|  |
| --- |
| Azure Sentinel  **SAP Continuous Threat Monitoring** |
| **NDA Private Preview** |Schema & Analytics 31.1.2021  Microsoft Preliminary and Confidential, under NDA Questions? Yoav.Daniely@microsoft.com |

# Logs Overview

## ABAP Security Audit Log

The main objective of the audit log is to record the following:

* Security-related changes to the SAP system environment (for example, changes to user master records)
* Information that provides a higher level of transparency (for example, successful and unsuccessful logon attempts)
* Information that enables the reconstruction of a series of events (for example, successful or unsuccessful transaction starts).

Available by using RFC XAL/SAL interfaces. SAL is available as of Basis 7.50.

SAP Documentation: [LINK](https://help.sap.com/viewer/280f016edb8049e998237fcbd80558e7/7.5.7/en-US/4d41bec4aa601c86e10000000a42189b.html)

## ABAP Change Documents Log

SAP NetWeaver Application Server (AS) ABAP logs changes to business data objects in change documents. Apart from business data, it also logs other entities in the SAP system such as user data, roles, addresses.

Available by using RFC with a custom service based on standard services.

SAP Documentation: [LINK](https://help.sap.com/viewer/6f51f5216c4b10149358d088a0b7029c/7.01.22/en-US/b8686150ed102f1ae10000000a44176f.html)

## ABAP Spool Log

Main log of SAP Printing contains history of spool requests. (SP01).

Available by using RFC with a custom service based on standard tables.

SAP Documentation: [LINK](https://help.sap.com/viewer/290ce8983cbc4848a9d7b6f5e77491b9/7.52.1/en-US/4eae791c40f72045e10000000a421937.html)

## ABAP Spool Output Log

Main log of SAP Printing contains history of spool output requests. (SP02).

Available by using RFC with a custom service based on standard tables.

SAP Documentation: [LINK](https://help.sap.com/viewer/290ce8983cbc4848a9d7b6f5e77491b9/7.52.1/en-US/4eae779e40f72045e10000000a421937.html)

## ABAP Job Log

This job consolidates all background processing Job Logs (SM37).

Available by using RFC with a custom service based on standard services of XBP interfaces.

SAP Documentation: [LINK](https://help.sap.com/viewer/b07e7195f03f438b8e7ed273099d74f3/7.31.19/en-US/4b2bc0974c594ba2e10000000a42189c.html)

## ABAP Application Log

Application logging records the progress of the execution of an application so that you can reconstruct it later if necessary. Whereas the system log records system events, you can use the application log to record application-specific events.

Available by using RFC with a custom service based on standard services of XBP interface.

SAP Documentation: [LINK](https://help.sap.com/viewer/56bf1265a92e4b4d9a72448c579887af/7.5.7/en-US/c769bcc9f36611d3a6510000e835363f.html)

## ABAP Workflow Log

SAP Business Workflow (WebFlow Engine) makes it easy to define business processes that are not yet mapped in the SAP system. These may be simple release or approval procedures, or more complex business processes such as creating a material master and the associated coordination of the departments involved.

Available by using RFC with a custom service based on standard tables and standard services.

SAP Documentation: [LINK](https://help.sap.com/viewer/56bf1265a92e4b4d9a72448c579887af/7.5.7/en-US/c769bcccf36611d3a6510000e835363f.html)

## ABAP CR Log

In addition to application logging, change documents, and table recording, all changes that you make to your productive system using the Change & Transport System are documented in the CTS and TMS logs.

The CR Log covers the CTS logs – which directory objects and customizing were performed.

Available by using RFC with a custom service based on standard tables and standard services.

SAP Documentation: [LINK](https://help.sap.com/viewer/56bf1265a92e4b4d9a72448c579887af/7.5.7/en-US/c769bcd5f36611d3a6510000e835363f.html)

## ABAP DB Table Data Log

DB table data log provides logging for those tables that are critical or susceptible to audits.

Available by using RFC with a custom service.

SAP Documentation: [LINK](https://help.sap.com/viewer/56bf1265a92e4b4d9a72448c579887af/7.5.7/en-US/c769bcd2f36611d3a6510000e835363f.html)

## ABAP SysLog

SAP NetWeaver Application Server (SAP NetWeaver AS) ABAP logs all system errors, warnings, user locks due to failed logon attempts from known users, and process messages in the system log.

Available by SAP Control Web Service.

SAP Documentation: [LINK](https://help.sap.com/viewer/56bf1265a92e4b4d9a72448c579887af/7.5.7/en-US/c769bcbaf36611d3a6510000e835363f.html)

## ABAP ICM Log

The ICM Log records inbound and outbound requests and compiles statistics of the HTTP requests.

Available by SAP Control Web Service.

SAP Documentation: [LINK](https://help.sap.com/viewer/683d6a1797a34730a6e005d1e8de6f22/7.52.4/en-US/a10ec40d01e740b58d0a5231736c434e.html)

## ABAP WorkProcess Log

ABAP WP Log consolidates all work process logs. (default: dev\_\*).

Available by SAP Control Web Service.

SAP Documentation: [LINK](https://help.sap.com/viewer/d0739d980ecf42ae9f3b4c19e21a4b6e/7.3.15/en-US/46fb763b6d4c5515e10000000a1553f6.html)

## ABAP Gateway Log

Gateway logging is used to monitor the activities of the Gateway.

Available by SAP Control Web Service.

SAP Documentation: [LINK](https://help.sap.com/viewer/62b4de4187cb43668d15dac48fc00732/7.5.7/en-US/48b2a710ca1c3079e10000000a42189b.html)

## JAVA Files

This log consolidates all Java files-based logs, including Security Audit Log, System (cluster and server process), Performance, Gateway. Developer Traces, Default Trace.

Available by SAP Control Web Service.

SAP Documentation: General [LINK](https://help.sap.com/viewer/2f8b1599655d4544a3d9c6d1a9b6546b/7.5.9/en-US/485059dfe31672d4e10000000a42189c.html)

SAP Documentation: Java Security Audit Log [LINK](https://help.sap.com/viewer/1531c8a1792f45ab95a4c49ba16dc50b/7.5.9/en-US/4b6013583840584ae10000000a42189c.html)

## HANA DB Audit Trail

Auditing provides you with visibility on who did what in the SAP HANA database (or tried to do what) and when. This allows you, for example, to log and monitor read access to sensitive data.

Available by Sentinel Linux Agent for Syslog.

SAP Documentation: General [LINK](https://help.sap.com/viewer/6b94445c94ae495c83a19646e7c3fd56/2.0.03/en-US/48fd6586304c4f859bf92d64d0cd8b08.html)

SAP Documentation: Audit Trail [LINK](https://help.sap.com/viewer/b3ee5778bc2e4a089d3299b82ec762a7/2.0.03/en-US/0a57444d217649bf94a19c0b68b470cc.html)

# Logs Detailed Structure

## ABAP Security Audit Log

Sentinel Log: ABAPAuditLog\_CL

|  |  |  |
| --- | --- | --- |
| Field | Description | Notes |
| ABAPProgramName | Program Name | SAL Only |
| AlertSeverity | Alert Severity |  |
| AlertSeverityText | Alert Severity Text | SAL Only |
| AlertValue | Alert Value |  |
| AuditClassID | Audit Class ID | SAL Only |
| ClientID | ABAP Client ID (MANDT) |  |
| Computer | User Terminal | SAL Only |
| Email | User Email |  |
| Host | Host |  |
| Instance | ABAP Instance | <HOST>\_<SYSID>\_<SYSNR> |
| MessageClass | Message Class |  |
| MessageContainerID | Message Container ID | XAL Only |
| MessageID | Message ID | E.g. ‘AU1’,’AU2’… |
| MessageText | Message Text |  |
| MonitoringObjectName | MTE Monitor Object Name | XAL Only |
| MonitorShortName | MTE Monitor Short Name | XAL Only |
| SAPProcesType | System Log: SAP Process Type | SAL Only  B\* - Background Processing  D\* - Dialog Processing  U\* - Update Tasks |
| SAPWPName | System Log: Work Process Number | SAL Only |
| SystemID | System ID |  |
| SystemNumber | System Number |  |
| TerminalIPv6 | User Terminal IP | SAL Only |
| TransactionCode | Transaction Code | SAL Only |
| User | User |  |
| Variable1 | Message Variable 1 |  |
| Variable2 | Message Variable 2 |  |
| Variable3 | Message Variable 3 |  |
| Variable4 | Message Variable 4 |  |

## ABAP Change Documents Log

Sentinel Log: ABAPChangeDocsLog\_CL

|  |  |  |
| --- | --- | --- |
| Field | Description | Notes |
| ActualChangeNum | Actial Change Number |  |
| ChangedTableKey | Changed Table Key |  |
| ChangeNumber | Change Number |  |
| ClientID | ABAP Client ID (MANDT) |  |
| CreatedfromPlannedChange | Created From Planned Change | (‘X’ , ‘ ‘) |
| CurrencyKeyNew | Currency Key: New Value |  |
| CurrencyKeyOld | Currency Key: Old Value |  |
| FieldName | Field Name |  |
| FlagText | Flag Text |  |
| Host | Host |  |
| Instance | ABAP Instance | <HOST>\_<SYSID>\_<SYSNR> |
| Language | Language |  |
| ObjectClass | Object Class | e.g. BELEG / BPAR / PFCG / IDENTITY |
| ObjectID | Object ID |  |
| PlannedChangeNum | Planned Change Number |  |
| SystemID | System ID |  |
| SystemNumber | System Number |  |
| TableName | Table Name |  |
| TransactionCode | Transaction Code |  |
| TypeofChange\_Header | Header Type of Change | (U – Change / I – Insert / E – Delete Single Docu / D – Delete / J – Insert Single Docu) |
| TypeofChange\_Item | Item Type of Change | (U – Change / I – Insert / E – Delete Single Docu / D – Delete / J – Insert Single Docu) |
| UOMNew | Unit of Measure: New Value |  |
| UOMOld | Unit of Measure: Old Value |  |
| User | User |  |
| ValueNew | Field Content: New Value |  |
| ValueOld | Field Content: Old Value |  |
| Version | Version |  |

## ABAP Spool Log

Sentinel Log: ABAPSpoolLog\_CL

|  |  |  |
| --- | --- | --- |
| Field | Description | Notes |
| ArchiveStatus | Archive Status |  |
| ArchiveType | Archive Type |  |
| ArchivingDevice | Archiving Device |  |
| AutoRereoute | Auto Rereoute |  |
| ClientID | ABAP Client ID (MANDT) |  |
| CountryKey | Country Key |  |
| DeleteSpoolRequestAuto | Delete Spool Request Auto |  |
| DelFlag | Deletion Flag |  |
| Department | Department |  |
| DocumentType | Document Type |  |
| ExternalMode | External Mode |  |
| FormatType | Format Type |  |
| Host | Host |  |
| Instance | ABAP Instance | <HOST>\_<SYSID>\_<SYSNR> |
| NumofCopies | Number of Copies |  |
| OutputDevice | Output Device |  |
| PrinterLongName | Printer Long Name |  |
| PrintImmediately | Print Immediately |  |
| PrintOSCoverPage | Print OSCover Page |  |
| PrintSAPCoverPage | Print SAPCover Page |  |
| Priority | Priority |  |
| RecipientofSpoolRequest | Recipient of Spool Request |  |
| SpoolErrorStatus | Spool Error Status |  |
| SpoolRequestCompleted | Spool Request Completed |  |
| SpoolRequestisALogForAnotherRequest | Spool Request is A Log For Another Request |  |
| SpoolRequestName | Spool Request Name |  |
| SpoolRequestNumber | Spool Request Number |  |
| SpoolRequestSuffix1 | Spool Request Suffix1 |  |
| SpoolRequestSuffix2 | Spool Request Suffix2 |  |
| SpoolRequestTitle | Spool Request Title |  |
| SystemID | System ID |  |
| SystemNumber | System Number |  |
| TelecommunicationsPartner | Telecommunications Partner |  |
| TelecommunicationsPartnerE | Telecommunications Partner E |  |
| TemSeGeneralcounter | Temse Counter |  |
| TemseNumAddProtectionRule | Temse Num Add Protection Rule |  |
| TemseNumChangeProtectionRule | Temse Num Change Protection Rule |  |
| TemseNumDeleteProtectionRule | Temse Num Delete Protection Rule |  |
| TemSeObjectName | Temse Object Name |  |
| TemSeObjectPart | TemSe Object Part |  |
| TemseReadProtectionRule | Temse Read Protection Rule |  |
| User | User |  |
| ValueAuthCheck | Value Auth Check |  |

## ABAP Spool Output Log

Sentinel Log: ABAPSpoolOutputLog\_CL

|  |  |  |
| --- | --- | --- |
| Field | Description | Notes |
| AppServer | Application Server |  |
| ClientID | ABAP Client ID (MANDT) |  |
| Comment | Comment |  |
| CopyCount | Copy Count |  |
| CopyCounter | Copy Counter |  |
| Department | Department |  |
| ErrorSpoolRequestNumber | Error Request Number |  |
| FormatType | Format Type |  |
| Host | Host |  |
| HostName | Host Name |  |
| HostSpoolerID | Host Spooler ID |  |
| Instance | ABAP Instance | <HOST>\_<SYSID>\_<SYSNR> |
| LastPage | Last Page |  |
| NumofCopies | Numof Copies |  |
| OutputDevice | Output Device |  |
| OutputRequestNumber | Output Request Number |  |
| OutputRequestStatus | Output Request Status |  |
| PhysicalFormatType | Physical Format Type |  |
| PrinterLongName | Printer Long Name |  |
| PrintRequestSize | Print Request Size |  |
| Priority | Priority |  |
| ReasonforOutputRequest | Reason for Output Request |  |
| RecipientofSpoolRequest | Recipientof Spool Request |  |
| SpoolNumberofOutputReqProcessed | Number of Output Requests - Processed |  |
| SpoolNumberofOutputReqWithErrors | Number of Output Requests - With Errors |  |
| SpoolNumberofOutputReqWithProblems | Number of Output Requests - With Problems |  |
| SpoolRequestNumber | Spool Request Number |  |
| StartPage | Start Page |  |
| SystemID | System ID |  |
| SystemNumber | System Number |  |
| TelecommunicationsPartner | Telecommunications Partner |  |
| TemSeGeneralcounter | Temse Counter |  |
| Title | Title |  |
| User | User |  |

## ABAP Job Log

Sentinel Log: ABAPJobLog\_CL

|  |  |  |
| --- | --- | --- |
| Field | Description | Notes |
| ABAPProgram | ABAP Program |  |
| BgdEventParameters | Background Event Parameters |  |
| BgdProcessingEvent | Background Processing Event |  |
| ClientID | ABAP Client ID (MANDT) |  |
| DynproNumber | Dynpro Number |  |
| GUIStatus | GUI Status |  |
| Host | Host |  |
| Instance | ABAP Instance (HOST\_SYSID\_SYSNR) | <HOST>\_<SYSID>\_<SYSNR> |
| JobClassification | Job Classification |  |
| JobCount | Job Count |  |
| JobGroup | Job Group |  |
| JobName | Job Name |  |
| JobPriority | Job Priority |  |
| MessageClass | Message Class |  |
| MessageNumber | Message Number |  |
| MessageText | Message Text |  |
| MessageType | Message Type |  |
| ReleaseUser | Job Release User |  |
| SchedulingDateTime | Scheduling Date Time |  |
| StartDateTime | Start Date Time |  |
| SystemID | System ID |  |
| SystemNumber | System Number |  |
| TargetServer | Target Server |  |
| User | User |  |
| UserReleaseInstance | ABAP Instance - User Release |  |
| WorkProcessID | Work Process ID |  |
| WorkProcessNumber | Work Process Number |  |

## ABAP Application Log

Sentinel Log: ABAPAppLog\_CL

|  |  |  |
| --- | --- | --- |
| Field | Description | Notes |
| AppLogDateTime | Application Log Date Time |  |
| CallbackProgram | Callback Program |  |
| CallbackRoutine | Callback Routine |  |
| CallbackType | Callback Type |  |
| ClientID | ABAP Client ID (MANDT) |  |
| ContextDDIC | Context DDIC Structure |  |
| ExternalID | External Log ID |  |
| Host | Host |  |
| Instance | ABAP Instance | <HOST>\_<SYSID>\_<SYSNR> |
| InternalMessageSerial | Application Log Message Serial |  |
| LevelofDetail | Level of Detail |  |
| LogHandle | Application Log Handle |  |
| LogNumber | Log Number |  |
| MessageClass | Message Class |  |
| MessageNumber | Message Number |  |
| MessageText | Message Text |  |
| MessageType | Message Type |  |
| Object | Application Log Object |  |
| OperationMode | Operation Mode |  |
| ProblemClass | Problem Class |  |
| ProgramName | Program Name |  |
| SortCriterion | Sort Criterion |  |
| StandardText | Standard Text |  |
| SubObject | Application Log Sub Object |  |
| SystemID | System ID |  |
| SystemNumber | System Number |  |
| TransactionCode | Transaction Code |  |
| User | User |  |
| UserChange | User Change |  |

## ABAP Workflow Log

Sentinel Log: ABAPWorkflowLog\_CL

|  |  |  |
| --- | --- | --- |
| Field | Description | Notes |
| ActualAgent | Actual Agent |  |
| Address | Address |  |
| ApplicationArea | Application Area |  |
| CallbackFunction | Callback Function |  |
| ClientID | ABAP Client ID (MANDT) |  |
| CreationDateTime | Creation Date Time |  |
| Creator | Creator |  |
| CreatorAddress | Creator Address |  |
| ErrorType | Error Type |  |
| ExceptionforMethod | Exception for Method |  |
| Host | Host |  |
| Instance | ABAP Instance (HOST\_SYSID\_SYSNR) | <HOST>\_<SYSID>\_<SYSNR> |
| Language | Language |  |
| LogCounter | Log Counter |  |
| MessageNumber | Message Number |  |
| MessageType | Message Type |  |
| MethodUser | Method User |  |
| Priority | Priority |  |
| SimpleContainer | Simple Container | Packed as list of Key-Value Entities of the workitem |
| Status | Status |  |
| SuperWI | Super WI |  |
| SystemID | System ID |  |
| SystemNumber | System Number |  |
| TaskID | Task ID |  |
| TasksClassification | Tasks Classification |  |
| TaskText | Task Text |  |
| TopTaskID | Top Task ID |  |
| UserCreated | User Created |  |
| WIText | Work Item Text |  |
| WIType | Work Item Type |  |
| WorkflowAction | Workflow Action |  |
| WorkItemID | Work Item ID |  |

## ABAP CR Log

Sentinel Log: ABAPCRLog\_CL

|  |  |  |
| --- | --- | --- |
| Field | Description | Notes |
| Category | Category (Workbench, Customizing) |  |
| ClientID | ABAP Client ID (MANDT) |  |
| Description | Description |  |
| Host | Host |  |
| Instance | ABAP Instance | <HOST>\_<SYSID>\_<SYSNR> |
| ObjectName | Object Name |  |
| ObjectType | Object Type |  |
| Owner | Owner |  |
| Request | Change Request |  |
| Status | Status |  |
| SystemID | System ID |  |
| SystemNumber | System Number |  |
| TableKey | Table Key |  |
| TableName | Table Name |  |
| ViewName | View Name |  |

## ABAP DB Table Data Log

Sentinel Log: ABAPTableDataLog\_CL

|  |  |  |
| --- | --- | --- |
| Field | Description | Notes |
| DBLogID | DB Log ID |  |
| Host | Host |  |
| Instance | ABAP Instance | <HOST>\_<SYSID>\_<SYSNR> |
| Language | Language |  |
| LogKey | Log Key |  |
| NewValue | Field New Value |  |
| OldValue | Field Old Value |  |
| OperationTypeSQL | Operation Type (Insert/Update/Delete) |  |
| Program | Program Name |  |
| SystemID | System ID |  |
| SystemNumber | System Number |  |
| TableField | Table Field |  |
| TableName | Table Name |  |
| TransactionCode | Transaction Code |  |
| UserName | User |  |
| VersionNumber | Version Number |  |

## ABAP SysLog

Sentinel Log: SysLog\_CL

|  |  |  |
| --- | --- | --- |
| Field | Description | Notes |
| ClientID | ABAP Client ID (MANDT) |  |
| Host | Host |  |
| Instance | ABAP Instance | <HOST>\_<SYSID>\_<SYSNR> |
| MessageNumber | Message Number |  |
| MessageText | Message Text |  |
| Severity | Message Severity | Debug/Info/Warning/Error |
| SystemID | System ID |  |
| SystemNumber | System Number |  |
| TransacationCode | Transaction Code |  |
| Type | SAP Process Type |  |
| User | User |  |

## ABAP ICM Log

Sentinel Log: ICM\_CL

|  |  |  |
| --- | --- | --- |
| Field | Description | Notes |
| Host | Host |  |
| Instance | ABAP Instance | <HOST>\_<SYSID>\_<SYSNR> |
| MessageText | Message Text |  |
| Severity | Message Severity | Debug/Info/Warning/Error |
| SystemID | System ID |  |
| SystemNumber | System Number |  |

## ABAP WorkProcess Log

Sentinel Log: WP\_CL

|  |  |  |
| --- | --- | --- |
| Field | Description | Notes |
| Host | Host |  |
| Instance | ABAP Instance | <HOST>\_<SYSID>\_<SYSNR> |
| MessageText | Message Text |  |
| Severity | Message Severity | Debug/Info/Warning/Error |
| SystemID | System ID |  |
| SystemNumber | System Number |  |
| WPNumber | Work Porcess Number |  |

## ABAP Gateway Log

Sentinel Log: GW\_CL

|  |  |  |
| --- | --- | --- |
| Field | Description | Notes |
| Host | Host |  |
| Instance | ABAP Instance | <HOST>\_<SYSID>\_<SYSNR> |
| MessageText | Message Text |  |
| Severity | Message Severity | Debug/Info/Warning/Error |
| SystemID | System ID |  |
| SystemNumber | System Number |  |

## JAVA Files

Sentinel Log: JavaFilesLogsCL

|  |  |  |
| --- | --- | --- |
| Field | Description | Notes |
| Application | Java Application |  |
| ClientID | Client ID |  |
| CSNComponent | CSN Component | e.g BC-XI-IBD |
| DCComponent | DC Component | e.g. com.sap.xi.util.misc |
| DSRCounter | DSR Counter |  |
| DSRRootContentID | DSR Context GUID |  |
| DSRTransaction | DSR Transaction GUID |  |
| Host | Host |  |
| Instance | Java Instance | <HOST>\_<SYSID>\_<SYSNR> |
| Location | Java Class |  |
| LogName | Java LogName | e.g. Available / defaulttrace / dev\* / security / etc.. |
| MessageText | Message Text |  |
| MNo | Message Number |  |
| Pid | Process ID |  |
| Program | Program Name |  |
| Session | Session |  |
| Severity | Message Severity | Debug/Info/Warning/Error |
| Solution | Solution |  |
| SystemID | System ID |  |
| SystemNumber | System Number |  |
| ThreadName | Thread Name |  |
| Thrown | Exception Thrown |  |
| TimeZone | Timezone |  |
| User | User |  |

## HANA DB Audit Trail

Sentinel Log: Syslog

|  |  |  |
| --- | --- | --- |
| Field | Description | Notes |
| Computer | Host Name |  |
| HostIP | Host IP |  |
| HostName | Host Name |  |
| ProcessID | Process ID |  |
| ProcessName | Process Name | HDB\* |
| SeverityLevel | Alert |  |
| SourceSystem |  | Linux |
| SyslogMessage | Message | Unparsed Audit Trail Message |

# Watchlists

Following watchlists are used in various Analytic Rules.

It is recommended to maintain them instead of using fixed values in queries.

Sample CSV files are Included. Watchlists should be created manually in Sentinel.

## SAP – Systems

Describes the landscape of SAP systems according to role and usage.

|  |  |  |
| --- | --- | --- |
| Field | Description | Notes |
| SystemID | System ID | SYSID |
| SystemRole | System Role | Sandbox/Developement/Quality Assurance/ Training/Production |
| SystemUsage | System Usage | ERP/BW/Solman/Gateway/Enterprise Portal |

## SAP - Privileged Users

Privileged users which are under extra restrictions.

|  |  |  |
| --- | --- | --- |
| Field | Description | Notes |
| User | ABAP User | e.g. DDIC / SAP\* |
| Description | Description | To be used for data management |

## SAP - Sensitive Tables

Sensitive tables which their access should be governed.

|  |  |  |
| --- | --- | --- |
| Field | Description | Notes |
| Table | ABAP Dictionary Table | e.g. USR02 / PA0008 |
| Description | Description | To be used for data management |

## SAP - Sensitive Transactions

Sensitive transactions which their execution should be governed.

|  |  |  |
| --- | --- | --- |
| Field | Description | Notes |
| TransactionCode | Transaction Code | e.g. RZ11 |
| Description | Description | To be used for data management |

## SAP - Sensitive Roles

Sensitive roles which their assignment should be governed.

|  |  |  |
| --- | --- | --- |
| Field | Description | Notes |
| Role | Authorization Role | e.g. SAP\_BC\_BASIS\_ADMIN |
| Description | Description | To be used for data management |

## SAP - Sensitive Profiles

Sensitive profiles which their assignment should be governed.

|  |  |  |
| --- | --- | --- |
| Field | Description | Notes |
| Profile | Authorizations Profile | e.g. SAP\_ALL / SAP\_NEW |
| Description | Description | To be used for data management |

## SAP - Sensitive ABAP Programs

Sensitive ABAP programs (reports) which their execution should be governed.

|  |  |  |
| --- | --- | --- |
| Field | Description | Notes |
| ABAPProgram | ABAP Program / Report | e.g. RSPFLDOC |
| Description | Description | To be used for data management |

## SAP - Critical Authorization Objects

Critical Authorizations object which their assignment should be governed.

Please see example in template CSV file for the different options of usage.

|  |  |  |
| --- | --- | --- |
| Field | Description | Notes |
| AuthorizationObject | SAP Authorization Object | e.g. S\_DEVELOP / S\_TCODE.  Table TOBJ |
| AuthorizationField | Authorization Field | e.g. OBJTYP / TCD |
| AuthorizationValue | Authorization Value | e.g. ‘DEBUG’ |
| ActivityField | Activity field | For Authorizations objects without Activity or only with Activity field to e filled with “NOT\_IN\_USE”. For most cases this will be “ACTVT”. |
| Activity | Activity | According to object, Usually 01 – Create, 02 – Change, 03 – Display etc.. |
| Description | Description |  |

## SAP – Networks

Internal and maintenance networks for identification of unauthorized logins.

|  |  |  |
| --- | --- | --- |
| Field | Description | Notes |
| Network | Network | e.g. 111.68.128.0/17 |
| Description | Description | To be used for data management |

## SAP – Sensitive Function Module

Sensitive Function Modules which their execution should be governed.

|  |  |  |
| --- | --- | --- |
| Field | Description | Notes |
| FunctionModule | ABAP Function Module | e.g. RSAU\_CLEAR\_AUDIT\_LOG |
| Description | Description | To be used for data management |

## SAP – Excluded Networks

Internal and maintenance of excluded networks for ignoring web dispatchers, terminal servers etc.

|  |  |  |
| --- | --- | --- |
| Field | Description | Notes |
| Network | Network to exclude | e.g. 111.68.128.0/17 |
| Description | Description | To be used for data management |

## SAP – Excluded Users

System users which are logged in the systems need to be ignored. (for example Multiple logons by user alert)

|  |  |  |
| --- | --- | --- |
| Field | Description | Notes |
| User | ABAP User | e.g. SYSWF |
| Description | Description | To be used for data management |

# Analytics

The following analytic rules are supplied using .json files and should be imported to Sentinel Workspace after connector is active and log tables were created.

**Installation:**

Use the PowerShell script available [here](https://github.com/Azure/Azure-Sentinel/tree/master/Tools/Az.SecurityInsights-Samples/Alert%20Rules/Import%20Analytics%20Rules) to import it.

Please also manually import the corresponding watchlists

**Limitations**: User IP (terminal IP) is only available via Audit SAL method - SAP 7.5 and above

## SAP - High - Data has Changed during Debugging Activity

### Description

Identifies changes for runtime data during a debugging activity. Source Action: Activate Debug ("/h"), Select a field for change and update it's value.

**Recommended for Production only**

*Data Sources: SAPcon - Audit Log*

### Tactics

Execution

Lateral Movement

### KQL

let Role = 'Production';

let AuditClasses = dynamic(['CUL']); // Audit Log Classes - Debug Change

let allSystemRoles = dynamic(['Sandbox','Developement','QualityAssurance','Training','Production']); // Available System Roles

let allSystemUsage = dynamic (['ERP','CRM','BW','Solman','Gateway','Enterprise Portal']); // Available System Usages

// Get Relevant Systems from WatchList

let systemID = \_GetWatchlist('SAP - Systems');

let fixedSID = datatable(SystemID:string, SystemRole:string, SystemUsage:string)

// Maintain these if WatchList is not available

["S4H","Production","ERP",

"XXX","Sandbox","BW"]

;

let UnitedSystem =

union systemID, fixedSID

| summarize by SystemID, SystemRole, SystemUsage

| where SystemRole == Role; // Recommended is Production only

//| where SystemRole in (allSystemRoles); // Use this for all system roles

ABAPAuditLog\_CL

| where MessageID\_s in (AuditClasses)

| project-rename SystemID = SystemID\_s

| lookup kind = inner (UnitedSystem) on SystemID

| project TimeGenerated, User\_s, MessageText\_s, ABAPProgramName\_s, TransactionCode\_s, SystemID, SystemRole, SystemUsage,

AccountCustomEntity = Email\_s, IPCustomEntity = TerminalIPv6\_s, HostCustomEntity = Host\_s

## SAP - High - System Configuration Change

### Description

Identifies changes for system configuration.

Source Action: Adapt system change options or software components modifcation using SE06 transaction code.

*Data Sources: SAPcon - Audit Log*

### Tactics

Exfiltration

Defense Evasion

Persistence

### KQL

// Audit Log Classes - System Change Configuration

let AuditClasses = dynamic(['EU1']);

ABAPAuditLog\_CL

| where MessageID\_s in (AuditClasses) // Relevant message

| project

// Details

TimeGenerated, SystemID\_s, User\_s, TransactionCode\_s, SoftwareComponent = Variable1\_s, NewModifiabilityStatus = Variable2\_s, MessageText\_s,

AccountCustomEntity = Email\_s, IPCustomEntity = TerminalIPv6\_s, HostCustomEntity = Host\_s

## SAP - High - Client Configuration Change

### Description

Identifies changes for client configuration such as: Client role, Changes recording mode.

Source Action: Perofrm client configurations changes using SCC4 transaction code.

Data Sources: SAPcon - Audit Log

### Tactics

Defense Evasion

Exfiltration

Persistence

### KQL

// Audit Log Classes - Client Change Configuration

let AuditClasses = dynamic(['EU2']); // Relevant message

ABAPAuditLog\_CL

| where MessageID\_s in (AuditClasses)

| extend ClientID = Variable1\_s

| parse Variable2\_s with Currency "|" ClientRole "|" RecordingChanges "|" CrossClientObjectChanges "|" ClientCopyProtectionLevel "|" ProtectionSAPUpgrade "|" CATTeCATT "|"  LockedforCopy // Parse every object before the | char

| project TimeGenerated, SystemID\_s, User\_s, ClientID,

Currency,ClientRole,RecordingChanges,CrossClientObjectChanges,ClientCopyProtectionLevel,CATTeCATT,LockedforCopy,ProtectionSAPUpgrade,

MessageText\_s,

AccountCustomEntity = Email\_s, IPCustomEntity = TerminalIPv6\_s, HostCustomEntity = Host\_s

## SAP - High - Deactivation of Security Audit Log

### Description

Identifies deactivation of Security Audit Log

Source Action: Disable secruity Audit Log using SM19/RSAU\_CONFIG.

Data Sources: SAPcon - Audit Log

### Tactics Exfiltration

Defense Evasion

Persistence

### KQL

// Audit Log Classes - Audit Log Active Status Events

let AuditClasses = dynamic(['AUJ']);

ABAPAuditLog\_CL

| where MessageID\_s in (AuditClasses)

| where Variable1\_s == '0' // Audit Active Status = 0

| project

// Details

TimeGenerated, SystemID\_s, User\_s, MessageText\_s,

AccountCustomEntity = Email\_s, IPCustomEntity = TerminalIPv6\_s, HostCustomEntity = Host\_s

## SAP - High - Sensitive privileged user logged in

### Description

Identifies Dialog logon of a sensitive privileged user.

Source Action: Logon to the backend system using SAP\* or anoter privileged user.

Priveleged users should be maintained in "SAP - Privileged Users" Watchlist

Data Sources: SAPcon - Audit Log

### Tactics

Initial Access

Credential Access

### KQL

// Define Variables

// Audit Log Classes - Dialog Logon Successful

let AuditClassesSuccess = dynamic(['AU1']);

let AuditClassesFail = dynamic(['BU1']);

let AuditRFCSuccess = dynamic(['AU5']);

let AuditRFCFail = dynamic(['AU6']);

let LogonTypes = dynamic(['A','H', 'R', 'S']); // Dialog / HTTP

// Get Relevant User from WatchList

let PrivelegedUsers = \_GetWatchlist('SAP - Privileged Users');

let fixedUsers = datatable(User:string)

// Maintain these if WatchList is not available

["SAP\*","DDIC"]

;

let UnitedPrivileged = union PrivelegedUsers, fixedUsers

| summarize by User;

// Query logic

ABAPAuditLog\_CL

| where (MessageID\_s in (AuditClassesSuccess) and Variable1\_s in (LogonTypes)) or // Success login

MessageID\_s in (AuditClassesFail) or // Failed login

(MessageID\_s in (AuditRFCSuccess) and Variable1\_s in (LogonTypes)) or // Success RFC login

MessageID\_s in (AuditRFCFail) // Failed RFC login

| where User\_s in (UnitedPrivileged)

| project-rename LogonType = Variable1\_s

| project TimeGenerated, SystemID\_s, ClientID\_s, LogonType, User\_s, MessageText\_s,

AccountCustomEntity = Email\_s, IPCustomEntity = TerminalIPv6\_s, HostCustomEntity = Host\_s

## SAP - High - Execution of a Sensitive ABAP Program

### Description

Identifies direct execution of a sensitive ABAP program.

Source Action: Execute a program directly using SE38/SA38/SE80.

**Recommended for Production only**

ABAP Programs should be maintained in watchlist "SAP - Sensitive ABAP Programs"

Data Sources: SAPcon - Audit Log

### Tactics

Exfiltration

Lateral Movement

Execution

### KQL

// Define Variables

let Role = "Production";

let AuditClasses = dynamic(['AUW']); // Audit Log Classes - Report Started

let allSystemRoles = dynamic(['Sandbox','Developement','QualityAssurance','Training','Production']); // Available System Roles

let allSystemUsage = dynamic (['ERP','CRM','BW','Solman','Gateway','Enterprise Portal']); // Available System Usages

// Get Relevant Systems from WatchList

let systemID = \_GetWatchlist('SAP - Systems');

let fixedSID = datatable(SystemID:string, SystemRole:string, SystemUsage:string)

// Maintain these if WatchList is not available

["S4H","Production","ERP",

"XXX","Sandbox","BW"]

;

// Get Relevant ABAP Programs

let SensitiveABAPReports = \_GetWatchlist('SAP - Sensitive ABAP Programs');

let fixedABAPReports = datatable(ABAPProgram:string)

// Maintain these if WatchList is not available

["RSPFLDOC"]

;

let UnionAbap =

union SensitiveABAPReports, fixedABAPReports

| summarize by ABAPProgram;

let UnitedSystem =

union systemID, fixedSID

| summarize by SystemID, SystemRole, SystemUsage

| where SystemRole == Role; // Reccommended is Production only

//| where SystemRole in (allSystemRoles); // Use this for all system roles

// Query logic

ABAPAuditLog\_CL

| where MessageID\_s in (AuditClasses)

| where ABAPProgramName\_s in (UnionAbap)

| project-rename SystemID = SystemID\_s

| lookup kind = inner (UnitedSystem) on SystemID

| order by TimeGenerated asc

| project TimeGenerated, SystemID, ClientID\_s, User\_s, ABAPProgramName\_s, MessageText\_s, TransactionCode\_s, MessageID\_s, AccountCustomEntity = Email\_s, IPCustomEntity = TerminalIPv6\_s, HostCustomEntity = Host\_s

## SAP - High - Execution of a Sensitive Transaction Code

### Description

Identifies execution of a sensitive Transaction Code.

Source Action: Execute a sensitive Transaction Code.

**Recommended for Production only**

Transaction Codes should be maintained in watchlist ""SAP - Sensitive Transaction Codes""

Data Sources: SAPcon - Audit Log

### Tactics Discovery

Execution

### KQL

// Define Variables

let Role = 'Production';

let AuditClasses = dynamic(['AU3']); // Audit Log Classes - Transaction Started

let allSystemRoles = dynamic(['Sandbox','Developement','QualityAssurance','Training','Production']); // Available System Roles

let allSystemUsage = dynamic (['ERP','CRM','BW','Solman','Gateway','Enterprise Portal']); // Available System Usages

// Get Relevant Systems from WatchList

let systemID = \_GetWatchlist('SAP - Systems');

let fixedSID = datatable(SystemID:string, SystemRole:string, SystemUsage:string)

// Maintain these if WatchList is not available

["S4H","Production","ERP",

"XXX","Sandbox","BW"]

;

// Get Relevant Transaction Codes

let SensitiveTcode = \_GetWatchlist('SAP - Sensitive Transactions');

let fixedTcode = datatable(TransactionCode:string)

// Maintain these if WatchList is not available

["RSAU\_CONFIG","RZ11","SM19"]

;

let UnitedCodes =

union SensitiveTcode, fixedTcode

| summarize by TransactionCode;

let UnitedSystem =

union systemID, fixedSID

| summarize by SystemID, SystemRole, SystemUsage

| where SystemRole == Role; // Recommended is Production only

//| where SystemRole in (allSystemRoles); // Use this for all system roles

// Query logic

ABAPAuditLog\_CL

| where MessageID\_s in (AuditClasses)

| where TransactionCode\_s in (UnitedCodes)

| project-rename SystemID = SystemID\_s

| lookup kind = inner (UnitedSystem) on SystemID

| order by TimeGenerated asc

| project TimeGenerated, SystemID, ClientID\_s, User\_s, TransactionCode\_s, MessageText\_s, MessageID\_s, AccountCustomEntity = Email\_s, IPCustomEntity = TerminalIPv6\_s, HostCustomEntity = Host\_s

## SAP - High - Login from unexpected network

### Description

Identifies logons from an unexpected network.

Source Action: Logon to the backend system from an IP address which is not assigned to one of the netwroks.

Networks should be maintained in watchlist "SAP - Networks"

Data Sources: SAPcon - Audit Log

### Tactics

Initial Access

### KQL

let AuditClasses = dynamic(['AU1','AU5']); // Audit Log Classes - Dialog Logon Successful, RFC Logon Successful

// Dialog / CPIC / RFC Int / RFC Ext / SRFC / User Switch / HTTP / Restore Session / API Call

let DialogLogonTypes = dynamic(['A', 'C', 'F', 'R', 'S', 'U', 'H', 'u', ' ']);

let Networks = \_GetWatchlist('SAP - Networks');

let fixedNetworks = datatable(Network: string)['111.68.128.0/17']; // Maintain these if watchlist is not available

let allNetworks = union Networks, fixedNetworks

    | summarize by Network;

ABAPAuditLog\_CL

// Add audit classes

| where MessageID\_s in (AuditClasses)

| where Variable1\_s in (DialogLogonTypes) // Is a dialog logon type from the list

| where isnotempty(TerminalIPv6\_s) // There is a Ipv6 address

| evaluate ipv4\_lookup(allNetworks, TerminalIPv6\_s, Network, return\_unmatched = true)

// Similar to regular lookup, by ipv4 address, unmatched is like left join

| where isempty(Network) // Network is not familiar

// Details

| project TimeGenerated, SystemID\_s, ClientID\_s, User\_s, TransactionCode\_s, MessageText\_s,

    AccountCustomEntity = Email\_s, IPCustomEntity = TerminalIPv6\_s, HostCustomEntity = Host\_s

## SAP - High - RFC Execution of a Sensitive Function Module

### Description

Identifies execution o f a sensitive function module using RFC.

Source Action: Execute a function module using RFC.

**Recommended for Production only**

Function Modules should be maintained in watchlist "SAP - Sensitive Function Modules"

Data Sources: SAPcon - Audit Log

### Tactics

Execution

Lateral Movement

Discovery

### KQL

let Role = "Production";

// Audit Log Classes - Successful RFC call &C (function group = &A)

let AuditClasses = dynamic(['AUK']);

let allSystemRoles = dynamic(['Sandbox', 'Developement', 'QualityAssurance', 'Training', 'Production']); // Available System Roles

let allSystemUsage = dynamic (['ERP', 'CRM', 'BW', 'Solman', 'Gateway', 'Enterprise Portal']); // Available System Usages

// Get Relevant Systems from WatchList

let systemID = \_GetWatchlist('SAP - Systems');

let fixedSID = datatable(SystemID: string, SystemRole: string, SystemUsage: string)

    // Maintain these if WatchList is not available

    ["S4H", "Production", "ERP",

    "XXX", "Sandbox", "BW"]

;

// Get Relevant Function Modules

let SensitiveFM = \_GetWatchlist('SAP - Sensitive Function Modules');

let fixedFM = datatable(FunctionModule: string)

    // Maintain these if WatchList is not available

    ["RSAU\_CLEAR\_AUDIT\_LOG"]

;

let UnitedSystems = union systemID, fixedSID

| where SystemRole == Role // Recommended is Production only

| summarize by SystemID;

//| where SystemRole in (allSystemRoles); // Use this for all system roles

let UnitedSensitive = union SensitiveFM, fixedFM

| summarize by FunctionModule;

ABAPAuditLog\_CL

| where MessageID\_s in (AuditClasses)

| project-rename FunctionModule = Variable3\_s, FunctionGroup = Variable1\_s

| where SystemID\_s in (UnitedSystems) // The systemID is in this list

| where FunctionModule in (UnitedSensitive) // Function module is sensitive

| order by TimeGenerated asc

| project TimeGenerated, User\_s, SystemID\_s, ClientID\_s, MessageText\_s, FunctionGroup, FunctionModule, MessageID\_s,

AccountCustomEntity = Email\_s, IPCustomEntity = TerminalIPv6\_s, HostCustomEntity = Host\_s

## SAP - High - Function Module tested

### Description

Identifies testing of a function module.

Source Action: Test a function module using SE37 / SE80.

**Recommended for Production only**

Data Sources: SAPcon - Audit Log

### Tactics

Collection

Defense Evasion

Lateral Movement

### KQL

let Role = 'Production';

let ProgramName = 'RS\_TESTFRAME\_CALL';

let AuditClasses = dynamic(['AUW']); // Audit Log Classes - Report Started

let allSystemRoles = dynamic(['Sandbox', 'Developement', 'QualityAssurance', 'Training', 'Production']); // Available System Roles

let allSystemUsage = dynamic (['ERP', 'CRM', 'BW', 'Solman', 'Gateway', 'Enterprise Portal']); // Available System Usages

// Get Relevant Systems from WatchList

let systemID = \_GetWatchlist('SAP - Systems');

let fixedSID = datatable(SystemID: string, SystemRole: string, SystemUsage: string)

// Maintain these if WatchList is not available

["S4H", "Production", "ERP",

"XXX", "Sandbox", "BW"]

;

let UnitedSystem =

union systemID, fixedSID

| summarize by SystemID, SystemRole, SystemUsage

| where SystemRole == Role; // Reccommended is Production only

//| where SystemRole in (allSystemRoles); // Use this for all system roles

ABAPAuditLog\_CL

| where MessageID\_s in (AuditClasses)

| where ABAPProgramName\_s == ProgramName

| project-rename SystemID = SystemID\_s

| lookup kind = inner (UnitedSystem) on SystemID

| order by TimeGenerated asc

| project TimeGenerated, User\_s, SystemID, ClientID\_s, MessageText\_s, MessageID\_s,

AccountCustomEntity = Email\_s, IPCustomEntity = TerminalIPv6\_s, HostCustomEntity = Host\_s

## SAP - High - Change in Sensitive privileged user

### Description

Identifies changes of sensitive privileged users.

Source Action: Change user details / authorizations using SU01.

Priveleged users should be maintained in "SAP - Privileged Users" Watchlist

Data Sources: SAPcon - Audit Log

### Tactics

Privilege Escalation

Credential Access

### KQL

// Audit Log Classes - User Master Changes

let AuditClasses = dynamic(['AU7', 'BUV', 'BUW', 'AU8', 'AU9', 'AUA', 'AUB', 'AUD', 'DUH', 'BU2']);

// Get Relevant User from WatchList

let PrivelegedUsers = \_GetWatchlist('SAP - Privileged Users');

let fixedUsers = datatable(User: string)

// Maintain these if WatchList is not available

["SAP\*", "DDIC"]

;

let UnitedPrivleged = union PrivelegedUsers, fixedUsers

| summarize by User;

ABAPAuditLog\_CL

| where MessageID\_s in (AuditClasses)

| where Variable1\_s in (UnitedPrivleged) // The user that we are making change in is a sensitive privileged user

| project TimeGenerated, SystemID\_s, ClientID\_s, User\_s, MessageText\_s, MessageID\_s,

AccountCustomEntity = Email\_s, IPCustomEntity = TerminalIPv6\_s, HostCustomEntity = Host\_s

## SAP - Medium - Security Audit Log Configuration Change

### Description

Idenitifes changes for configuration in Securiy Audit Log

Source Action: change any Security Audit Log Configuration using SM19/RSAU\_CONFIG. (Filters/Status/Recording mode etc..)

Data Sources: SAPcon - Audit Log

### Tactics

Persistence

Exfiltration

Defense Evasion

### KQL

// Audit Log Classes - Audit Log Configuration Events

let AuditClasses = dynamic(['AUE','AUF','AUI','AUJ','FU0','E05']); // Relevant messages

ABAPAuditLog\_CL

| where MessageID\_s in (AuditClasses)

| project

// Details

TimeGenerated, SystemID\_s, User\_s, MessageText\_s,

AccountCustomEntity = Email\_s, IPCustomEntity = TerminalIPv6\_s, HostCustomEntity = Host\_s

## SAP - Medium - Transaction is unlocked

### Description

Identifies unlocking of a transaction.

Source Action: Unlock a transaction code using SM01/SM01\_DEV/SM01\_CUS.

Data Sources: SAPcon - Audit Log

### Tactics

Persistence

Execution

### KQL

// Audit Log Classes - Transaction UnLock Events

// AUQ - Transaction Unlocked

let AuditClasses = dynamic(['AUQ']);

ABAPAuditLog\_CL

| where MessageID\_s in (AuditClasses)

| project-rename TransactionCode = Variable1\_s

| parse TransactionCode with "( TR ) " \_TCODE " - " ClientTR // Parse to \_TCODE and ClientTR

// Specific Client Action (SM01\_CUS) / Cross Client (SM01\_DEV)

| extend TransactionCode = iif(\_TCODE != "",\_TCODE, TransactionCode) // Check if \_TCODE is not empty

| extend ClientTR = iif(ClientTR != "",ClientTR, "CrossClient") // Check if ClientTR is not empty

| project

// Details

TimeGenerated, SystemID\_s, User\_s, MessageText\_s,TransactionCode, ClientTR,

AccountCustomEntity = Email\_s, IPCustomEntity = TerminalIPv6\_s, HostCustomEntity = Host\_s

## SAP - Medium - Assignment of a sensitive profile

### Description

Identifies new assignments of a sensitive profile to a user.

Source Action: Assign a profile to a User using SU01.

Sensitive profiles should be maintained in watchlist "SAP - Sensitive Profiles"

Data Sources: SAPcon - Change Documents Log

### Tactics

Privilege Escalation

### KQL

// Define Variables

// Audit Log Classes - Authorizations for user changed

let Identity = 'IDENTITY';

let ProfileChangeDoc = 'SUSR\_UST04';

let Insert = "I";

let logsThreshold = 3; // 3 seconds

let AuditClasses = dynamic(['AUB']); // Authorizations for user &A changed.

// Maintain these if WatchList is not available

let SensitiveProfiles = \_GetWatchlist('SAP - Sensitive Profiles');

let fixedProfile = datatable(Profile:string)['SAP\_ALL','SAP\_NEW'];

// Maintain these if System doesn't have CR's

let fixedChangeDocs = datatable(User\_s : string, ObjectClass\_s : string, TableName\_s : string, TypeofChange\_Item\_s : string , ChangedTableKey\_s : string, ObjectID\_s : string, TimeGenerated : datetime, ValueNew\_s : string, SystemID\_s : string)[];

let ChangeDocs =

union isfuzzy=true table("ABAPChangeDocsLog\_CL"), fixedChangeDocs;

let IdentityChangeDocuments =

// Identity Change documents which represents profiles assignment

ChangeDocs

| where ObjectClass\_s == Identity // Identity

and TableName\_s == ProfileChangeDoc // Profile Change Doc

and TypeofChange\_Item\_s == Insert // Insert

| extend Profile = ChangedTableKey\_s

| extend UserAssigned = ObjectID\_s;

let UnitedProfiles =

toscalar(union fixedProfile, SensitiveProfiles

| summarize Profiles = make\_list(Profile));

// Query logic

ABAPAuditLog\_CL

| where MessageID\_s in (AuditClasses)

| summarize by TimeGenerated, TerminalIPv6\_s, User\_s, Host\_s, Email\_s

| lookup kind = leftouter (IdentityChangeDocuments) on User\_s

| where Profile in (UnitedProfiles)

| project-rename TimeGenAudit = TimeGenerated1

| where abs(datetime\_diff('second',TimeGenerated,TimeGenAudit)) <= logsThreshold

or isnull(TimeGenAudit)

| project

// Details

TimeGenerated, SystemID\_s, ClientID\_s, Profile, User\_s, UserAssigned,

AccountCustomEntity = Email\_s, IPCustomEntity = TerminalIPv6\_s, HostCustomEntity = Host\_s

## SAP - Medium - Assignment of a sensitive role

### Description

Identifies new assignments for a sensitive role to a user.

Source Action: Assign a role to a User using SU01 / PFCG.

Sensitive roles should be maintained in watchlist "SAP - Sensitive Roles"

Data Sources: SAPcon - Change Documents Log, Audit Log

### Tactics

Privilege Escalation

### KQL

// Define Variables

let Roles = 'PFCG';

let UsersRoles = 'AGR\_USERS';

let Insert = "I";

let logsThreshold = 3; // 3 seconds

// Audit Log Classes - Authorizations for user changed

let AuditClasses = dynamic(['AUB']); // Authorizations for user &A changed.

// Maintain these if WatchList is not available

let SensitiveRoles = \_GetWatchlist('SAP - Sensitive Roles');

let fixedRoles = datatable(Role: string)['SAP\_BC\_BASIS\_ADMIN', 'SAP\_BC\_AUTH\_PROFILE\_ADMIN'];

// Maintain these if System doesn't have CR's

let fixedChangeDocs = datatable(User\_s : string, ObjectClass\_s : string, TableName\_s : string, TypeofChange\_Item\_s : string ,

ChangedTableKey\_s : string, ObjectID\_s : string, TimeGenerated : datetime, ValueNew\_s : string, SystemID\_s : string)[];

let ChangeDocs =

union isfuzzy=true table("ABAPChangeDocsLog\_CL"), fixedChangeDocs;

let ChangeCheck =

ChangeDocs

| where ObjectClass\_s == Roles // Roles

and TableName\_s == UsersRoles // Users Roles

and TypeofChange\_Item\_s == Insert // Insert

| extend UserAssigned = extract(@"^.{1,33}\s\*?(.{1,12})\s\*?\d{16}", 1, ChangedTableKey\_s)

| extend Role = ObjectID\_s;

let UnitedRoles =

toscalar(union fixedRoles, SensitiveRoles

| summarize Roles = make\_list(Role));

// Query logic

ABAPAuditLog\_CL

| where MessageID\_s in (AuditClasses)

| summarize by TimeGenerated, TerminalIPv6\_s, User\_s, Host\_s, Email\_s

| lookup kind = leftouter (ChangeCheck) on User\_s

| where Role in (UnitedRoles)

| project-rename TimeGenAudit = TimeGenerated1

| where abs(datetime\_diff('second', TimeGenerated, TimeGenAudit)) <= logsThreshold

or isnull(TimeGenAudit)

| project

// Details

TimeGenerated, SystemID\_s, ClientID\_s, Role, User\_s, UserAssigned,

AccountCustomEntity = Email\_s, IPCustomEntity = TerminalIPv6\_s, HostCustomEntity = Host\_s

## SAP - Medium - Multiple Logons by User

### Description

Identifies logon of the same user from several terminals within scheduled time interval.

Source Action: Logon using the same user thorugh different IP's.

Data Sources: SAPcon - Audit Log

### Tactics

PreAttack

Credential Access

Initial Access

Collection

### KQL

// Define variables

// Audit Log Classes - Dialog Logon Successful, RFC Logon Successful

let AuditClasses = dynamic(['AU1','AU5']);

// Dialog / CPIC / RFC Int / RFC Ext / SRFC / User Switch / HTTP / Restore Session / API Call

let DialogLogonTypes = dynamic(['A', 'C', 'F', 'R', 'S', 'U', 'H', 'u', ' ']);

let excUsers = \_GetWatchlist('SAP - Excluded Users'); // Users that should be removed from query

let fixedExcUsers = datatable(User:string)

// Maintain these if WatchList is not available

["SYSWF"]

;

let UnitedExcUsers =

toscalar(union excUsers, fixedExcUsers

| summarize Users = make\_set(User));

let IPThreshold = 1;

// Query Logic

ABAPAuditLog\_CL

| where MessageID\_s in (AuditClasses)

| where Variable1\_s in (DialogLogonTypes)

| where User\_s !in (UnitedExcUsers)

| summarize CountIP = dcount(TerminalIPv6\_s), IPs = make\_set(TerminalIPv6\_s), StartTime = min(TimeGenerated), EndTime = max(TimeGenerated) by SystemID\_s, ClientID\_s, User\_s, Email\_s

| where CountIP > IPThreshold // Count of IP logins from the user is higher than threshold

| mv-expand IPs to typeof(string ) // Show for each IP

| project SystemID\_s, ClientID\_s, User\_s, StartTime, EndTime,

AccountCustomEntity = Email\_s, IPCustomEntity = IPs

## SAP - Medium - Critical authorizations assignment - New User Assignment

### Description

Identifies assignment of a critical authorization object value to a new user.

Source Action: Assign a new user to a role which holds critical authorization values using SU01/PFCG.

Critical authorization objects should be maintained in watchlist ""SAP - Critical Authorization Objects""

Data Sources: SAPcon - Change Documents Log

### Tactics

Privilege Escalation

### KQL

// New Assigned Users

let ObjectClassRoles = 'PFCG';

let TableName = 'CD1251';

let UsersRoles = 'AGR\_USERS';

let Insert = "I";

let NotInUse = 'NOT\_IN\_USE';

let logsThreshold = 3; // 3 seconds

// Audit Log Classes - Authorizations for user changed

let AuditClasses = dynamic(['AUB','AUD']); // Authorizations for user &A changed. User Master Record Changed

// Roles Change Documents - Extract Auth Object and Obj Field

let allHistory = ago(0d);

let alertSched = ago(6h); // Please maintain according to schedule

// Maintain these if System doesn't have CR's

let fixedChangeDocs = datatable(User\_s : string, ObjectClass\_s : string, TableName\_s : string, TypeofChange\_Item\_s : string , ChangedTableKey\_s : string, ObjectID\_s : string, TimeGenerated : datetime, ValueNew\_s : string, SystemID\_s : string)[];

let ChangeDocs =

union isfuzzy=true table("ABAPChangeDocsLog\_CL"), fixedChangeDocs;

let RolesAuthObject = ChangeDocs

| where TimeGenerated <= allHistory

| where ObjectClass\_s == ObjectClassRoles and TableName\_s == TableName // Role-Obj-Profile-ObjField

| where TypeofChange\_Item\_s in ('J', 'I', 'U') // Insert

| extend RoleObjProfileObjFieldVer = ChangedTableKey\_s, Role = ObjectID\_s

| extend ObjFieldValue = ValueNew\_s

| extend ObjField = trim(@"\s\*?", extract(@"(^.{1,30})\s\*?(.{1,10})\s\*?(.{1,12})\s\*?(.{1,10})\s\*?\d{6}", 4, RoleObjProfileObjFieldVer, typeof(string)))

| extend AuthObject = trim(@"\s\*?", extract(@"(^.{1,30})\s\*?(.{1,10})\s\*?(.{1,12})\s\*?(.{1,10})\s\*?\d{6}", 2, RoleObjProfileObjFieldVer, typeof(string)))

| summarize by SystemID\_s, Role, AuthObject, ObjField, ObjFieldValue;

let ComplexAuth = \_GetWatchlist('SAP - Critical Authorizations');

let SimpleAuth = \_GetWatchlist('SAP - Critical Authorizations');

let fixedComplexAuth = datatable(AuthorizationObject: string, AuthorizationField: string, AuthorizationValue: string, ActivityField: string, Activity: string)

['S\_DEVELOP', 'OBJTYPE', 'DEBUG', 'ACTVT', '\*',

'S\_DEVELOP', 'OBJTYPE', 'DEBUG', 'ACTVT', '02']; // Maintain these if WatchList is not available

let fixedSimpleAuth = datatable(AuthorizationObject: string, AuthorizationField: string, AuthorizationValue: string, ActivityField: string, Activity: string)

['S\_TCODE', 'TCD', '\*', 'NOT\_IN\_USE', '',

'S\_TZONE', 'ACTVT', '\*', 'NOT\_IN\_USE', '']; // Maintain these if WatchList is not available

let usersinRole =

ChangeDocs

| where TimeGenerated >= alertSched

| where ObjectClass\_s == ObjectClassRoles // Roles

and TableName\_s == UsersRoles // Users Roles

and TypeofChange\_Item\_s == Insert // Insert

| extend UserAssigned = extract(@"^.{1,33}\s\*?(.{1,12})\s\*?\d{16}", 1, ChangedTableKey\_s)

| extend Role = ObjectID\_s

| extend TimeGenUserinRole = TimeGenerated;

//| summarize by TimeGenerated, SystemID\_s, ClientID\_s, Role, UserAssigned, User\_s

let RolesAuthObjectCheck =

RolesAuthObject

| extend ObjFieldVal = ObjFieldValue

| lookup kind = leftouter

(RolesAuthObject

| extend ActivityVal = ObjFieldValue)

on Role, AuthObject;

let complexScenario =

union ComplexAuth, fixedComplexAuth

| where ActivityField != NotInUse

| summarize by AuthorizationObject, AuthorizationField, AuthorizationValue, ActivityField, Activity

| lookup kind = inner (RolesAuthObjectCheck)

on $left.AuthorizationObject == $right.AuthObject

and $left.AuthorizationField == $right.ObjField

and $left.AuthorizationValue == $right.ObjFieldValue

and $left.ActivityField == $right.ObjField1

and $left.Activity == $right.ActivityVal;

let simpleScenario =

union SimpleAuth, fixedSimpleAuth

| where ActivityField == NotInUse

| summarize by AuthorizationObject, AuthorizationField, AuthorizationValue, ActivityField, Activity

| lookup kind = inner (RolesAuthObject)

on $left.AuthorizationObject == $right.AuthObject

and $left.AuthorizationField == $right.ObjField

and $left.AuthorizationValue == $right.ObjFieldValue;

let GetEntites =

ABAPAuditLog\_CL

| where TimeGenerated >= alertSched

| where MessageID\_s in (AuditClasses)

| summarize by TimeGenerated, TerminalIPv6\_s, ClientID\_s, User\_s, Host\_s, Email\_s

| extend TimeGenAudit = TimeGenerated;

union complexScenario, simpleScenario

| lookup kind = inner (usersinRole) on SystemID\_s, Role

| lookup kind = leftouter (GetEntites) on User\_s

| where abs(datetime\_diff('second', TimeGenUserinRole, TimeGenAudit)) <= logsThreshold or

isnull(TimeGenAudit)

| project

// Details

TimeGenUserinRole, SystemID\_s, ClientID\_s, Role, User\_s, UserAssigned, AuthorizationObject, AuthorizationField, AuthorizationValue, ActivityField, Activity,

AccountCustomEntity = Email\_s, IPCustomEntity = TerminalIPv6\_s, HostCustomEntity = Host\_s

## SAP - Medium - Critical authorizations assignment - New Authorization Value

### Description

Identifies assignment of a critical authorization object value to a new user.

Source Action: Assign a new authorization object / update existing one in a role using PFCG.

Critical authorization objects should be maintained in watchlist ""SAP - Critical Authorization Objects""

Data Sources: SAPcon - Change Documents Log

### Tactics

Privilege Escalation

### KQL

// New Assigned Objects

let ObjectClassRoles = 'PFCG';

let TableName = 'CD1251';

let UsersRoles = 'AGR\_USERS';

let Insert = "I";

let NotInUse = 'NOT\_IN\_USE';

let logsThreshold = 3; // 3 seconds

// Audit Log Classes - Authorizations for user changed

let AuditClasses = dynamic(['AUR','AUT']); // Authorization/Authorization Profile &B created / changed.

// Roles Change Documents - Extract Auth Object and Obj Field

let allHistory = ago(0d);

let alertSched = ago(6h); // Please maintain according to schedule

// Maintain these if System doesn't have CR's

let fixedChangeDocs = datatable(User\_s : string, ObjectClass\_s : string, TableName\_s : string, TypeofChange\_Item\_s : string , ChangedTableKey\_s : string, ObjectID\_s : string, TimeGenerated : datetime, ValueNew\_s : string, SystemID\_s : string)[];

let ChangeDocs =

union isfuzzy=true table("ABAPChangeDocsLog\_CL"), fixedChangeDocs;

let RolesAuthObject = ChangeDocs

| where TimeGenerated >= alertSched

| where ObjectClass\_s == ObjectClassRoles and TableName\_s == TableName // Role-Obj-Profile-ObjField

| where TypeofChange\_Item\_s in ('J', 'I', 'U') // Insert

| extend RoleObjProfileObjFieldVer = ChangedTableKey\_s, Role = ObjectID\_s

| extend ObjFieldValue = ValueNew\_s

| extend ObjField = trim(@"\s\*?", extract(@"(^.{1,30})\s\*?(.{1,10})\s\*?(.{1,12})\s\*?(.{1,10})\s\*?\d{6}", 4, RoleObjProfileObjFieldVer, typeof(string)))

| extend AuthObject = trim(@"\s\*?", extract(@"(^.{1,30})\s\*?(.{1,10})\s\*?(.{1,12})\s\*?(.{1,10})\s\*?\d{6}", 2, RoleObjProfileObjFieldVer, typeof(string)))

| extend TimeGenRoleAuth = TimeGenerated;

let ComplexAuth = \_GetWatchlist('SAP - Critical Authorizations');

let SimpleAuth = \_GetWatchlist('SAP - Critical Authorizations');

let fixedComplexAuth = datatable(AuthorizationObject: string, AuthorizationField: string, AuthorizationValue: string, ActivityField: string, Activity: string)

['S\_DEVELOP', 'OBJTYPE', 'DEBUG', 'ACTVT', '\*',

'S\_DEVELOP', 'OBJTYPE', 'DEBUG', 'ACTVT', '02']; // Maintain these if WatchList is not available

let fixedSimpleAuth = datatable(AuthorizationObject: string, AuthorizationField: string, AuthorizationValue: string, ActivityField: string, Activity: string)

['S\_TCODE', 'TCD', '\*', 'NOT\_IN\_USE', '',

'S\_TZONE', 'ACTVT', '\*', 'NOT\_IN\_USE', '']; // Maintain these if WatchList is not available

let usersinRole =

ChangeDocs

| where TimeGenerated <= allHistory

| where ObjectClass\_s == ObjectClassRoles // Roles

and TableName\_s == UsersRoles // Users Roles

and TypeofChange\_Item\_s == Insert // Insert

| extend UserAssigned = extract(@"^.{1,33}\s\*?(.{1,12})\s\*?\d{16}", 1, ChangedTableKey\_s)

| extend Role = ObjectID\_s

| summarize by SystemID\_s, Role, UserAssigned;

let RolesAuthObjectCheck =

RolesAuthObject

| extend ObjFieldVal = ObjFieldValue

| lookup kind = leftouter

(RolesAuthObject

| extend ActivityVal = ObjFieldValue)

on Role, AuthObject;

let complexScenario = union ComplexAuth, fixedComplexAuth

| where ActivityField != NotInUse

| summarize by AuthorizationObject, AuthorizationField, AuthorizationValue, ActivityField, Activity

| lookup kind = inner (RolesAuthObjectCheck)

on $left.AuthorizationObject == $right.AuthObject

and $left.AuthorizationField == $right.ObjField

and $left.AuthorizationValue == $right.ObjFieldValue

and $left.ActivityField == $right.ObjField1

and $left.Activity == $right.ActivityVal;

let simpleScenario =

union SimpleAuth, fixedSimpleAuth

| where ActivityField == NotInUse

| summarize by AuthorizationObject, AuthorizationField, AuthorizationValue, ActivityField, Activity

| lookup kind = inner (RolesAuthObject)

on $left.AuthorizationObject == $right.AuthObject

and $left.AuthorizationField == $right.ObjField

and $left.AuthorizationValue == $right.ObjFieldValue;

let GetEntities =

ABAPAuditLog\_CL

| where MessageID\_s in (AuditClasses)

| summarize by TimeGenerated, ClientID\_s, TerminalIPv6\_s, User\_s, Host\_s, Email\_s

| extend TimeGenAudit = TimeGenerated;

union complexScenario, simpleScenario

| lookup kind = inner (usersinRole) on SystemID\_s, Role

| lookup kind = leftouter (GetEntities) on User\_s

| where abs(datetime\_diff('second', TimeGenRoleAuth, TimeGenAudit)) <= logsThreshold or

isnull(TimeGenAudit)

| project

// Details

TimeGenRoleAuth, SystemID\_s, ClientID\_s, Role, User\_s, UserAssigned, AuthorizationObject, AuthorizationField, AuthorizationValue, ActivityField, Activity,AccountCustomEntity = Email\_s, IPCustomEntity = TerminalIPv6\_s, HostCustomEntity = Host\_s

## SAP - Medium - Multiple Logons by IP

### Description

Identifies logon of several users from same IP within scheduled time interval.

Source Action: Logon using several users thorugh the same IP.

Data Sources: SAPcon - Audit Log

### Tactics

Initial Access

### KQL

// Define variables

let AuditClasses = dynamic(['AU1', 'AU5']); // Audit Log Classes - Dialog Logon Successful, RFC Logon Successful

let DialogLogonTypes = dynamic(['A', 'C', 'F', 'R', 'S', 'U', 'H', 'u', ' ']); // Dialog / CPIC / RFC Int / RFC Ext / SRFC / User Switch / HTTP / Restore Session / API Call

let excNetworks = \_GetWatchlist('SAP - Excluded Networks'); // Networks that should be removed from query

let fixedNetworks =

datatable(Network:string)

// Maintain these if WatchList is not available

["111.68.128.0/1", "123.68.128.0/1"]

;

let UnitedNetworks =

toscalar(union excNetworks, fixedNetworks

| summarize Networks = make\_set(Network));

let UsersperIP = 1;

// Query logic

ABAPAuditLog\_CL

| where MessageID\_s in (AuditClasses)

| where Variable1\_s in (DialogLogonTypes)

| where TerminalIPv6\_s !in (UnitedNetworks)

| extend UserandEmail = pack("ID", User\_s, "Email", Email\_s)

| summarize CountUsers = dcount(strcat(User\_s, "\_&\_", Email\_s)), Users = make\_set(UserandEmail), StartTime = min(TimeGenerated), EndTime = max(TimeGenerated)

by SystemID\_s, ClientID\_s, TerminalIPv6\_s

| where CountUsers > UsersperIP

| mv-expand Users

| evaluate bag\_unpack(Users, "User\_")

| project SystemID\_s, ClientID\_s, IPCustomEntity = TerminalIPv6\_s, StartTime, EndTime,

column\_ifexists("User\_ID", ""),

AccountCustomEntity = column\_ifexists("User\_Email", "")

## SAP - Medium - Brute force attacks

### Description

Identifies brute force attacks on SAP system according to failed logon attempts for the backend system.

Source Action: Attempt to login from the same IP to several systems/clients within the scheduled time interval.

Data Sources: SAPcon - Audit Log

### Tactics

Credential Access

### KQL

// Define variables

// Audit Log Classes - Failed Logons / Password Check

let AuditClasses = dynamic(['AUO', 'AU2', 'AU6', 'BU1']);

let perIPLimit = 6;

// Query logic

ABAPAuditLog\_CL

| where MessageID\_s in (AuditClasses)

| extend DetailsBy = pack("User", User\_s, "Email", Email\_s, "SystemID", SystemID\_s, "ClientID", ClientID\_s)

| summarize LoginbyIPAttempts = count(), Details = make\_set(DetailsBy), StartTime = min(TimeGenerated), EndTime = max(TimeGenerated)

    by TerminalIPv6\_s

// Check if number of login attempts per IP is higher than limit

| where LoginbyIPAttempts > perIPLimit

| mv-expand Details

| evaluate bag\_unpack(Details, "Details\_")

| project

    StartTime, EndTime,

    IPCustomEntity = TerminalIPv6\_s,

    AccountCustomEntity  = column\_ifexists("Details\_Email", ""),

    column\_ifexists("Details\_User", ""),

    column\_ifexists("Details\_SystemID", ""),

    column\_ifexists("Details\_ClientID", "")

## SAP - Medium - Debugging Activities

### Description

Identifies all debugging related activities.

Source Action: Activate Debug ("/h") in system, debug an active process, add breakpoint to source code etc.

**Recommended for Production only**

Data Sources: SAPcon - Audit Log

### Tactics

Discovery

### KQL

let Role = 'Production';

let DebuggerProgram = 'RSTPDAMAIN';

let AuditClasses = dynamic(['CUK','CUL','CUM','CUN','CUO','CUP']); // Audit Log Classes - Debug Activities

let allSystemRoles = dynamic(['Sandbox','Developement','QualityAssurance','Training','Production']); // Available System Roles

let allSystemUsage = dynamic (['ERP','CRM','BW','Solman','Gateway','Enterprise Portal']); // Available System Usages

// Get Relevant Systems from WatchList

let systemID = \_GetWatchlist('SAP - Systems')

| where SystemRole == Role; // Reccommended is Production only

let fixedSID = datatable(SystemID:string, SystemRole:string, SystemUsage:string)

// Maintain these if WatchList is not available

["S4H","Production","ERP",

"XXX","Sandbox","BW"]

| where SystemRole == Role // Recommended is Production only

;

let SystemUnited = union systemID, fixedSID

| summarize by SystemID, SystemRole, SystemUsage;

//| where SystemRole in (allSystemRoles); // Use this for all system roles

ABAPAuditLog\_CL

| where MessageID\_s in (AuditClasses) or ABAPProgramName\_s == DebuggerProgram // Get logs by messege ID or program name

| project-rename SystemID = SystemID\_s

| lookup kind=inner (SystemUnited) on SystemID

| order by TimeGenerated asc

| project TimeGenerated, User\_s, MessageText\_s, ABAPProgramName\_s, TransactionCode\_s, SystemID, SystemRole, SystemUsage,MessageID\_s, AccountCustomEntity = Email\_s, IPCustomEntity = TerminalIPv6\_s, HostCustomEntity = Host\_s

## SAP - Low - Sensitive Tables Direct Access By Dialog Logon

### Description

Identifies generic table access by dialog logon

Source Action: Open table contents using SE11/SE16/SE16N.

**Recommended for Production only**

Tables should be maintained in ""SAP - Sensitive Tables"" Watchlist.

Data Sources: SAPcon - Audit Log

### Tactics

Discovery

### KQL

// Define variables

let Role = "Production";

let AuditClasses = dynamic(['DU9']); // Dialog, Audit Log Classes - Generic Table Access

let allSystemRoles = dynamic(['Sandbox','Developement','QualityAssurance','Training','Production']); // Available System Roles

let allSystemUsage = dynamic (['ERP','CRM','BW','Solman','Gateway','Enterprise Portal']); // Available System Usages

// Get Relevant Systems from WatchList

let systemID = \_GetWatchlist('SAP - Systems');

let fixedSID = datatable(SystemID:string, SystemRole:string, SystemUsage:string)

// Maintain these if WatchList is not available

["S4H","Production","ERP",

"XXX","Sandbox","BW"]

;

// Get Relevant Tables

let SensitiveTables = \_GetWatchlist('SAP - Sensitive Tables');

let fixedTables = datatable(Table:string)

// Maintain these if WatchList is not available

["USR02"]

;

let RelSystemID = union systemID, fixedSID // Create a variable that stores relevant Systems

| where SystemRole == Role // Recommended is Production only

//| where SystemRole in (allSystemRoles); // Use this for all system roles

| summarize by SystemID;

let SensitiveUnionTables = union SensitiveTables, fixedTables // Create a variable that stores relevant sensitive tables

| summarize by Table;

// Query logic

ABAPAuditLog\_CL

| project-rename Table = Variable1\_s, Activity = Variable2\_s

| where MessageID\_s in (AuditClasses)

| where SystemID\_s in (RelSystemID)

| where Table in (SensitiveUnionTables)

| order by TimeGenerated asc

| project TimeGenerated, SystemID\_s, ClientID\_s, User\_s, TransactionCode\_s, ABAPProgramName\_s, Table, Activity, MessageText\_s, MessageID\_s,

AccountCustomEntity = Email\_s, IPCustomEntity = TerminalIPv6\_s, HostCustomEntity = Host\_s

## SAP - Low - Multiple Password Changes by User

### Description

Identifies multiple password changes by user.

Source Action: Change user password

Data Sources: SAPcon - Audit Log

### Tactics

Credential Access

### KQL

// Define variables

let systemsPerUser = 3; // Systems Clients per User

let AuditClasses = dynamic(['BU2']); // Audit Log Classes - Password Changed

// Query logic

ABAPAuditLog\_CL

| where MessageID\_s in (AuditClasses)

| project-rename UserReset = Variable2\_s

| extend DetailsBy = pack("SystemID", SystemID\_s, "ClientID", ClientID\_s)

| summarize CountSysClient = dcount(strcat(SystemID\_s, ClientID\_s)), Details = make\_set(DetailsBy), StartTime = min(TimeGenerated), EndTime = max(TimeGenerated)

    by UserReset, User\_s, Email\_s, TerminalIPv6\_s

| where CountSysClient > systemsPerUser // Number of passwords changed by user

| mv-expand Details

| evaluate bag\_unpack(Details, "Details\_") // Unpack the details to a couple of fields

| project

    StartTime, EndTime, UserReset, User\_s,

    AccountCustomEntity = Email\_s,

    IPCustomEntity = TerminalIPv6\_s,

    column\_ifexists("Details\_SystemID", ""),

    column\_ifexists("Details\_ClientID", "")

## SAP - High - HANA DB - Assign Admin Authorizations

### Description

Identifies admin privileges/roles assignment.

Source Action: Assign a user with any Admin role / privileges.

Data Sources: Linux Agent - Syslog

### Tactics

Privilege Escalation

### KQL

Syslog

| where ProcessName startswith "HDB"

| where SyslogMessage contains "ADMIN" and (SyslogMessage contains "GRANT PRIVILEGE" or SyslogMessage contains "GRANT ROLE")

// Entities Mapping

| extend HostCustomEntity = HostName

## SAP - High - HANA DB - User Admin actions

### Description

Identifies user administration actions.

Souirce Action: Create/Update/Delete a DB User.

*Data Sources: Linux Agent - Syslog*

### Tactics

Privilege Escalation

### KQL

Syslog

| where ProcessName startswith "HDB"

| where SyslogMessage contains "CREATE USER" or

SyslogMessage contains 'ALTER USER' or

SyslogMessage contains 'DROP USER' or

SyslogMessage contains 'DROP SCHEMA'

// Entities Mapping

| extend HostCustomEntity = HostName

## SAP - High - HANA DB - Audit Trail Policy Changes

### Description

Identifies changes for HANA DB audit trail policies.

Source Action: Create / update existing audit policy in security definitions.

*Data Sources: Linux Agent - Syslog*

### Tactics

Lateral Movement

Defense Evasion

Persistence

### KQL

Syslog

| where ProcessName startswith "HDB"

| where SyslogMessage contains "AUDIT POLICY"

// Entities Mapping

| extend HostCustomEntity = HostName

## SAP - High - HANA DB - Deactivation of Audit Trail

### Description

Identifies deactivation of HANA DB audit log.

Source Action: Deactivate Audit Log in HANA DB security defnitions.

*Data Sources: Linux Agent - Syslog*

### Tactics

Persistence

Lateral Movement

Defense Evasion

### KQL

Syslog

| where ProcessName startswith "HDB"

| where SyslogMessage contains "AUDIT CONFIGURATION" and

SyslogMessage contains 'global\_auditing\_state' and

SyslogMessage contains 'False'

// Entities Mapping

| extend HostCustomEntity = HostName

## SAP - High - Sensitive privileged user makes a change in other user

Identifies changes of sensitive privileged users in other users.

Source Action: Change user details / authorizations using SU01.

Priveleged users should be maintained in "SAP - Privileged Users" Watchlist

Data Sources: SAPcon - Audit Log

### Tactics

Privilege Escalation

Credential Access

### KQL

// Audit Log Classes - User Master Changes

let AuditClasses = dynamic(['AU7', 'BUV', 'BUW', 'AU8', 'AU9', 'AUA', 'AUB', 'AUD', 'DUH', 'BU2']);

// Get Relevant User from WatchList

let PrivelegedUsers = \_GetWatchlist('SAP - Privileged Users');

let fixedUsers = datatable(User: string)

// Maintain these if WatchList is not available

["SAP\*", "DDIC"]

;

let UnitedPrivleged = union PrivelegedUsers, fixedUsers

| summarize by User;

ABAPAuditLog\_CL

| where MessageID\_s in (AuditClasses)

| where User\_s in (UnitedPrivleged) // The user that makes a change is a sensitive privileged user

| project TimeGenerated, SystemID\_s, ClientID\_s, User\_s, MessageText\_s, MessageID\_s,

AccountCustomEntity = Email\_s, IPCustomEntity = TerminalIPv6\_s, HostCustomEntity = Host\_s