

SMS Aged Password Detector

Description

The Microsoft SMS (System Management Server) periodically gathers configuration information from Windows domain members. One piece of information is a list of accounts and the corresponding password age for each domain member.

The Authentication and Directory Services group has selected accounts from the SMS database which have password ages that are greater than the number of days since the system joined the Windows domain. These selected account records have been placed in a database view named 'Password_overdue'.

The database containing the view Password_overdue is named SCCM_DCM and is hosted on server Fermi-SCCM01.fnal.gov. The Authentication group supplied a read-only account and password to access this database.

Implementation

The detector is implemented via a UPD package named 'sms_aged_pwd_detector'. The detector module is named 'SMS_Aged_Pwd_Detector.py'.

The detector requires a database connection to the MSSQL database. The FreeTDS ODBC driver is used. To query MISCOMP, the detector uses the 'cst_ra_api' package.

Deployment

The detector is designed to run daily and can be configured to run from a crontab file. Here is an example from the nimisrv (development) system:

```
# Run the SMS aged password detector once per day (development mode, no real events!)
-- send mail w/output)
30 4 * * * . /usr/local/etc/setups.sh && setup -qdev sms_aged_pwd_detector &&
SMS_Aged_Pwd_Detector.py -n --yesterday -d3
```

The environment needs to be set before running the detector. The command 'setup -qdev sms_aged_pwd_detector' initializes the environment. The most used options are:

```
-n                do no generate TIssue events
--yesterday      select rows with Timekey field equal to yesterday's date
--date=YYYY-MM-DD    select rows with Timekey field equal to given date
-d3              set debug level to 3 (HIGH)
```

For production, the only option needed is '--yesterday'. The environment is initialized using 'setup -qprd sms_aged_pwd_detector'.