on the documented Operation values first.

Schema Verification

Table	Verified
OfficeActivity	Yes
Field	**Verified**
TimeGenerated	Yes
UserId	Yes

ClientIP	Yes
Operation	Yes
OfficeWorkload	Yes
Parameters	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
OfficeActivity	Moderate	BEC / mailbox-rule hunting

Query 39 — Mailbox Forwarding Configured

Security Use Case: Detect SMTP forwarding configured on mailboxes, a persistent data-exfiltration mechanism in account compromise.

Supported Platform: Microsoft Sentinel · Azure Monitor / Log Analytics

Tables Used

Table	Purpose
OfficeActivity	Microsoft 365 unified audit activity

Event / Action Types Used

Field	Value
Operation	Set-Mailbox
Operation	Set-MailboxAutoReplyConfiguration

Fields Used

Field	Table	Description
TimeGenerated	OfficeActivity	Event time
UserId	OfficeActivity	Acting user
ClientIP	OfficeActivity	Source IP
Operation	OfficeActivity	Audited operation
Parameters	OfficeActivity	Operation parameters

Query

```
OfficeActivity
| where TimeGenerated > ago(14d)
| where Operation in ("Set-Mailbox", "Set-MailboxAutoReplyConfiguration")
| where Parameters has_any ("ForwardingSmtpAddress", "ForwardingAddress",
"DeliverToMailboxAndForward")
| project TimeGenerated, UserId, ClientIP, Operation, Parameters
| sort by TimeGenerated desc
```

Expected Output

- **Returned records:** Mailbox operations that set forwarding addresses / behavior.
- **Detection value:** External forwarding silently exfiltrates a victim's mail stream.
- **Investigation value:** Verify forwarding is sanctioned; check whether the target is an external domain.

Performance Notes

- **High volume table:** No

- **Recommended time range:** 14 days
- **Optimization:** Filter Operation first, then match forwarding parameters with has_any.

Schema Verification

Table	Verified
OfficeActivity	Yes
Field	**Verified**
TimeGenerated	Yes
UserId	Yes
ClientIP	Yes
Operation	Yes
Parameters	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
OfficeActivity	Moderate	Exfiltration / BEC hunting

Query 40 — Storage Account Key Listing

Security Use Case: Detect listKeys operations on storage accounts — retrieving access keys can enable data access / exfiltration.

Supported Platform: Microsoft Sentinel · Azure Monitor / Log Analytics

Tables Used

Table	Purpose
AzureActivity	Azure Resource Manager control-plane activity

Event / Action Types Used

No specific event type filter used.

Fields Used

Field	Table	Description
TimeGenerated	AzureActivity	Event time
OperationNameValue	AzureActivity	ARM operation
ActivityStatusValue	AzureActivity	Operation status
Caller	AzureActivity	Acting identity
CallerIpAddress	AzureActivity	Source IP
ResourceProviderValue	AzureActivity	Resource provider
ResourceGroup	AzureActivity	Resource group
_ResourceId	AzureActivity	Target resource

Query

```
AzureActivity
| where TimeGenerated > ago(7d)
| where ResourceProviderValue =~ "Microsoft.Storage"
| where OperationNameValue endswith "listKeys/action"
```

```
| project TimeGenerated, OperationNameValue, ActivityStatusValue,
          Caller, CallerIpAddress, ResourceGroup, _ResourceId
| sort by TimeGenerated desc
```

Expected Output

- **Returned records:** Storage account key-listing operations with caller and source IP.
- **Detection value:** Unexpected key retrieval can be a precursor to bulk data access.
- **Investigation value:** Correlate the caller with sign-in activity and known automation principals.

Performance Notes

- **High volume table:** No
- **Recommended time range:** 7 days
- **Optimization:** Filter the provider and operation suffix first.

Schema Verification

Table	Verified
AzureActivity	Yes
Field	**Verified**
TimeGenerated	Yes
OperationNameValue	Yes
ActivityStatusValue	Yes
Caller	Yes
CallerIpAddress	Yes
ResourceProviderValue	Yes
ResourceGroup	Yes
_ResourceId	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
AzureActivity	Moderate	Cloud data-access hunting

Query 41 — Network Security Group Rule Changes

Security Use Case: Detect creation / modification / deletion of NSGs, which can open paths for attacker access or exfiltration.

Supported Platform: Microsoft Sentinel · Azure Monitor / Log Analytics

Tables Used

Table	Purpose
AzureActivity	Azure Resource Manager control-plane activity

Event / Action Types Used

No specific event type filter used.

Fields Used

Field	Table	Description

TimeGenerated	AzureActivity	Event time
OperationNameValue	AzureActivity	ARM operation
ActivityStatusValue	AzureActivity	Operation status
Caller	AzureActivity	Acting identity
CallerIpAddress	AzureActivity	Source IP
ResourceProviderValue	AzureActivity	Resource provider
ResourceGroup	AzureActivity	Resource group
_ResourceId	AzureActivity	Target resource

Query

```
AzureActivity
| where TimeGenerated > ago(7d)
| where ResourceProviderValue =~ "Microsoft.Network"
| where OperationNameValue has "networkSecurityGroups"
| where OperationNameValue has_any ("write", "delete")
| project TimeGenerated, OperationNameValue, ActivityStatusValue,
          Caller, CallerIpAddress, ResourceGroup, _ResourceId
| sort by TimeGenerated desc
```

Expected Output

- **Returned records:** NSG write / delete operations with the acting identity.
- **Detection value:** Firewall-rule changes can expose management ports or disable controls.
- **Investigation value:** Validate against change tickets; review the resulting rule set for risky openings.

Performance Notes

- **High volume table:** No
- **Recommended time range:** 7 days
- **Optimization:** Filter provider first, then narrow to NSG write / delete operations.

Schema Verification

Table	Verified
AzureActivity	Yes
Field	**Verified**
TimeGenerated	Yes
OperationNameValue	Yes
ActivityStatusValue	Yes
Caller	Yes
CallerIpAddress	Yes
ResourceProviderValue	Yes
ResourceGroup	Yes
_ResourceId	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
AzureActivity	Moderate	Cloud network-control monitoring

Query 42 — Azure RBAC Role Assignment Changes

Security Use Case: Detect new role assignments granting Azure RBAC permissions — a cloud privilege-escalation / persistence step.

Supported Platform: Microsoft Sentinel · Azure Monitor / Log Analytics

Tables Used

Table	Purpose
AzureActivity	Azure Resource Manager control-plane activity

Event / Action Types Used

No specific event type filter used.

Fields Used

Field	Table	Description
TimeGenerated	AzureActivity	Event time
OperationNameValue	AzureActivity	ARM operation
ActivityStatusValue	AzureActivity	Operation status
Caller	AzureActivity	Acting identity
CallerIpAddress	AzureActivity	Source IP
ResourceGroup	AzureActivity	Resource group
_ResourceId	AzureActivity	Target resource

Query

```
AzureActivity
| where TimeGenerated > ago(7d)
| where OperationNameValue has "Microsoft.Authorization/roleAssignments"
| where ActivityStatusValue == "Success"
| project TimeGenerated, OperationNameValue, Caller, CallerIpAddress, ResourceGroup,
_ResourceId
| sort by TimeGenerated desc
```

Expected Output

- **Returned records:** Successful RBAC role-assignment operations with caller and scope.
- **Detection value:** Assigning Owner / Contributor to unexpected principals is a key escalation signal.
- **Investigation value:** Confirm the assignment is authorized; review the assigned principal and scope.

Performance Notes

- **High volume table:** No
- **Recommended time range:** 7 days
- **Optimization:** Filter on the roleAssignments operation and success status first.

Schema Verification

Table	Verified
AzureActivity	Yes
Field	**Verified**
TimeGenerated	Yes

OperationNameValue	Yes
ActivityStatusValue	Yes
Caller	Yes
CallerIpAddress	Yes
ResourceGroup	Yes
_ResourceId	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
AzureActivity	Moderate	Cloud privilege escalation hunting

Query 43 — Bulk Azure Resource Deletion

Security Use Case: Detect a single caller deleting many Azure resources in a short window — possible destruction or sabotage.

Supported Platform: Microsoft Sentinel · Azure Monitor / Log Analytics

Tables Used

Table	Purpose
AzureActivity	Azure Resource Manager control-plane activity

Event / Action Types Used

No specific event type filter used.

Fields Used

Field	Table	Description
TimeGenerated	AzureActivity	Event time
OperationNameValue	AzureActivity	ARM operation
ActivityStatusValue	AzureActivity	Operation status
Caller	AzureActivity	Acting identity
CallerIpAddress	AzureActivity	Source IP
_ResourceId	AzureActivity	Target resource

Query

```
AzureActivity
| where TimeGenerated > ago(1d)
| where OperationNameValue endswith "/delete"
| where ActivityStatusValue == "Success"
| summarize Deletions = count(), Resources = make_set(_ResourceId, 30)
            by Caller, CallerIpAddress, bin(TimeGenerated, 1h)
| where Deletions >= 10
| sort by Deletions desc
```

Expected Output

- **Returned records:** Callers performing 10+ successful resource deletions per hour.
- **Detection value:** Mass deletion can indicate destructive action (T1485) or a compromised automation principal.
- **Investigation value:** Verify against planned decommissioning; review the caller's recent activity.

Performance Notes

- **High volume table:** No
- **Recommended time range:** 24 hours
- **Optimization:** Filter delete operations and success first; tune the deletion threshold.

Schema Verification

Table	Verified
AzureActivity	Yes
Field	**Verified**
TimeGenerated	Yes
OperationNameValue	Yes
ActivityStatusValue	Yes
Caller	Yes
CallerIpAddress	Yes
_ResourceId	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
AzureActivity	Moderate	Destructive-action / sabotage hunting

Query 44 — VM / Agent Heartbeat Gap (Silent Hosts)

Security Use Case: Identify monitored machines that have stopped sending heartbeats — agent failure, host down, or evasion.

Supported Platform: Microsoft Sentinel · Azure Monitor / Log Analytics

Tables Used

Table	Purpose
Heartbeat	Log Analytics agent heartbeat telemetry

Event / Action Types Used

No specific event type filter used.

Fields Used

Field	Table	Description
TimeGenerated	Heartbeat	Heartbeat time
Computer	Heartbeat	Machine name
OSType	Heartbeat	Operating system type

Query

```
Heartbeat
| where TimeGenerated > ago(1d)
| summarize LastHeartbeat = max(TimeGenerated) by Computer, OSType
| where LastHeartbeat < ago(1h)
| extend SilentFor = now() - LastHeartbeat
| sort by SilentFor desc
```

Expected Output

- **Returned records:** Machines whose most recent heartbeat is older than one hour.
- **Detection value:** Coverage gaps can hide compromise or indicate deliberate agent tampering.
- **Investigation value:** Confirm whether the host is intentionally offline; investigate unexpected silence.

Performance Notes

- **High volume table:** No
- **Recommended time range:** 24 hours
- **Optimization:** Lightweight aggregation; tune the silence threshold to your heartbeat interval.

Schema Verification

Table	Verified
Heartbeat	Yes
Field	**Verified**
TimeGenerated	Yes
Computer	Yes
OSType	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
Heartbeat	Low	Monitoring coverage / VM health

Query 45 — High-Severity Defender Alert Triage

Security Use Case: Summarize high / medium severity Defender alerts by category and source for daily triage.

Supported Platform: Microsoft Defender XDR

Tables Used

Table	Purpose
AlertInfo	Defender alert metadata

Event / Action Types Used

Field	Value
Severity	High
Severity	Medium

Fields Used

Field	Table	Description
Timestamp	AlertInfo	Alert time
AlertId	AlertInfo	Alert identifier
Title	AlertInfo	Alert title
Category	AlertInfo	Threat category
Severity	AlertInfo	Severity
ServiceSource	AlertInfo	Originating service

Query

```
AlertInfo
| where Timestamp > ago(7d)
| where Severity in ("High", "Medium")
| summarize Alerts = count(), Titles = make_set(Title, 15)
            by Category, ServiceSource, Severity
| sort by Alerts desc
```

Expected Output

- **Returned records:** Alert counts grouped by category, originating service and severity.
- **Detection value:** Gives a triage view of where high-severity detections concentrate.
- **Investigation value:** Drill from a category into individual AlertIds for full investigation.

Performance Notes

- **High volume table:** No
- **Recommended time range:** 7 days
- **Optimization:** Filter severity first; AlertInfo is comparatively low volume.

Schema Verification

Table	Verified
AlertInfo	Yes
Field	**Verified**
Timestamp	Yes
AlertId	Yes
Title	Yes
Category	Yes
Severity	Yes
ServiceSource	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
AlertInfo	Low	Alert triage

Query 46 — Alert Evidence Entity Pivot

Security Use Case: Expand high-severity alerts into their associated entities (users, IPs, URLs, files) for rapid scoping.

Supported Platform: Microsoft Defender XDR

Tables Used

Table	Purpose
AlertInfo	Defender alert metadata
AlertEvidence	Entities associated with alerts

Event / Action Types Used

Field	Value
Severity	High

Fields Used

Field	Table	Description
Timestamp	AlertInfo	Alert time
AlertId	AlertInfo	Alert identifier
Title	AlertInfo	Alert title
Category	AlertInfo	Threat category
EntityType	AlertEvidence	Entity type
EvidenceRole	AlertEvidence	Role of the evidence
AccountName	AlertEvidence	Account entity
RemoteIP	AlertEvidence	IP entity
RemoteUrl	AlertEvidence	URL entity
FileName	AlertEvidence	File entity
SHA256	AlertEvidence	File hash

Query

```
AlertInfo
| where Timestamp > ago(7d)
| where Severity == "High"
| join kind=inner (AlertEvidence | where Timestamp > ago(7d)) on AlertId
| where EntityType in ("User", "Ip", "Url", "File")
| project Timestamp, Title, Category, EntityType, EvidenceRole,
          AccountName, RemoteIP, RemoteUrl, FileName, SHA256
| sort by Timestamp desc
```

Expected Output

- **Returned records:** High-severity alerts joined to their constituent entities.
- **Detection value:** Accelerates blast-radius scoping (which users / hosts / IOCs are involved).
- **Investigation value:** Use the entity values as pivots into device, identity and email tables.

Performance Notes

- **High volume table:** No
- **Recommended time range:** 7 days
- **Optimization:** Filter severity on AlertInfo before the join; restrict to the entity types of interest.

Schema Verification

Table	Verified
AlertInfo	Yes
AlertEvidence	Yes
Field	**Verified**
Timestamp	Yes
AlertId	Yes
Title	Yes
Category	Yes
EntityType	Yes
EvidenceRole	Yes
AccountName	Yes
RemoteIP	Yes

RemoteUrl	Yes
FileName	Yes
SHA256	Yes
Severity	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
AlertInfo	Low	Alert triage
AlertEvidence	Low	Entity / IOC correlation

Query 47 — Sentinel Alerts by MITRE Tactic

Security Use Case: Aggregate Microsoft Sentinel security alerts by MITRE ATT&CK tactic to see where coverage and activity concentrate.

Supported Platform: Microsoft Sentinel · Azure Monitor / Log Analytics

Tables Used

Table	Purpose
SecurityAlert	Sentinel security alerts

Event / Action Types Used

No specific event type filter used.

Fields Used

Field	Table	Description
TimeGenerated	SecurityAlert	Alert time
AlertName	SecurityAlert	Alert name
AlertSeverity	SecurityAlert	Severity
ProductName	SecurityAlert	Detecting product
Tactics	SecurityAlert	MITRE ATT&CK tactics

Query

```
SecurityAlert
| where TimeGenerated > ago(7d)
| mv-expand Tactic = todynamic(Tactics)
| summarize Alerts = count(), Products = make_set(ProductName, 10)
            by Tactic = tostring(Tactic), AlertSeverity
| sort by Alerts desc
```

Expected Output

- **Returned records:** Alert counts per MITRE tactic and severity, with contributing products.
- **Detection value:** Reveals which attack stages are generating the most detections.
- **Investigation value:** Drill into a tactic to review the specific AlertNames and affected entities.

Performance Notes

- **High volume table:** No
- **Recommended time range:** 7 days

- **Optimization:** todynamic handles both JSON-array and single-value Tactics; mv-expand after filtering.

Schema Verification

Table	Verified
SecurityAlert	Yes
Field	**Verified**
TimeGenerated	Yes
AlertName	Yes
AlertSeverity	Yes
ProductName	Yes
Tactics	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
SecurityAlert	Low	Alert analytics / coverage review

Query 48 — Off-Hours Privileged Sign-ins

Security Use Case: Detect successful sign-ins to administrative apps outside normal business hours — an insider / takeover heuristic.

Supported Platform: Microsoft Sentinel · Azure Monitor / Log Analytics

Tables Used

Table	Purpose
SigninLogs	Entra ID sign-in telemetry

Event / Action Types Used

Field	Value
ResultType	0 (successful sign-in)

Fields Used

Field	Table	Description
TimeGenerated	SigninLogs	Event time (UTC)
UserPrincipalName	SigninLogs	Account
AppDisplayName	SigninLogs	Target application
IPAddress	SigninLogs	Source IP
Location	SigninLogs	Geo of source IP
ResultType	SigninLogs	Sign-in result code

Query

```
SigninLogs
| where TimeGenerated > ago(7d)
| where ResultType == 0
| extend Hour = datetime_part("Hour", TimeGenerated)
| where Hour < 6 or Hour >= 22
| where AppDisplayName has_any
        ("Azure Portal", "PowerShell", "Microsoft Graph", "Command Line Interface")
```

```
| project TimeGenerated, UserPrincipalName, AppDisplayName, IPAddress, Location, Hour
| sort by TimeGenerated desc
```

Expected Output

- **Returned records:** Successful admin-tooling sign-ins occurring late night / early morning (UTC).
- **Detection value:** Off-hours administrative access is a common insider-threat / takeover heuristic.
- **Investigation value:** Adjust the hour window to local time zone; correlate with the user's normal pattern.

Performance Notes

- **High volume table:** Yes
- **Recommended time range:** 7 days
- **Optimization:** Hours are UTC — shift the window to your time zone; filter success first.

Schema Verification

Table	Verified
SigninLogs	Yes
Field	**Verified**
TimeGenerated	Yes
UserPrincipalName	Yes
AppDisplayName	Yes
IPAddress	Yes
Location	Yes
ResultType	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
SigninLogs	High	Insider threat / anomaly hunting

Query 49 — Mass File Downloads from SharePoint / OneDrive

Security Use Case: Detect users downloading an unusually large number of files — potential insider data theft or compromised account staging.

Supported Platform: Microsoft Sentinel · Azure Monitor / Log Analytics

Tables Used

Table	Purpose
OfficeActivity	Microsoft 365 unified audit activity

Event / Action Types Used

Field	Value
Operation	FileDownloaded

Fields Used

Field	Table	Description
TimeGenerated	OfficeActivity	Event time
UserId	OfficeActivity	Acting user

ClientIP	OfficeActivity	Source IP
Operation	OfficeActivity	Audited operation
OfficeObjectId	OfficeActivity	Downloaded object

Query

```
OfficeActivity
| where TimeGenerated > ago(1d)
| where Operation == "FileDownloaded"
| summarize Downloads = count(), Files = dcount(OfficeObjectId), IPs = make_set(ClientIP,
10)
            by UserId, bin(TimeGenerated, 1h)
| where Downloads >= 100
| sort by Downloads desc
```

Expected Output

- **Returned records:** Users downloading 100+ files within an hour, with source IPs.
- **Detection value:** Bulk download spikes can indicate data theft before departure or after compromise.
- **Investigation value:** Compare against the user's baseline; review the files and source location.

Performance Notes

- **High volume table:** No
- **Recommended time range:** 24 hours
- **Optimization:** Filter Operation first; tune the download threshold to normal usage.

Schema Verification

Table	Verified
OfficeActivity	Yes
Field	**Verified**
TimeGenerated	Yes
UserId	Yes
ClientIP	Yes
Operation	Yes
OfficeObjectId	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
OfficeActivity	Moderate	Insider threat / data theft hunting

Query 50 — Mass File Deletion by a User (Destruction / Insider)

Security Use Case: Detect an account deleting a very large number of files quickly — possible sabotage or destructive insider action.

Supported Platform: Microsoft Defender XDR

Tables Used

Table	Purpose
DeviceFileEvents	File system event telemetry

Event / Action Types Used

Field	Value
ActionType	FileDeleted

Fields Used

Field	Table	Description
Timestamp	DeviceFileEvents	Event time
DeviceName	DeviceFileEvents	Host
ActionType	DeviceFileEvents	File operation
FolderPath	DeviceFileEvents	Affected folder
InitiatingProcessAccountName	DeviceFileEvents	Acting account

Query

```
DeviceFileEvents
| where Timestamp > ago(1d)
| where ActionType == "FileDeleted"
| summarize Deletions = count(), Folders = dcount(FolderPath)
            by InitiatingProcessAccountName, DeviceName, bin(Timestamp, 30m)
| where Deletions >= 200
| sort by Deletions desc
```

Expected Output

- **Returned records:** Accounts deleting 200+ files within a 30-minute window, by host.
- **Detection value:** Mass deletion can indicate destructive insider activity or wiper behavior (T1485).
- **Investigation value:** Identify the account and host; determine whether deletion was authorized.

Performance Notes

- **High volume table:** Yes
- **Recommended time range:** 24 hours
- **Optimization:** High-volume table — keep window short; tune the deletion threshold.

Schema Verification

Table	Verified
DeviceFileEvents	Yes
Field	**Verified**
Timestamp	Yes
DeviceName	Yes
ActionType	Yes
FolderPath	Yes
InitiatingProcessAccountName	Yes

Table Usage Summary

Table	Typical Daily Volume	Investigation Use
DeviceFileEvents	Very High	Destruction / insider hunting